Page No.:
Date: / /

Building the analysis model:-

Analysis model is a technical representation of the ~~s/m~~ system. It acts as a link bet^n ~~s/m~~ system description & design model.

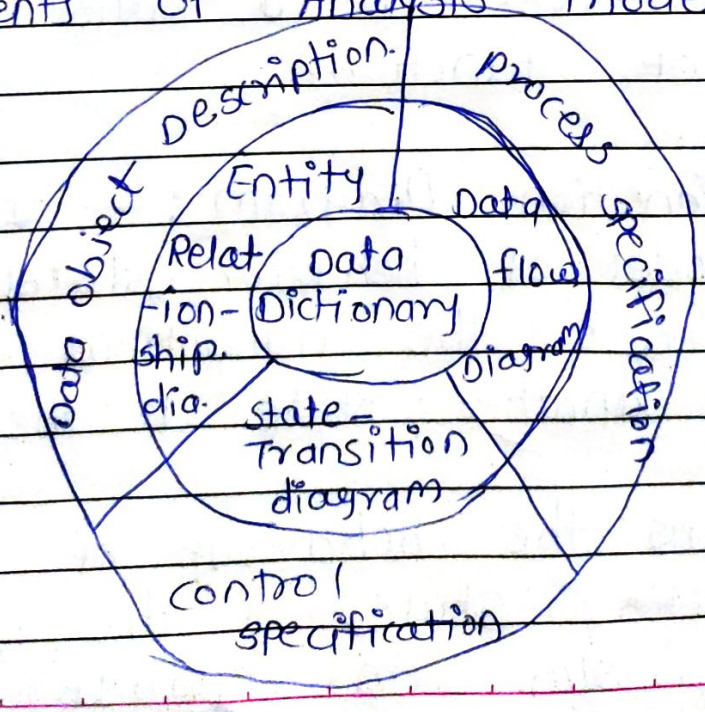
- In Analysis modeling, information, behaviour & fun^n of the s/m are defined & translated into the architecture, component, & interface level design in the design modeling.

objectives of Analysis modelling:

(develop)

- It must establish a way of creating s/w design
- It must describe the requirements of the customer
- It must define a set of requirements that can be validated, once the s/w is built.

Elements of Analysis model:



1) Data Dictionary:- It is a consists of a description of all data objects used for produced by the s/w. It stores the colle^n of data present in the s/w.

(Important)

- It is a very crucial element of the analysis model.

2) Entity Relationship Diagram (ERD):-
The attributes of each object in the ERD can be described using data object description.

- It provides the basis for activity related to data design.

3) Data flow Diagram (DFD):-
It provides the additional information which used during the analysis of the info^n domain & serves as a basis for the modeling of function.

4) State Transition Diagram :- It shows various modes of behavior (states) of the s/m & also shows the transitions from one state to anothe state in the s/m.

- It represents the behaviour of a system by presenting it's states & the events that cause the s/m to change state.

5) process Specification :-
- It stores the description of each fun^n present in the data flow diagram.
- It describes the input to a fun^n, the algorithm that is applied for the transformation of input & the output that is produced.

6) Control specification :=
It stores additional info^n about the control aspects of the s/w.

- It also provides the details of the processes which are executed to manage events.

7) Data object Description :-

- It stores & provides complete knowledge about a data object present & used in the s/w.
- It incorporates (combines) all the data objects & their attributes.

CH-2.

Date

~~Risk~~ Management.

Requirement task:-

1) Negotiation

- This is fourth phase of the requirements analysis process.
- In the negotiation is between the developer and the customer.
- customer and developer are both satisfied with reference to the implementation.
- customers are asked to prioritize the requirements and make conflicts that may arise along with it.
- This phase emphasizes discussion and exchanging conversion is needed and to be eliminated.

The following are discussed in the negotiation phase:

- Availability of Resource.
- Delivery Time
- Scope of requirements
- project cost.
- Estimations on development.

Page No.
Date

4) Validation:-

This is the sixth phase of the requirement analysis process.

- This phase focuses on checking for error and debugging. In validation phase, the developer scan the specification document and check for the following:

1) All requirement have been stated & met correctly.
2) Error have been dubugged & corrected.
3) work product is built according the standard.

- This requirement validation mechanism is known ~~technical~~ as the formal technical review.
- validate the include software engineers, customer user and other Stakeholders.

