

Q1. What is SDLC?

A1. SDLC (Software Development Life Cycle) is a step-by-step approach to develop software product with high quality, low cost and within shortest possible time.

It is a structure imposed on the development of a software product that defines the process for Planning, Analysis, Designing, Coding, Testing and Maintenance.

A Software Development Life Cycle is essentially a series of steps, or phases, that provide a model for the development and lifecycle management of an application or software. There are a number of different development models derived from SDLC like Waterfall Model, V-Model, Bohem's Spiral Model, Iterative and Incremental Model and Agile Model.

Q2. What is software testing?

A2. Software testing is a process of validating and verifying that software product meets the customer requirements and functions as expected.

It can also be stated as a process to identify correctness, completeness and quality of the developed software product.

Q3. What is Agile methodology?

A3. Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. It prioritizes flexibility, collaboration, and customer satisfaction.

Q4. What is SRS?

A4. Software Requirement Specification is a document which has complete description of the system to be developed. It is generally prepared in the Analysis Phase of Software Development.

It contains functional and non-functional requirements that the software must satisfy.

It also includes a set of use cases that describe all of the interactions that the users will have with the system.

Recommended approaches for the specification of software requirements are described by IEEE 830-1998.

This standard describes possible structures, desirable contents, and quality of a SRS document.

Q5. What is OOPS?

A5. Object-Oriented Programming System or OOPs refers to programming languages that make use of objects. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function. It is used to structure the software program into simple reusable code.

Q6. Write Basic Concepts of oops

A6. OOPs Concepts are:

- Class
- Object
- Encapsulation
- Abstraction
- Polymorphism
- Inheritance

Q7. What is object

A7. An object is the basic unit of OOP which is accessed by its properties called data members & member function. It creates the memory for the class.

Q8. What is class

A8. Class is a collection of data members and member functions. It is a blueprint or template to access properties and functions of the objects.

Q9. What is encapsulation

A9. A wrapping up of data and functions into a single unit (i.e Class) is called Encapsulation. Encapsulation enables data hiding by declaring data members and member functions of a class as private.

Q10. What is inheritance

A10. Inheritance means that one class (Child Class, Sub Class, Derived Class) inherits the characteristics of another class (Parent Class, Super Class, Base Class).

Types of Inheritance:

- Single Inheritance
- Multilevel Inheritance
- Hierarchical Inheritance
- Hybrid Inheritance
- Multiple Inheritance

