

UML:- UML is a standard for specifying, visualising, constructing & documenting the artifacts of the software system

Book:- UML User guide by Booch, Rumbaugh and Jacobson, published by Pearson's Publications.

* Virtual language to make software blue-prints.

Building Blocks of UML:-

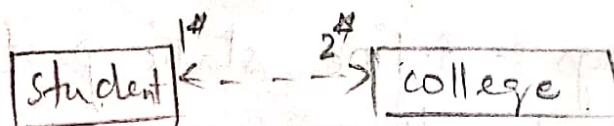
- | | | |
|-----------------|------------------|-------------------|
| 1) Things | 2) Relationships | 3) Diagrams |
| ↳ Structural | ↳ Association | ↳ class Diagram |
| ↳ Behavioural | ↳ Dependency | ↳ Object " |
| ↳ Group | ↳ Generalisation | ↳ Use case " |
| ↳ Annotational. | ↳ Realisation | ↳ Sequence " |
| | | ↳ Collaboration " |
| | | ↳ State chart " |
| | | ↳ Activity " |
| | | ↳ Component " |
| | | ↳ Deployment " |

Relationships:-

1) Dependency Relationships:-



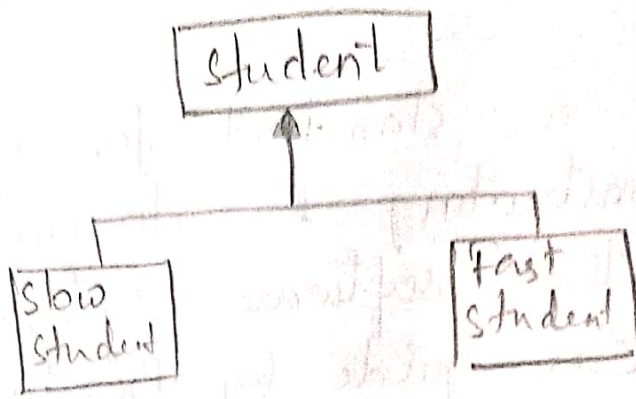
2) Association:-



1 - student
2 - college

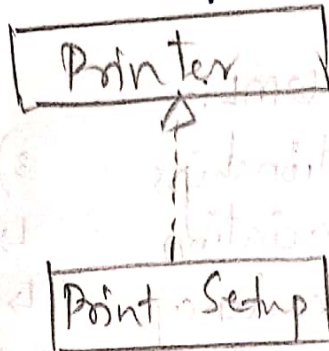
3) Generalisation:-

specialized entity to its more generalized entity.



g) Realisation :-

An entity specifies some responsibilities but other entity will implement it.



Apart from these, the other two forms of relationships are aggregation & composition.

Aggregation:- whole & part relationship.



Composition:- depicts strong relationship

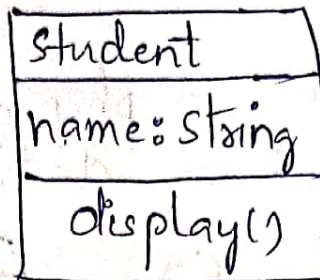
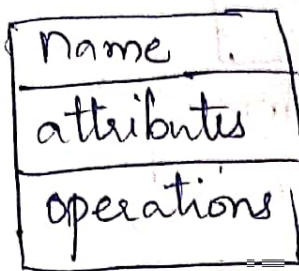


Things:-

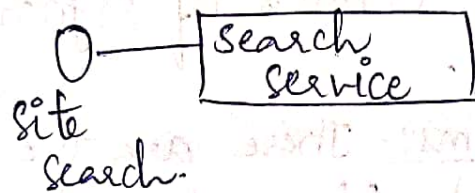
1) Structural Things :- define static part of the model. Structural things represent conceptual & physical elements.

Ex:- class.

1) class represents set of objects having similar properties.



2) Interface



Interface defines set of operations that specifies responsibility of the class.

Interface name - site search.

Interface site search is implemented by the service called search services.