* Design process and Design quality:-

- The design phase of software development deals with transforming the customers requirements as described in the SRS documents into a form implementable using a programming language.

- SDP can be divided into the following three levels of phases of design:

1) Interface Design
2) Architectural Design
3) Detailed Design.

1) Interface Design:-

- It is the specification of the interaction between system & its environment.

- It proceeds at a high level of abstraction with respect to the inner workings of the system ; i.e during design interface design, the internal of the systems are completely ignored & the system is treated as a black box.

2) Architectural Design:-

- It is a major com specification of the major components of a system. their responsibilities, properties interfaces & the relationships & interactions between them.
- In AD, the overall structure of the system is choosen. but the internal details of major components are ignored.

3) Detailed design:-

- Design is the specification of the internal elements of all major system components, their properties, relationships.

4) Design quality:-

- In the context of software engineering, software quality measures how well the software is designed & how well the software conforms to that design.
- Software quality product is defined in term of its fitness of purpose

* ~~The modern way virus~~
* several quality methods such as the following:-

1) Portability:-

- A software device is said to be portable, if it can be freely made to work in various operating system environments, in multiple machines, with other software products, etc.

2) Usability:- A software product has better usability if various categories of users can easily invoke the functions of the product

3) Reusability:- A software product has excellent reusability if different modules of the product can quickly be reused to develop new products.

4) Correctness:- A software product is correct if various requirements as specified in the SRS document have been correctly implemented.

5) maintainability:-
- A software product is maintainable if bug can be easily corrected.

Page No.
Date

Design concept is compilation of sketches, photographs & a written statement that explains the primary idea behind a product design.

good design concept constitutes a collection of mood boards, rough sketches, ideas & images.

It helps to keep track of the creative process so that the best design principle you can apply meet the target popwation's need.

design concept makes the goal of product & vision of the company very evident.

It not only helps to build brands & gain trust but also boosts the growth & sales of the company.

Design concept example -

Amazon logo concept is brilliant. Since it shows Amazon as a one - stop shop for everything from A to Z. The arrows shape also represents a smile, emphasizing its goal of keeping customer have.

majority of product designers idean depend on usefulness.

UNIT - II

Page No.
Date

* Design model :- Design modeling in software engineering represent the features of software that helps to develop it effectively, the architecture, the user interface and the component level detail.
- Design modeling provides different view of system like architecture ~~of~~ plan for home or building.

Working of Design modeling :-

- It is mainly classified into four categories - Data design, architecture design, interface design and component-level design.

- Data design: It represents the data object & their interrelationship in an entity relationship diagram.
- It shows the structure of data in terms of the tables.
- It shows three type of relationship - one to one, one to many, many to many.
- one to many relations ^ one entity is connected to another entity.
- In one to many relations ^ one entity is ~~com~~ connected more than one entity.
- many to many relations one entity is connected to more than one entity and also other entity is connected with first entity using more than one entity.

2) Architectural design: It is expressed as a block diagram defining an overview of the system structure - features of the components & how these components communicate with each other to share data.
- It defines the structure & properties of the component that are involved in the system & also the interrelationship among these components.

3) user Interface design:- It represents how the software communicates with the user i.e the behavior of the system.
- For ex: Military, vehicles, aircraft, audio equipment, computer peripherals are the areas where User interface design is implemented.
- UI design becomes efficient only after performing usability testing.
- This is done to test what works & what does not work as expected.

4) component level design:- It is a perfect way to share a large amount of data.
- components need ~~not be concerned with how~~ need not ~~be~~ worry about issues like backups & security of the data.