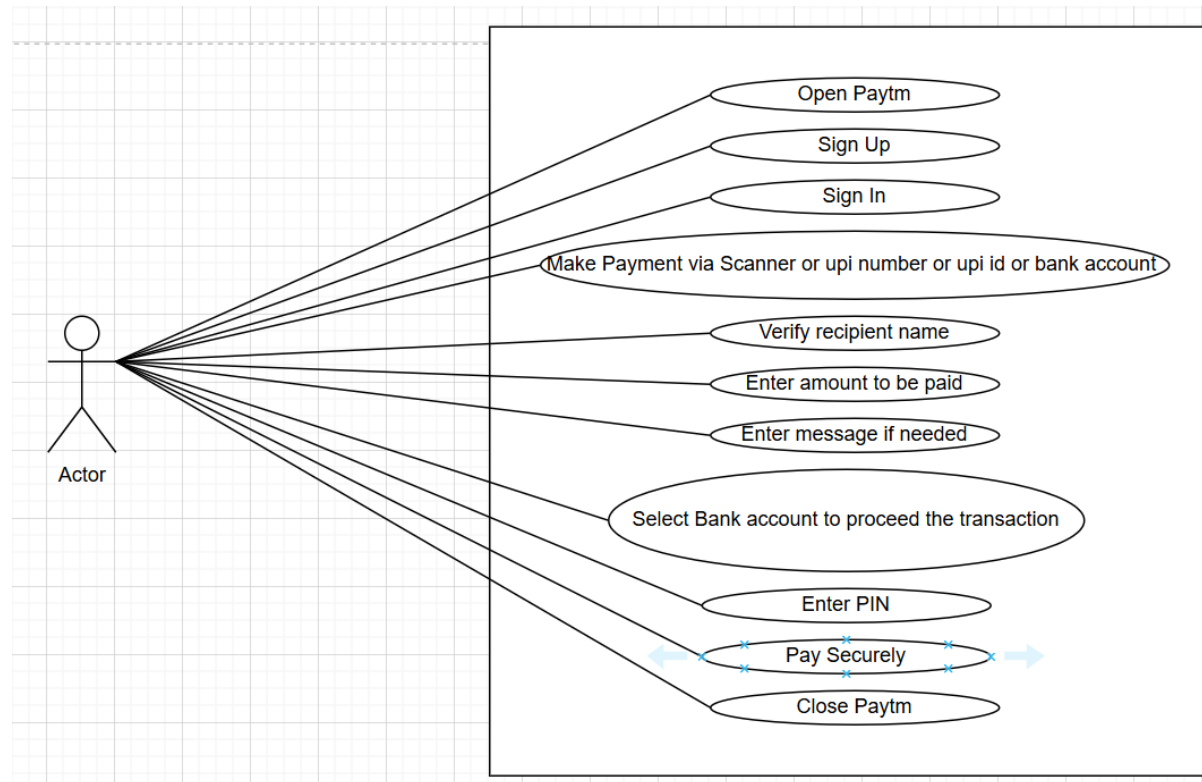
Q11. What is Polymorphism

A11. An ability to take one name having many different forms. Following are two types of Polymorphism

- Compile time Polymorphism: (Operator Overloading)
 - Method name should be same in single class but its behaviour (Arguments & Data type) is different.
- Run time Polymorphism (Operator Overriding)
 - Method should be same in super class and sub class but its behaviour is different.

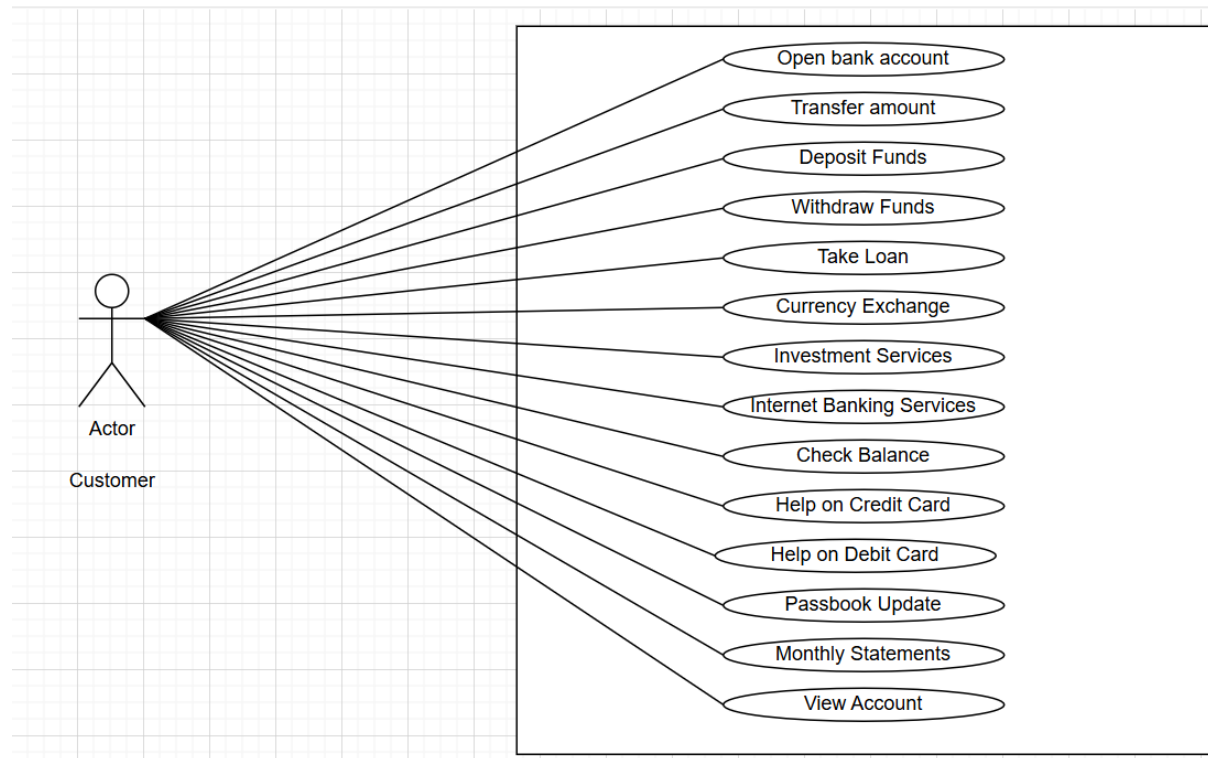
Q12. Draw Usecase on online bill payment system (paytm)

A12.



