**Module - 1**

**1) What is SDLC ( Software Development Life Cycle).?**

Ans. - Software development life Cycle is a structure of development in software product and define the process is planning, implementation, testing, deployment, ongoing project maintenance, and support. It is called SDLC.

**2) What is software testing?**

Ans. - Software testing is process of completeness, Correctness and quality of developed Computer Software.

**3) What is agile methodology?**

Ans. - Agile SDLC model is combination of iterative and incremental process model with focus on adaptability and costumer satisfaction by working software product.

**4) What is SRS(Software Requirement Specification).?**

Ans.- Software Requirement Specification is complete description of behavior of the system to be developed.

**5) What is oops?**

Ans.- oops is identifying objects and assigning responsibilities to these objects.

**6) Write Basic Concepts of oops?**

Ans. - Object, Class, Encapsulation, Inheritance, Polymorphism, abstraction.

**7) What is object?**

Ans. - object is an instances of a class.

**8) What is class?**

Ans. - class is an collection of data member(variable) and member function (process, method) with its behaviors.

**9) What is Encapsulation?**

Ans.- Encapsulation is wrapping up of data into single unit i.e. private your data member and member function.

**10) What is Inheritance?**

Ans.- Properties of parent class extends into child class. Main purpose is reusability, extensibility.

# Mainly 5 types. :

1. Single

2. Multilevel

3. Hierarchical

4. Multiple

5. Hybrid

**11) What is polymorphism?**

Ans.- Polymorphism is ability to take one name having many forms.

Main 2 type:

1. Compile time (method overloading)

2. Run time (method overriding)

**12) Write SDLC phases with basic introduction?**

Ans.-

1. Requirement gathering : Establish customer needs.

2. Analysis: Model and specify the Requirements "what".

3. Design : Model and specify a solution- "Why".

4. Implementation: Construct a solution in software.

5. Testing: Validate the solution against the Requirements.

6. Maintenance : Repair defects and adapt the solution to the new Requirements.

**13) Explain Phases of the waterfall model?**

Ans. -

1. Requirement gathering : The requirements phase states what the system should do.

2. Analysis: defines the requirements of the system, independent of how these requirements will be accomplished.

3. Design : After gathering all the requirements, it's time to move on to the design stage.

4. Implementation: Once the design is finalized and approved, it's time to implement it.

5. Testing: After the developers code the design, it’s time for quality assurance. It’s important to test for all use cases to ensure a good user experience.

6. Maintenance : Repair defects and adapt the solution to the new Requirements

**14) Write phases of spiral model?**

Ans.-

1. Planning: Determination of objectives, alternatives and constraints.

2. Risk analysis: Analysis of alternatives and identification/ resolution of risks.

3. Engineering : Development of the next level product.

4. Customer Evaluation: Assessment of the results of engineering.

**15) Write agile manifesto principles?**

Ans. -

1. Individual interaction.

2. Working software.

3. Customer collaboration.

4. Responding to change.

**16) Explain working methodology of agile model and also write pros and cons?**

Ans.- It is a combination of iterative and incremental process model with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

In the Agile model, the requirements are decomposed into many small parts that can be incrementally developed. The Agile model adopts Iterative development. Each incremental part is developed over an iteration. Each iteration is intended to be small and easily manageable and can be completed within a couple of weeks only. At a time one iteration is planned, developed, and deployed to the customers. Long-term plans are not made.

**Pros**:

1.Resource requirements are minimum.

2.Suitable for fixed or changing requirements

3.Delivers early partial working solutions.

4.Good model for environments that change steadily.

5.Minimal rules, documentation easily employed.

