Assigment - 2

Q1 What is software design? Explain software measurement and metrics?

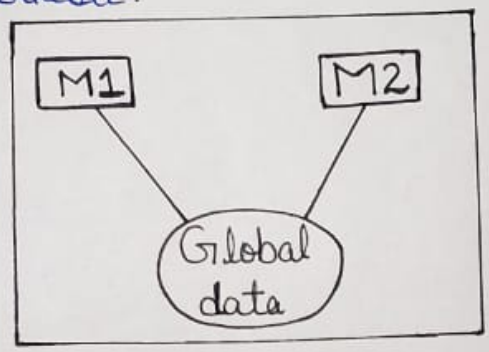
Ans. Software design is a mechanism to transform user requirement into some form, which help the programmer in software coding and implementation. It deals with representing the client's requirement, as described in SRS (Software Requirement specification) document, into a form, i.e, easily implementable using programming language. Software design phase is the first step in SDLC, which moves the concentration from the problem domain to the solution domain.

Software Measurement :- A measurement is a manifestation of the size, quantity, amount, or dimension of a particular attribute of a product or process. Software measurement is a titrate impute of a characteristic of a software product or the software process. It is an authority within software engineering. There are 2 types of software measurement:-

@ Direct Measurement :- in direct measurement, the product, process, or thing is measuredest directly using a standard scale.

ⓑ Indirect Measurement :- In direct measurement, the quality or quantity to be measured is measured using related parameters i.e. by use of referance.

(e) Common Coupling :- If two modules communicate information through some global data items, they are said to be common linked. Example :- global variable.



(f) Content Coupling :- If two modules share code, such as when two modules branch into one another, then there is content coupling between them. Same function (change in one module also affects other modules).

Software Metrics:- A metric is a measurement of the level at which any impute belongs to a system product or process. Software metrics will be useful only if they are characterized effectively and validated so that their worth is proven. There are 4 functions related to software metrics planning, Organizing, controlling and Improving. There are 3 types of software metrics:-

(a) Product Metrics :- it is used to evaluate the state of the product, tracking risks and undercover prospective problem areas. The ability of the team to control quality is evaluated. Examples include lines of code, cyclomatic complexity, code coverage, defect density and code maintainability index.

(b) Process Metrics :- it pay particular attention to enhancing the long term process of the team or organization. Examples :- variance, schedule variance etc.

(c) Project Metrics :- it describes the project characteristic and execution process. Examples :- effort estimation accuracy, schedule deviation, cost variance and productivity.

- Number of software developer.
- Staffing patterns over the life cycle of software.
- Cost and schedule.
- Productivity.

Q2 Define module coupling and Explain different types of coupling.

Ans. The degree of reliance between software modules is referred to as coupling in software engineering. Two tightly connected modules are highly reliant on one another. Loosely connected modules, however, are independent of one another. Within uncoupled modules, there is absolutely no interdependence. Types of module coupling are :-

(a) Data Coupling :- Data coupling is the transfer of data from one module to another. Ex:- Function Parameters.

(b) Stamp Coupling :- If two modules communicate via composite data items like structure, objects, etc., they are said to be stamp linked. The modules are reffered to as stamp linked when they transfer a full structural or non-global data structure to another module. For example :- passing structure variable in C or object in C++ language to a module. Mod2 access DS & algo of Mod1.

(c) Control Coupling :- If information from one module is utilised to control how instructions are executed in another module, then there is control coupling between the two units. Fuction flow b/w module set flag = 1) then only module 2 works.

(d) External Coupling :- When two modules use the same externally imposed data format, communication protocols, or device interface, this is known as external coupling. This has to do with connecting to outside equipment and gadgets. Example :- Communication protocols, ext lib.

Q3 Write short notes on the following :-

(i) Verification and Validation :-

Verification is the process of checking that software achieves its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfils the requirements that we have. Verification is simply known as Static Testing.

Validation is the process of checking whether the software product is up to the mark or in other words product has high-level requirements. It is the process of checking the validation of the product i.e. it checks what we are developing is the right product. It is a validation of actual and expected products. Validation is simply known as Dynamic Testing.

(ii) Cohesion :- in computer programming refers to how well-integrated a module's components are. Cohesion then, gauges the quality of connections between functional components inside a specific module. Functionality, for instance, and cohesiveness are closely related.