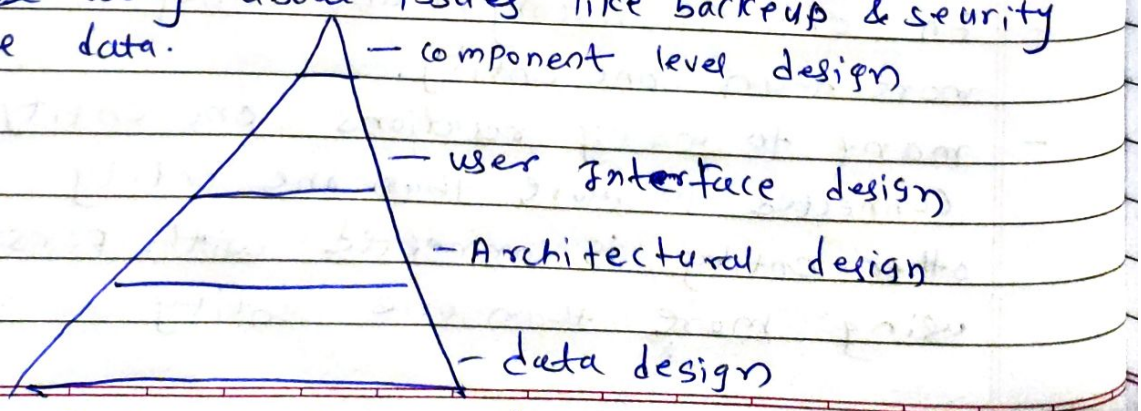


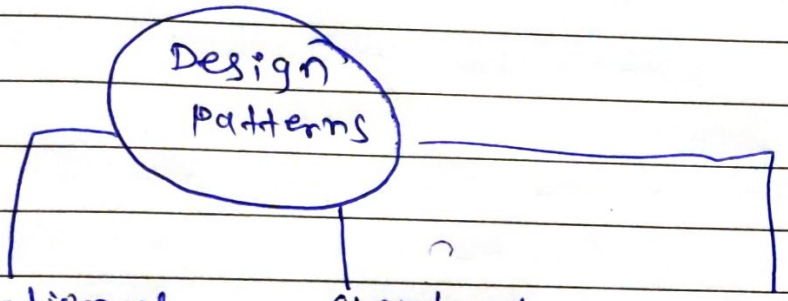
Fig:- Design modeling

pattern based software design

Page No.
Date

- In software engineering, a software design pattern is a general, reusable solution of how to solve a common problem when designing an application or system.

- Design patterns are used to support object oriented programming (OOP) is based on the concepts of both objects & classes.



creational	Structural	Behavioral
① Factoral method	① Adapter	① chain of responsability
② Abstract factory	② Bridge	② command
③ Builder	③ composite	③ Interpreter
④ prototype	④ Decorator	④ Iterator
⑤ Singleton	⑤ Facade	⑤ mediator
	⑥ Flyweight	⑥ memento
	⑦ proxy	⑦ observer
		⑧ state
		⑨ Strategy
		⑩ visitor
		⑪ Template method

① creational design patterns :-
A creational design pattern deals with object creation & initialization, providing guidance about which object are created for a given situation. These design patterns are used to increase flexibility and to reuse existing mode.

Page No.
Date

2. Structural design patterns: A structural design pattern deals with class & object composition, or how to assemble objects & classes into large structures.