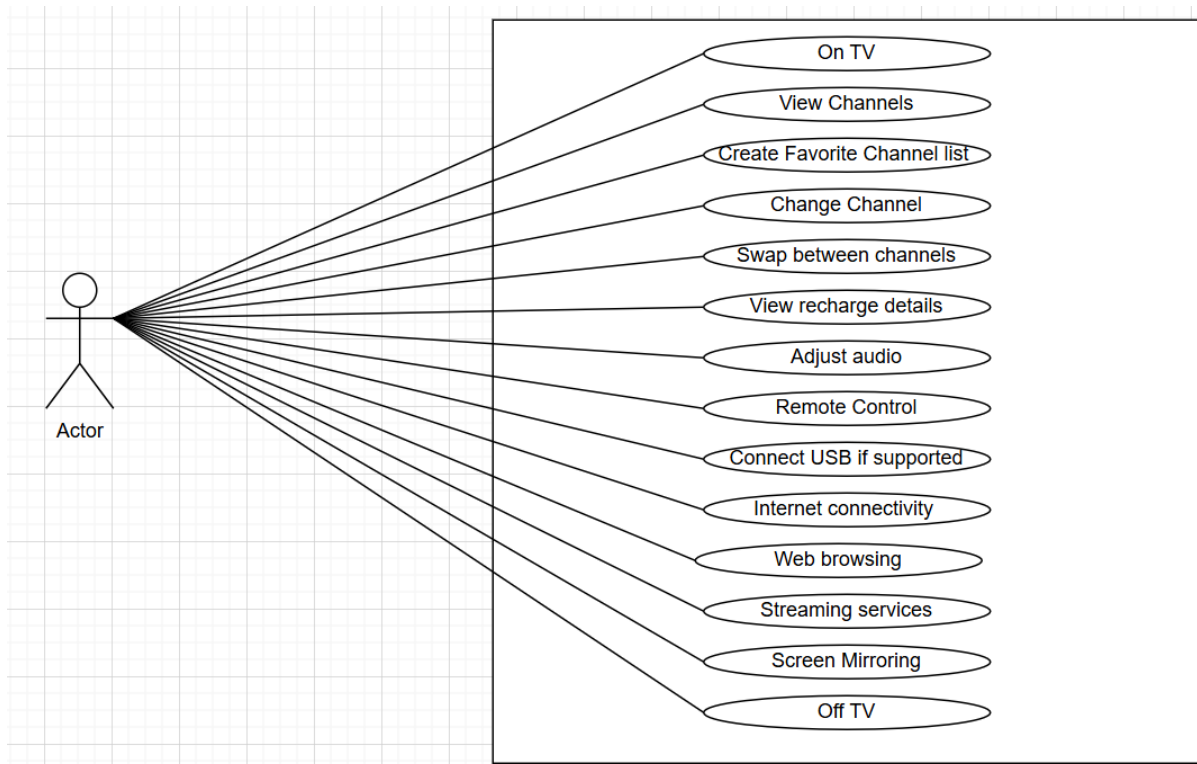
Q13. Draw Usecase on banking system for customers

A13.



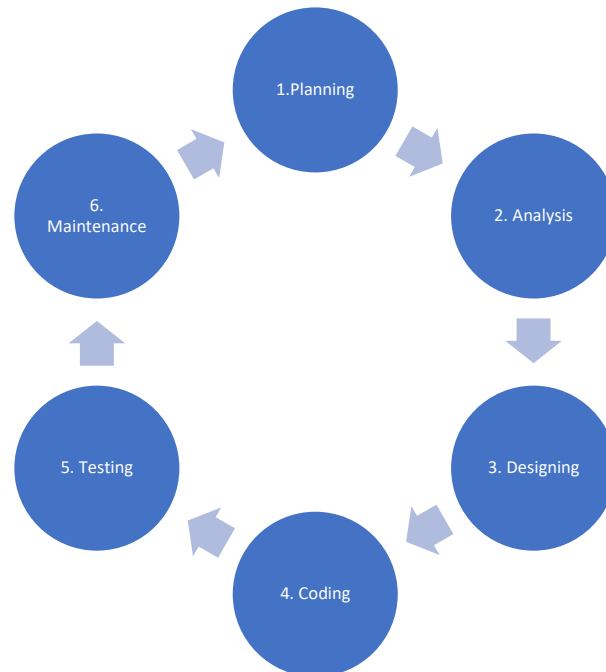
Q14. Draw Use case on Broadcasting system

A14. For Broadcasting system below is the use case shown for Smart TV



Q15. Write SDLC phases with basic introduction

A15.