3) Collaboration:-

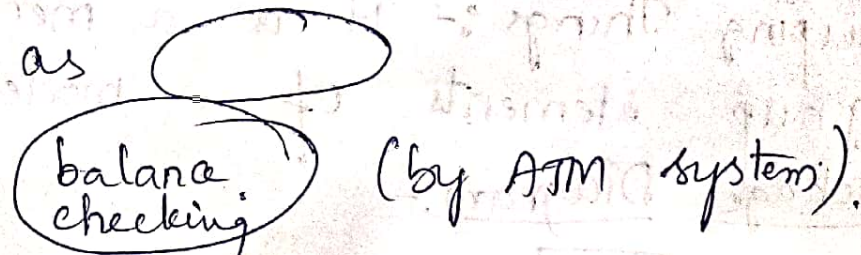
Notation for representing collaboration [

> It defines interaction b/w elements.

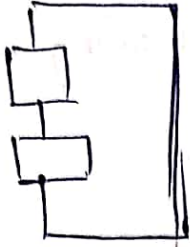
4) Use-case Diagram:- represents set of actions performed by the system for a specific task.

> denoted as

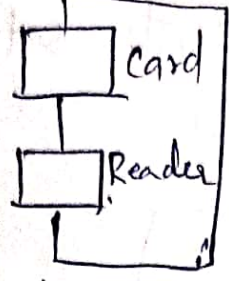
Ex:-



5) Component Diagram:- describes physical part of the system & can be drawn as.



Ex:-



5) Node:- Node is a physical element that exists at run time. Ex:- database, webpage.



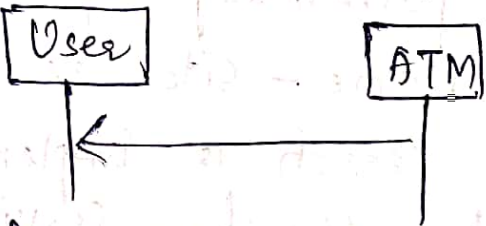
Ex:-



Behavioural things:- consists of dynamic part of the model.


1) Interaction diagrams:- These are the messages that are exchanged b/w elements for a particular task & are shown as. →

Ex:-



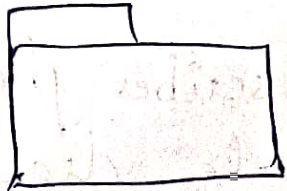
2) State Machine Diagram:- (i). These represent state of the during its lifecycle.

(ii) State Machine Diagram also represents sequence of states upon events.

(iii) denoted as 

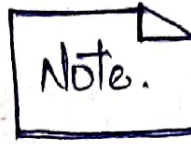
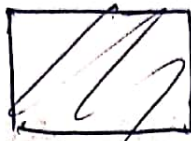
Grouping Things:- It is a mechanism to group elements of a model.

1) Package Diagram:-



Ex:- ATM Testing

Annotational ^{things} Testing :- It is a mechanism to capture comments or remarks of the model elements.



UML Diagrams :-

UML diagrams are classified into

1) Structural

(i) Use case

(ii) class

(iii) Object

(iv) Component

(v) Deployment

2) Behavioural.