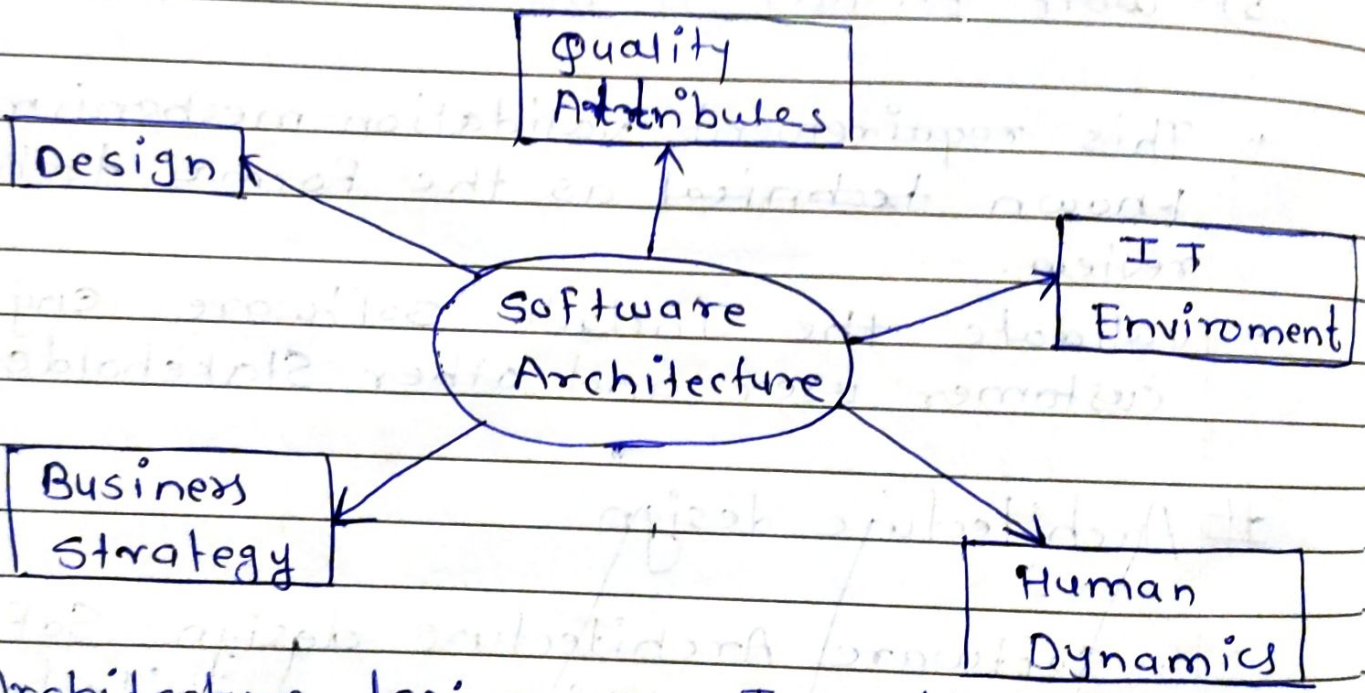
3. Behavioral design patterns: A behavioral design pattern is concerned with comm bet1 object & how responsibilities are assigned bet ween objects.

Software architecture design helps to connect the technical and operational.

- Architecture Design

- The architecture of a system describes major components, their relationships and iteract each other.

- Architecture design include several contributory factors such as Business strategy, quality attributes, human, dynamics, design and IT enviroment.



Architecture design are Two distint phases:-

1) Software Architecture:-

- Architecture serves as blueprint for a System.
- It provide abstraction manage the system complexity & establish communication & coordination machanism among components.

Page No.
Date

2) Software Design:-
Software design provide a design plan that describe element of a system.

- How they fit and work together to fulfill the requirement of the system.

- Act as a blueprint during the development process.

User interface design -

User interface is front-end application view to which user interacts in order to use the software.

The software becomes more popular if it is user interface -

1) Attractive
   Simple to use
   Responsive in short time
   clear to understand
   consistent on all interfaces screen.

There are two types of User Interface.

1) command line interfac :-

It provides a command prompt, where the user types the command & feeds to the s/m.

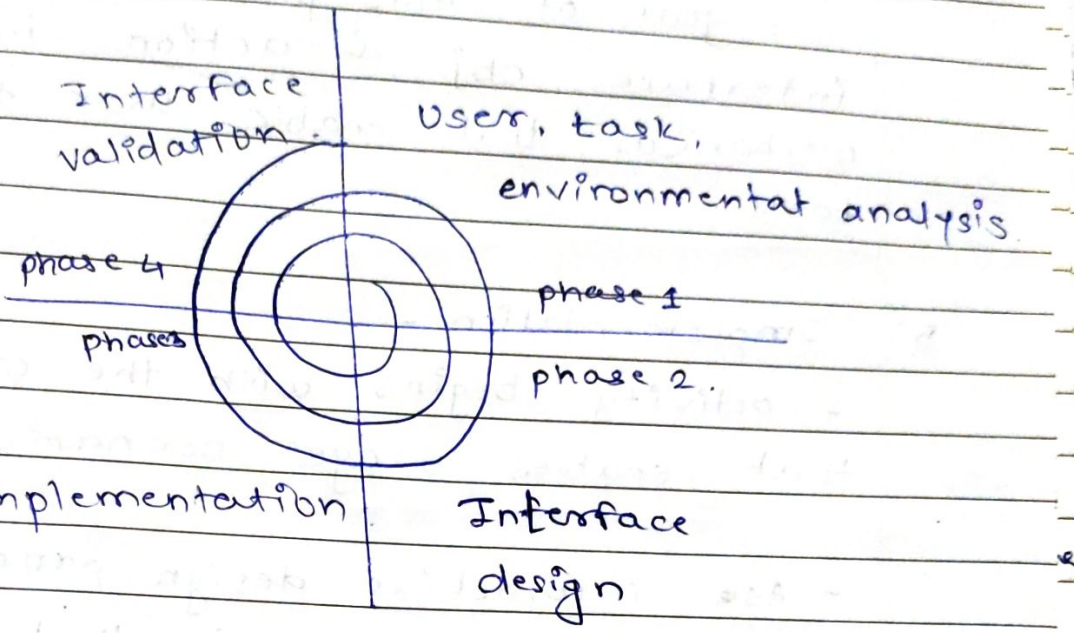
- User need to remember the syntax of the command & it use.