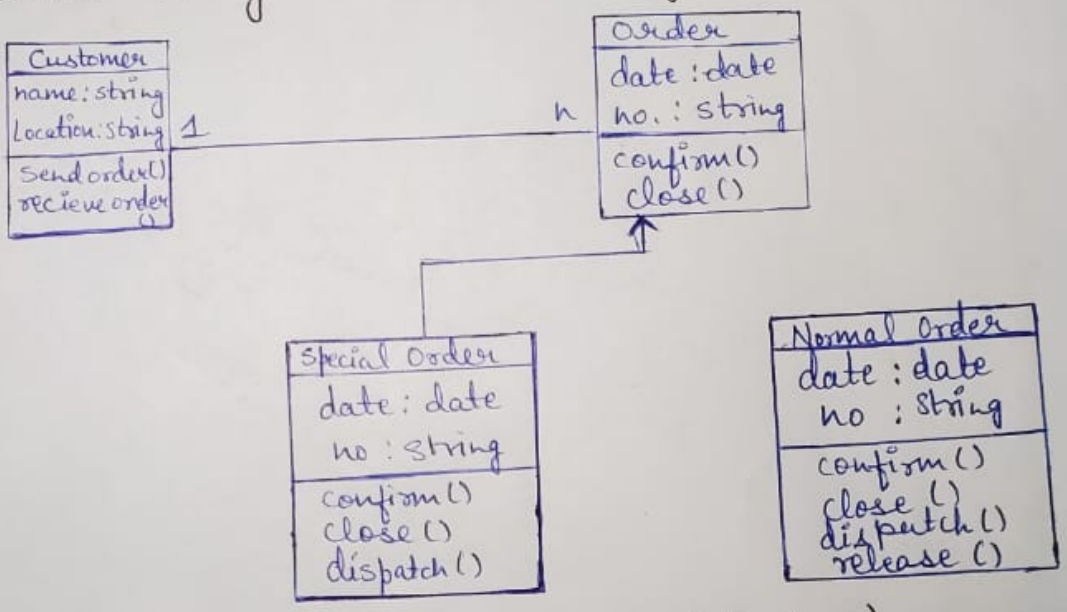
(iii) Software Requirement Engineering :- The process of gathering software needs from clients, analysing them, and then documenting them is known as requirement engineering. The goal of Software requirement engineering is to produce and maintain detailed and intricate system requirements specification papers. It consists of four steps, which include

- Feasibility Study
- Requirement Crathering
- Software Requirement Specification
- Software Requirement Validation.

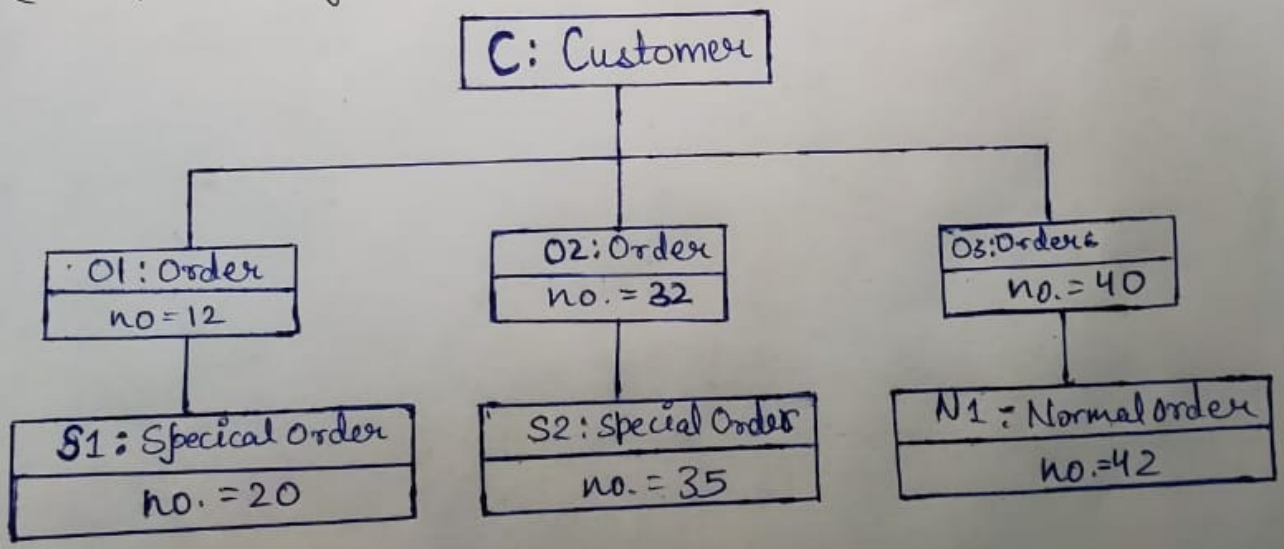
(iv) Object oriented approaches :- in the object oriented approach, the focus is on capturing the structure and behaviour of information systems into small modules that combines both data and process. The main aim of it is to improve the quality and productivity of system analysis and design by making it more usable.