(i) Sequence

(ii) Collaboration

(iii) State chart

(iv) Activity

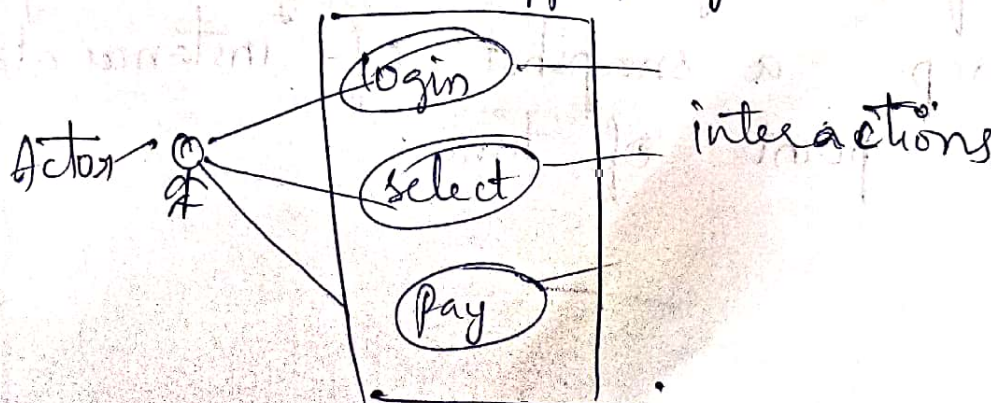
(i) Use-Case Diagrams :-

~~Use Case Diagrams~~

> Use Case Diagram represent functionalities of the system

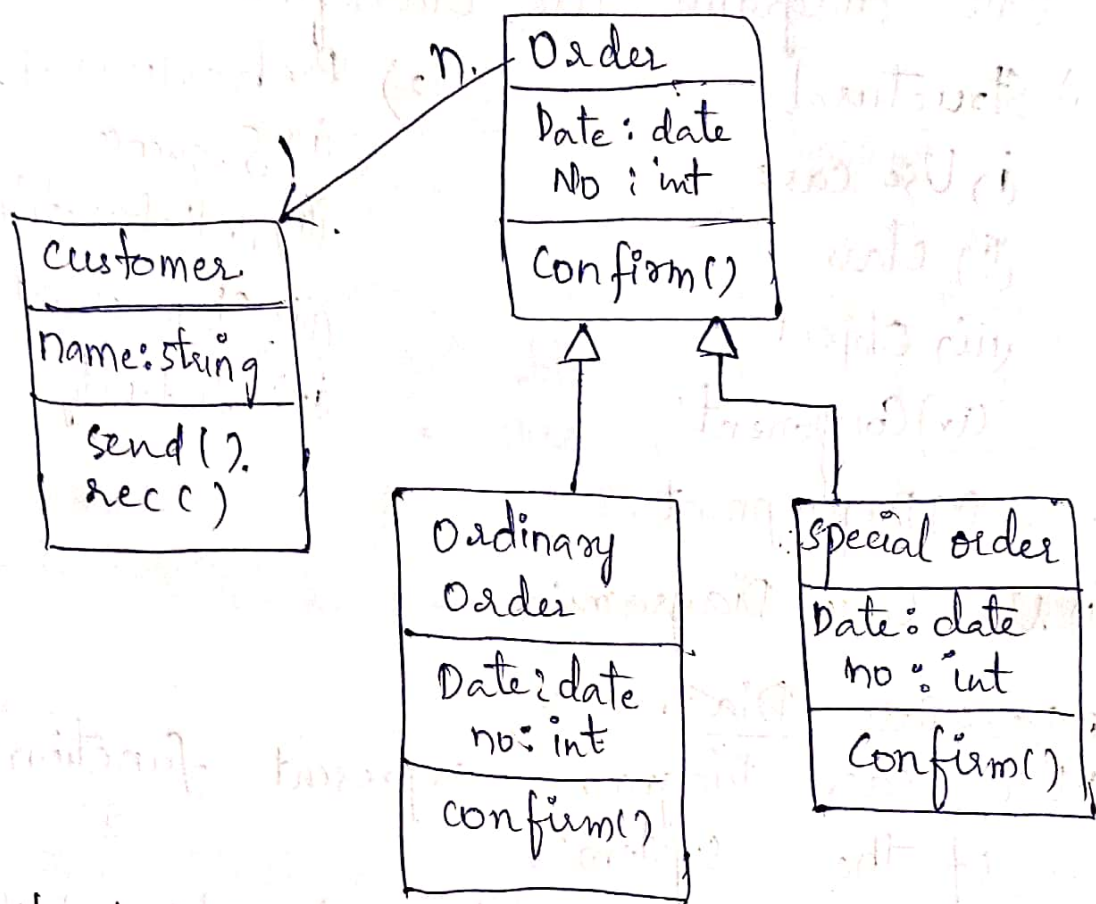
> They also represent interactions between actor and use cases.