2) Graphical user interface :-

It ~~provive~~ provides the simple interactive interface to interact with the s/m.

GUI can be a combination of bothe h/w & s/w.

User Interface design Process -



User interface is interactive & can be represented by a spiral model.

Analysis & design process of user interface consists of four framework activities.

1) User, task, enviornmental analysis & modeling.

User who will interact with the system i.e. understanding skill & knowledges, etc. based on user's profile users are made into categories.

- From each category requirements are gathered. Based on requirements developer understand how to develop interface.
- Once all requirements are gathered a detailed analysis is conducted.

- In analysis - the task that user performs to establish goal of slm are identified, described & elaborated. It focuses on physical work.

Page No.
Date

2) Interface Design :-

goal of this phase is to define set of interface obj & action i.e. control mechanism that enable user to perform desired task.

3) Implementation -

- activity begins with the creation of prototype that enables usage scenarios to be evaluated

- As interactive design process continues a user interface toolkit ~~that allows the creation of user interface toolkit that~~ that allows creation of windows, menus, device interaction, error messages, commands & many other elements are used to complete the contructon of an interface.

4) Interface Validation -

- This phase focuses on testing the interface.

- It should be in way that should be able to perform tasks correctly & able to handle a variety of tasks.

- It should achieve all user's requirements.
- It should be easy to use & easy to learn.

UML :-

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

Date: / /

UML is not a programming language, it is rather a visual language. we use UML diagrams to portray the behavior & structure of a s/m.

- UML helps s/w engineers, businessment & s/m architects with modelling, design & analysis.

Different methods in UML

① Rumbaugh :-

The rumbaugh methodology also known as OMT (object modeling Technique) is an approach used to develop manageable object-oriented s/ms. & host ~~object~~ ~~ori~~ oop

- The purpose is to allow for class attributes, methods, inheritance & association to be easily expressed.

2) Booch methodology :- focuses on ~~oo analysi~~ object oriented analysis & design & consists of five activities :- Conceptualization, analysis, design, evolution & maintainance.

3) Jacobson methodology :-

The jacobson methodology also known as object-oriented s/w Engg. (OOSE) or even objectory, is a method used to plan, design & implement object-oriented software.

Need for standardization

- The standardization is nessasary because this process considered building block for developing product.

- Under the defined rules and protocols that are universally and adopted everyone.

- Standardization in the development of product, interperability and compatibility maintained help the development.

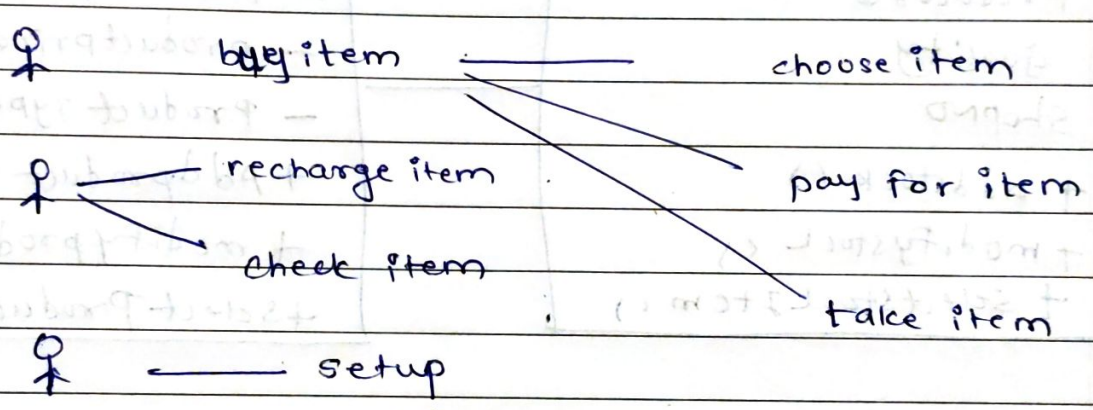
- Standardization processes are necessary to maintain product quality.