Q4 Explain the different kinds of UML diagrams used with an example for each UML diagram.

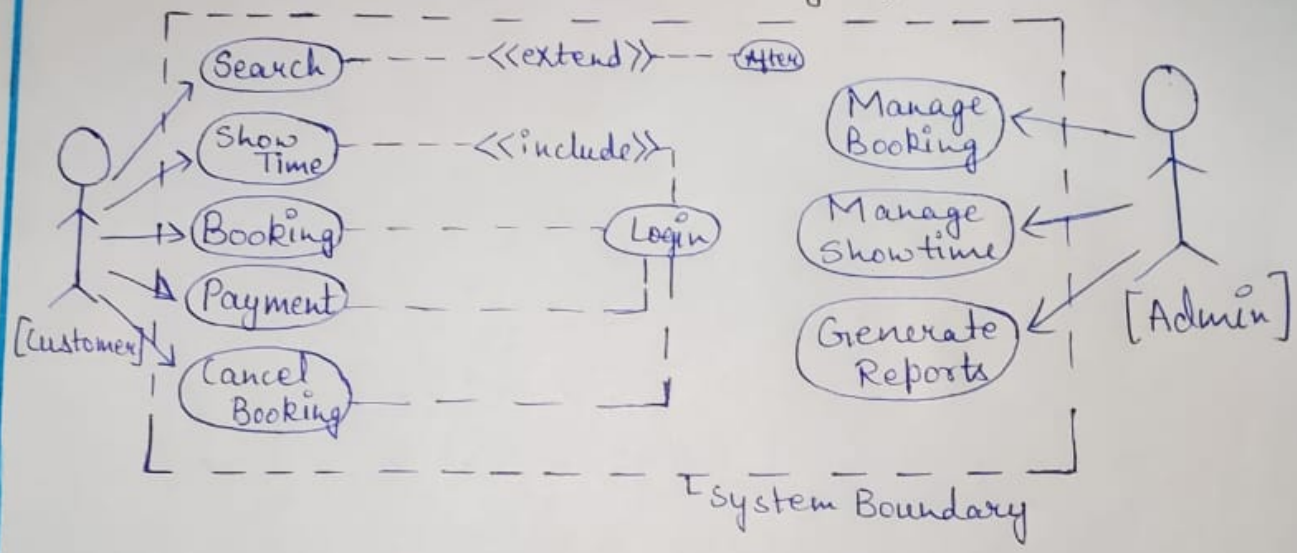
(i) Class Diagram (Order Management System)