**Cons**:

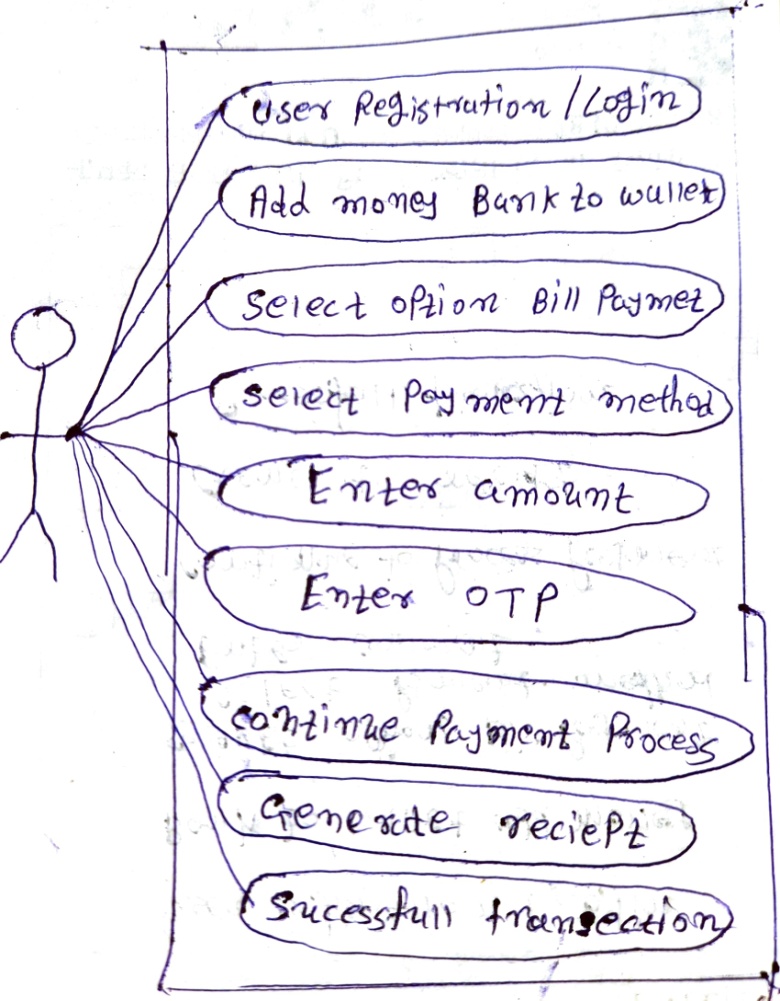
1.Not suitable for handling complex dependencies.

2.More risk of sustainability, maintainability and extensibility.

3.lack of emphasis on necessary designing and documentation.

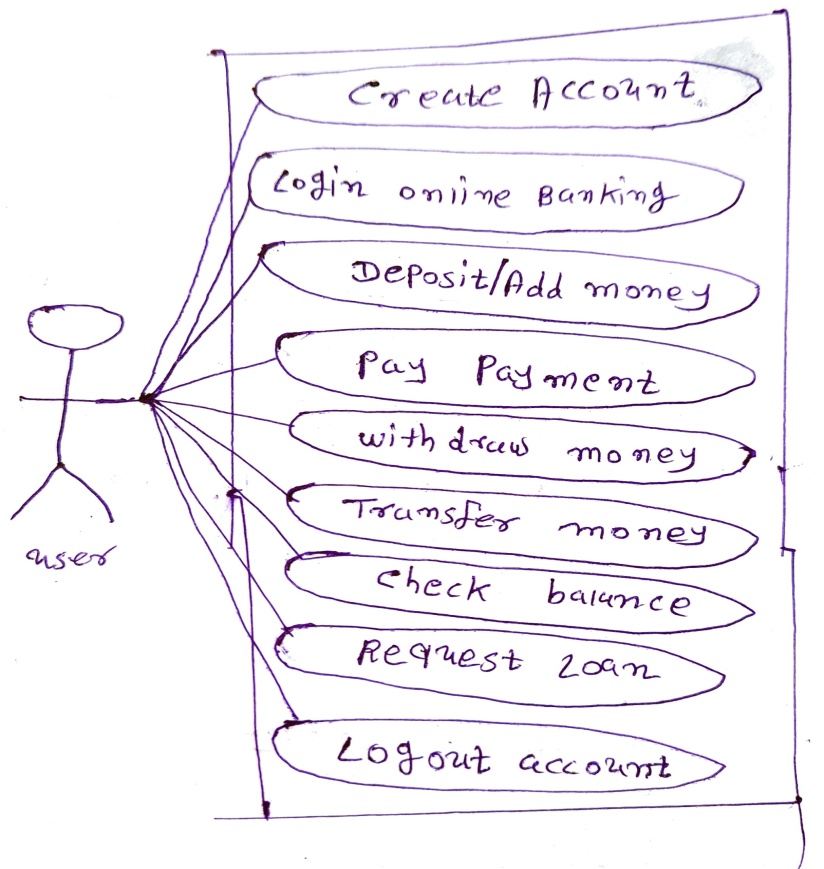
**17) Draw use case on online bill payment system (Paytm).**