Developing diagram in UML.

Use case diagram :-

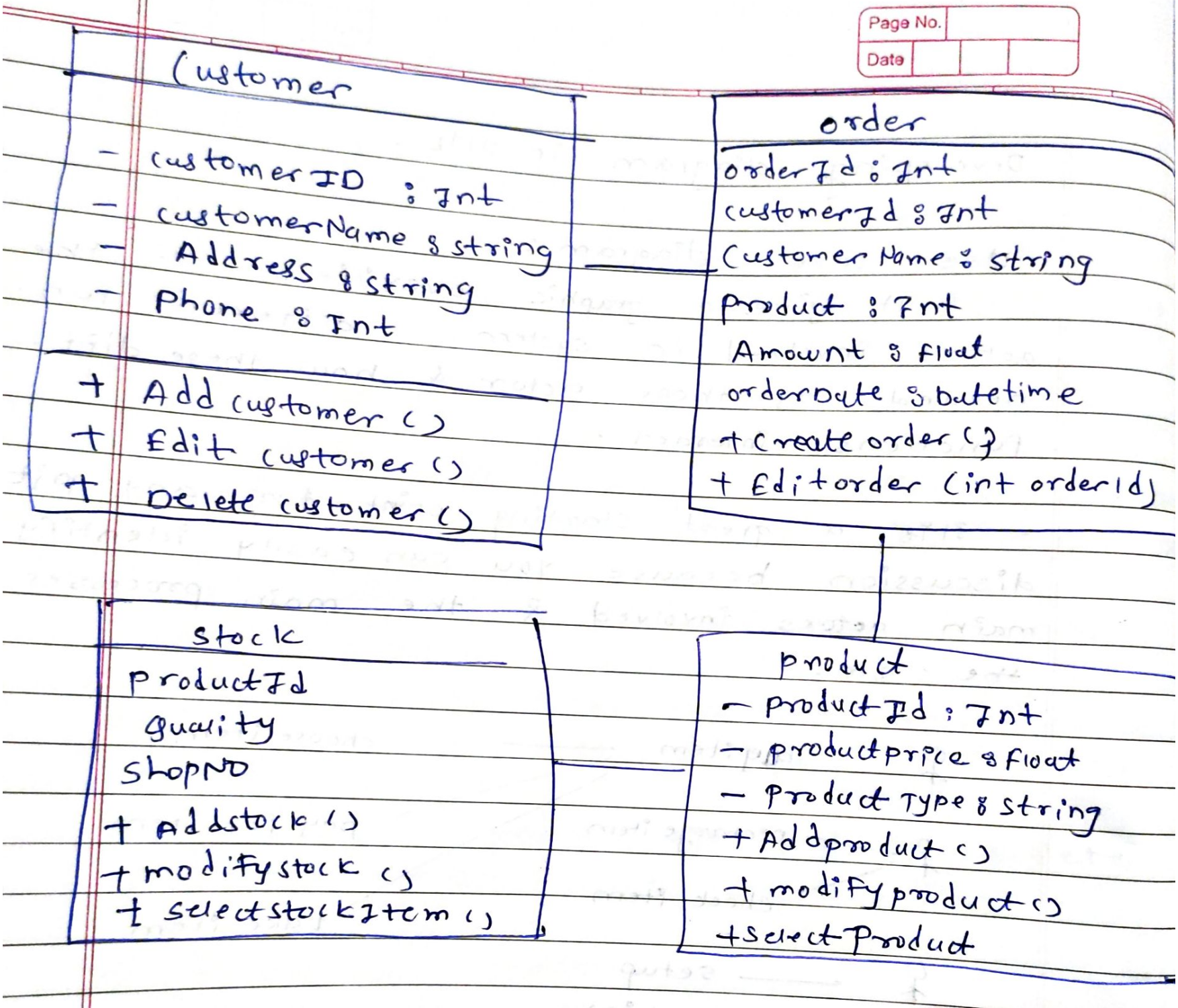
It give a graphic overview of the actors involved in system. different fun. needed by those actor & how these diff. functions interact.