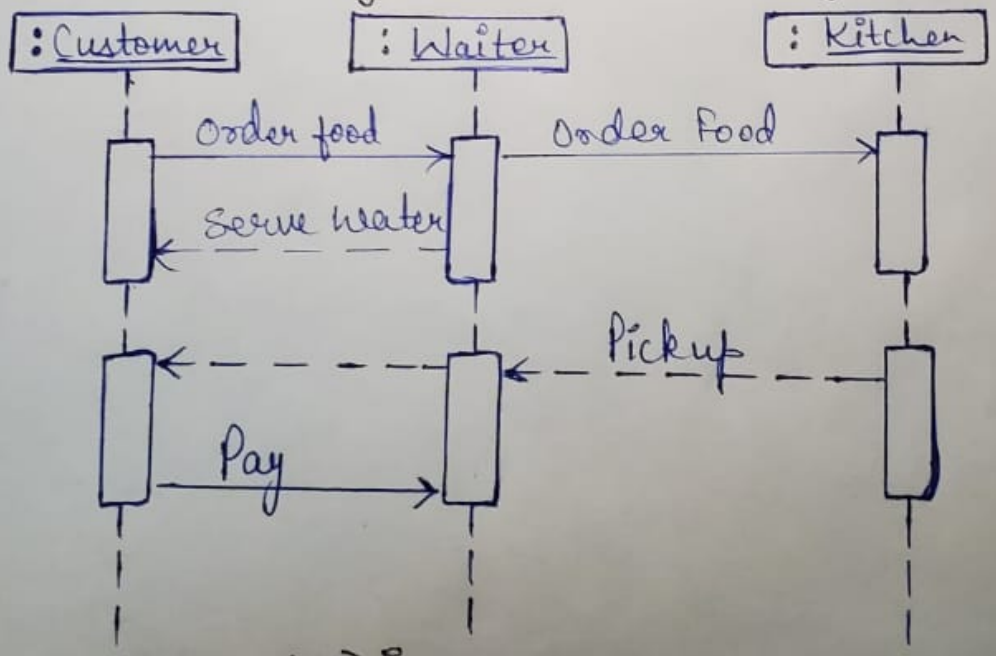
(ii) Object Diagram (Order management System)



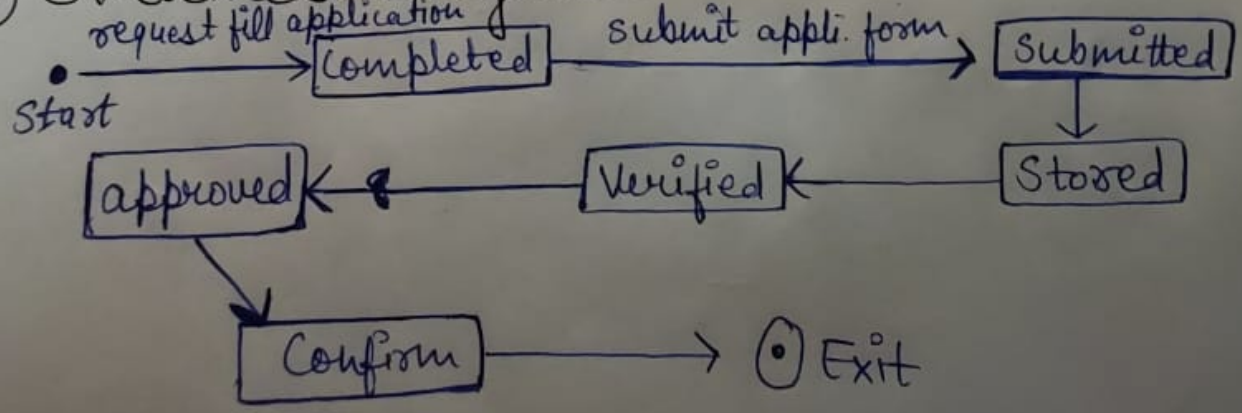
(iii) Behavioral Diagrams – USE CASE DIAGRAM
(Show ticket Booking system)