Ans.-



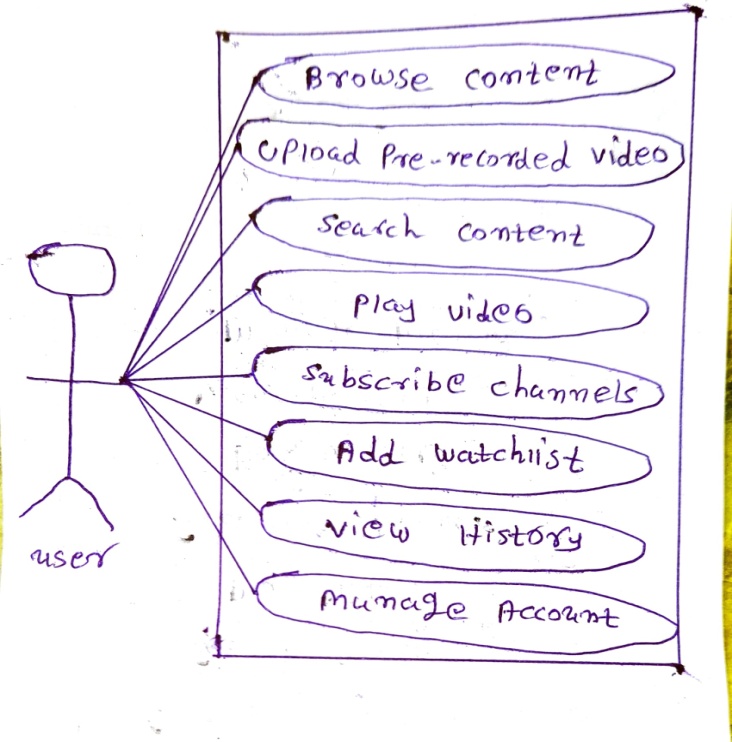
**18) Draw Usecase on banking system for customers.**

Ans.-



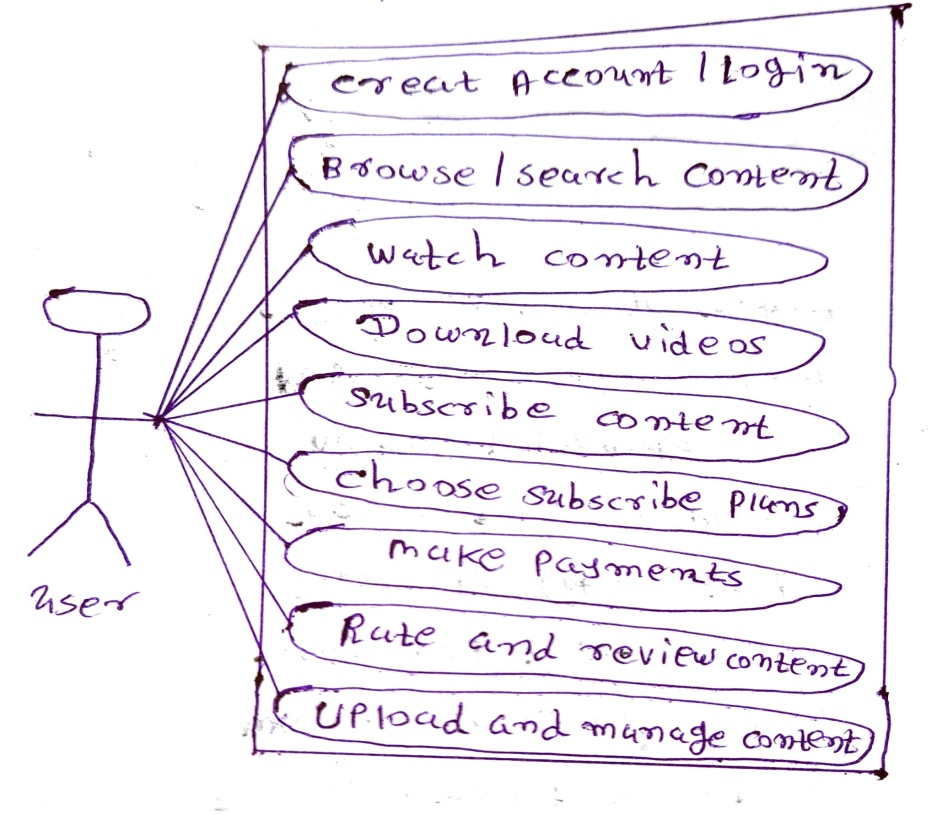
**19) Draw Usecase on Broadcasting System.**