- It is a great starting point for any pjt discussion because you can easily identify the main actors involved & the main processes of the s/m.



Class Diagram

- class diagrams are the main building block of any object-oriented solution.
- It shows the classes in a system, attributes & operations of each class & the relationship between each class.


Page No.
Date
Customer
- customerID : Int
- customerName : string
- Address : string
- Phone : Int
+ Add customer ()
+ Edit customer ()
+ Delete customer ()
order
orderId : Int
customerId : Int
Customer Name : string
product : Int
Amount : float
orderDate : datetime
+ create order ()
+ Editorder (int orderId)
stock
productId
quality
shopNO
+ addstock ()
+ modifystock ()
+ selectstockItem ()
product
- productId : Int
- productprice : float
- Product Type : string
+ Addproduct ()
+ modifyproduct ()
+ select Product

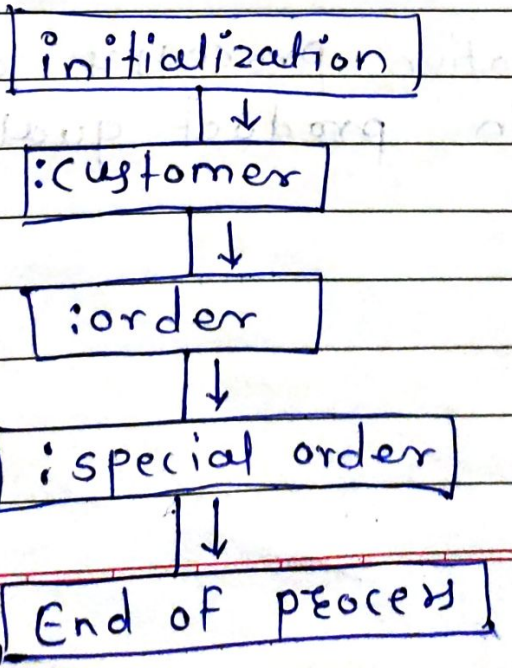
UML - interaction Diagrams :-

- from the interaction .
- It is clear that diagram used to the describe same type of interactions among the different element in the model.
- This interaction is a part of dynamic behaviour of the system.
- This interactive behaviour represented in UML by two diagrams known as sequence diagrams and collaboration diagram

purpose of interaction diagrams.

- To capture the dynamic behaviour system
- To describe the msg flow in the system.
- To describe the structure organization of the object.
- To describe interaction among objects

- Diagrams



State Diagrams :-

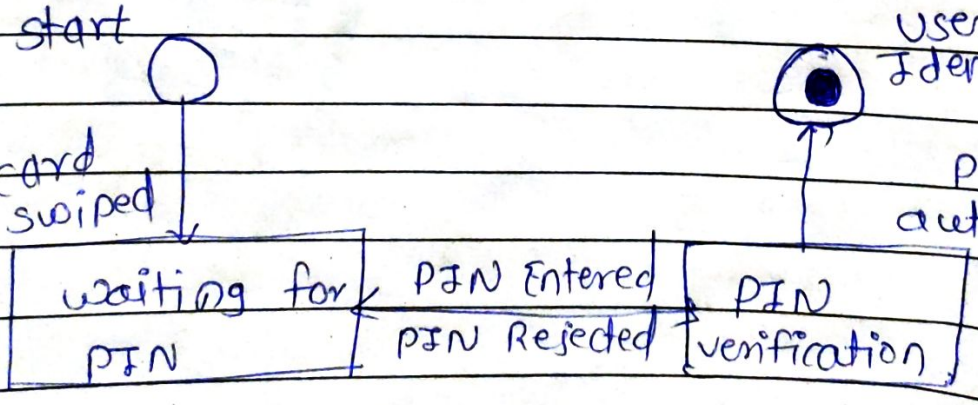
A state diagram is used to represent the condition of the sIm or part of the sIm at finite instances of time.

- it's a behavioral diagram & it represents the behaviour using finite state transitions. state diagrams are also referred to as state machines & ~~state~~-

*. There are two types of diagrams in UML :-

① Structure diagrams :- used to model the static structure of a sIm. ex :- class dia. package dia. object dia. deployment dia etc.

2) Behavior diagram :- used to model the dynam-ic. change in the sIm over time. They are used to model & construct the functionality of a sIm.



*. State Diagrams.