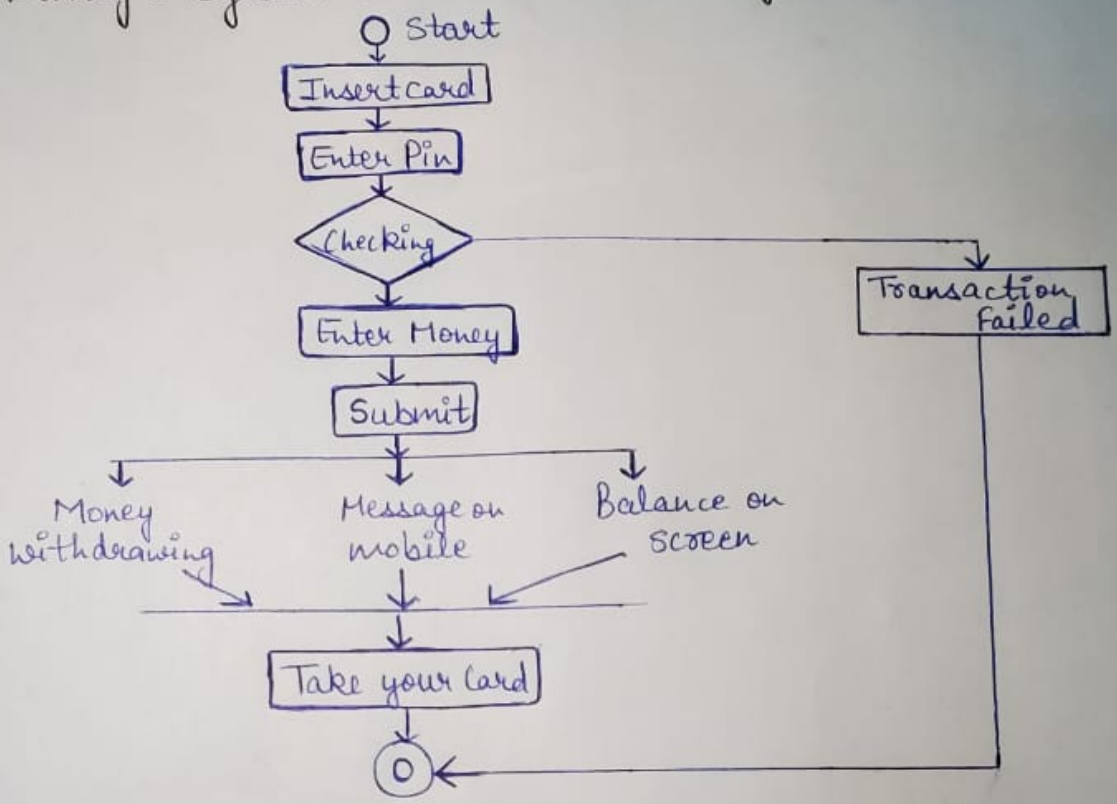
(iv) Sequence Diagram (Food Ordering System)



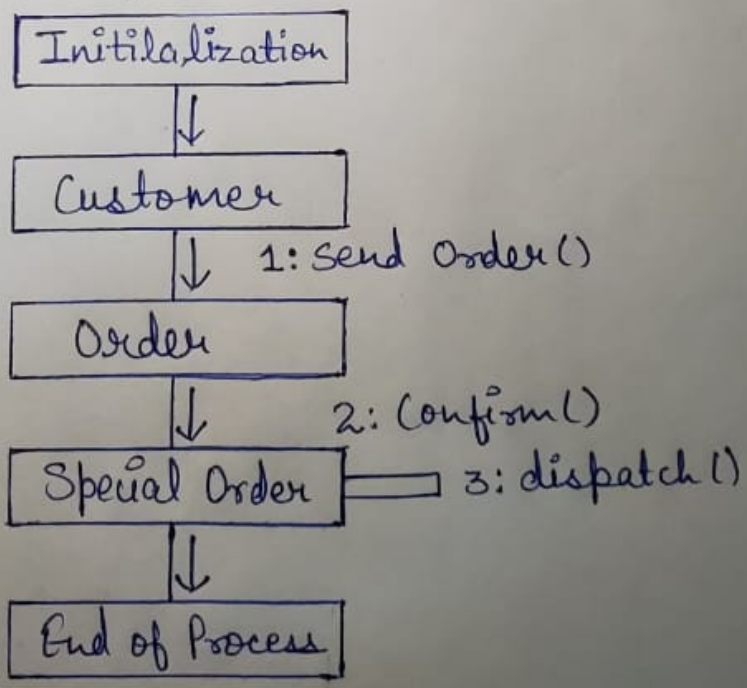
(v) Statement Diagram



(vi) Activity Diagram (Card Transaction System)



(vii) Collaboration Diagram - (Order System)



Q5 Briefly explain project monitoring and scheduling.

Ans Project-task scheduling is a significant project planning activity. It comprises deciding which functions would be taken up when. To schedule the project plan, a software manager wants to do the following:

i. Identify all the functions required to complete the project.
ii. Break down large functions into small activities.
iii. Determine the dependency among various activities.
iv. Establish the most likely size for the time duration required to complete the activities.
v. Allocate resources to activities.
vi. Plan the beginning and ending dates for different activities.
vii. Determine the critical path. A critical way is the group of activities that decide the duration of the project.