Ex:- Online shopping system



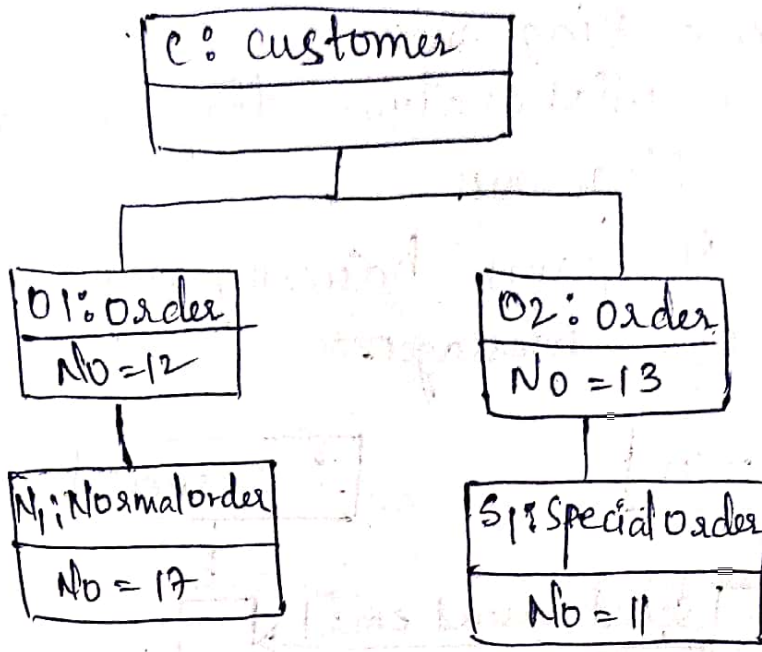
(ii) Class Diagram:-

- > In class Diagram, we represent set of class, their interfaces, collaborations and relationships.
- > Basically, class diagram represents static design view of the system.



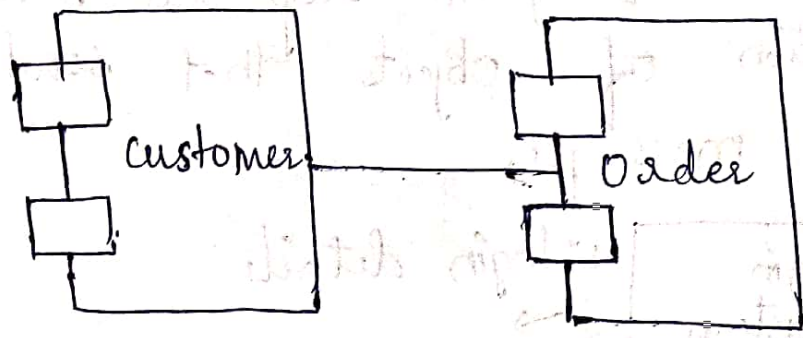
(iii) Object Diagram:-

- > Object diagrams represent static design view of the system. Not only that, they also rep. a snapshot of instances at a given point of time.



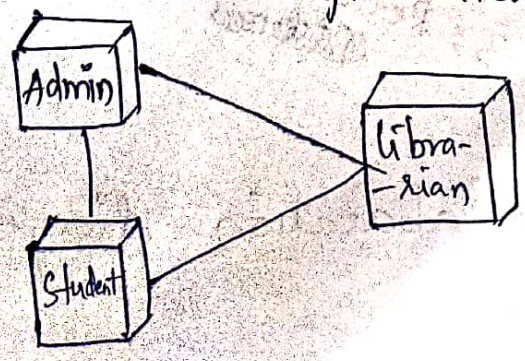
(iv) Component Diagram:-

- > Component Diagrams are used to describe working of system components.
- > They represent static implementation view of the system.