Ans.-



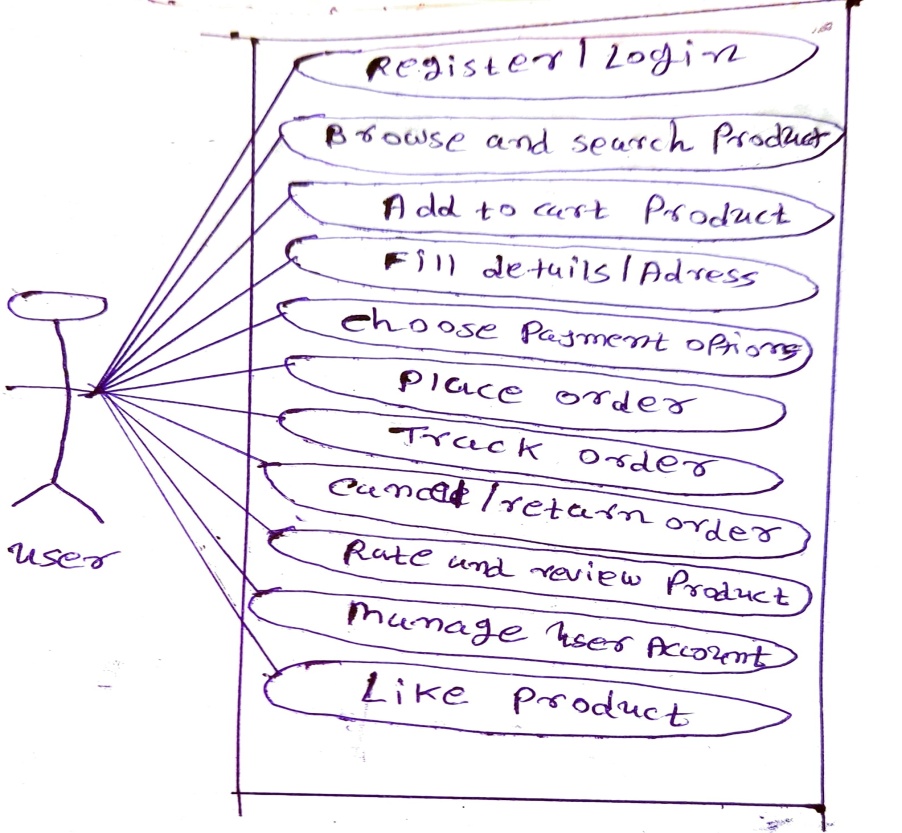
**20) Draw usecase on OTT Platform.**

Ans.-



**21) Draw usecase on E-commerce application.**

Ans.-



**22) Draw usecase on Online shopping product using payment gateway.**

Ans.-