(v) Deployment Diagram:-

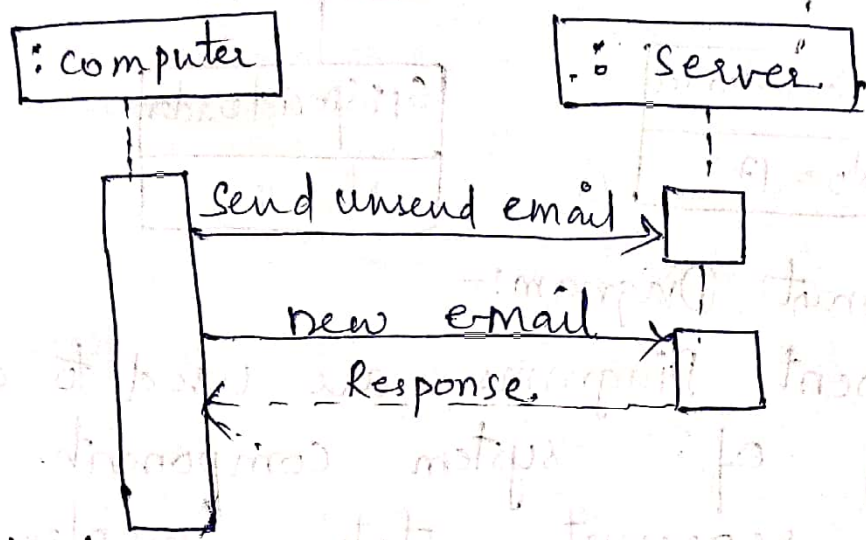
- > represent how components are deployed in hardware. Not only that they also represent static design view of the system.



29/12/2020

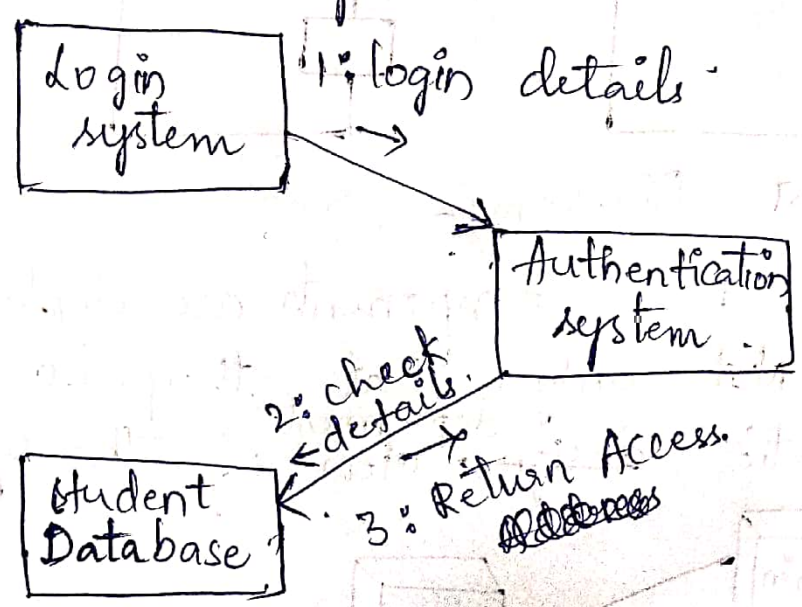
(vi) Sequence Diagrams:-

- > Sequence & Collaboration diagrams are called interaction Diagrams.
- > Sequence Diagrams basically represent time ordering of messages.



(vii) Collaboration Diagram:-

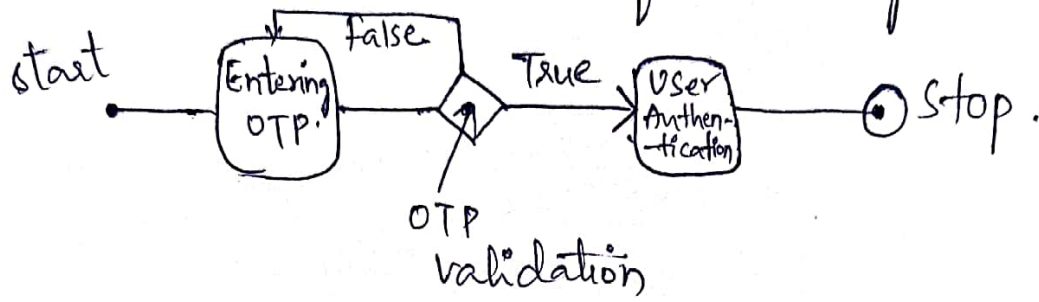
Collaboration Diagram represent structural organisation of objects that send and receive messages.



> Sequence & Collaboration diagrams are isomorphic.

(viii) State chart Diagram:-

State chart Diagrams represent event ordered behaviours of an object.



(ix) Activity Diagram:-

Activity Diagram represent flow from activity to activity.