SDLC Phases consists of Planning, Analysis, Designing, Coding/Implementation, Testing and Maintenance as shown in above diagram.

Planning Phase:

- This is the initial phase of SDLC to determine the project's objectives and to identify the business and technical requirements that the project must meet.
- Requirement definitions usually consist of natural language, diagrams and tables.
- Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.
- Following problems may arise due to this:
 1. Lack of Clarity
 2. Requirement confusion between functional and non-functional requirements
 3. Requirement Amalgamation
- To avoid these problems validation and constant feedback are needed throughout the software lifecycle, not only when the "final system" is delivered.

Analysis Phase:

- The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
- The deliverable result at the end of this phase is a SRS (Software Requirement Specification) document.
- Ideally, this document states in a clear and precise fashion what is to be built.

Designing Phase:

- In this phase overall system architecture is designed using the SRS document.

- Details on computer programming languages and environments, machines, packages, application architecture, distributed architecture layering, memory size, platform, algorithms, data structures, global type definitions, interfaces, and many other engineering details are established in this phase.
- The deliverable result at the end of this phase is DAD (Design Architecture Document) document.
- Test Plans are created based on this document.
- Ideally, this document represents the “how” phase.

Coding/Implementation Phase:

- In this phase software product is itself developed using the SRS document and DAD document.
- The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.
- The end deliverable is the software product itself.

Testing Phase:

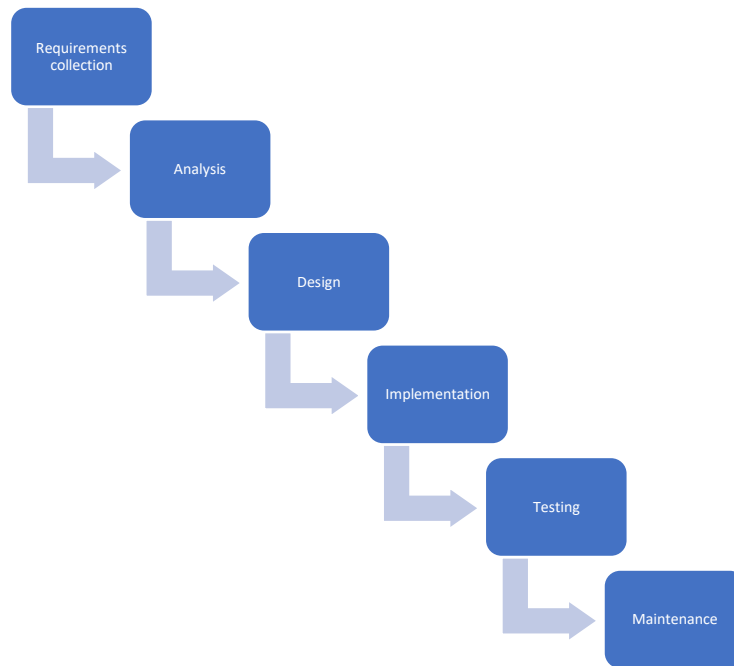
- This is the critical part of SDLC, that ensures the developed software product is functioning as expected and that all customer requirements are met.
- The testing phase is a separate phase which is performed by a different team after the implementation is completed.
- Main objective of this phase is to identify defects in the developed software and work with developers to fix them.
- This phase gives information on the level of quality of the developed software product.
- Once testing phase is done, software is ready for deployment.

Maintenance Phase:

- This is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.
- It can also be stated as a process of changing a system after it has been deployed.
- The developing organization or team will have some mechanism to document and track defects and deficiencies.
- Types of Maintenance are as follows:
 1. Corrective maintenance: identifying and repairing defects
 2. Adaptive maintenance: adapting the existing solution to the new platforms.
 3. Perfective Maintenance: implementing the new requirements

Q16. Explain Phases of the waterfall model

A16.



Waterfall Model was the first SDLC model to be used widely in Software Engineering. It is also referred to as a linear – sequential life cycle model. In this model it is important to complete one phase at a time in order to go to the next phase, not possible to move back to previous phase. This Model is suitable for projects with well-defined requirements, stable project goals and non-dynamic technology. However, it has limitations such as difficulty in accommodating changes and late defect detection

Requirements Gathering:

- All the customer requirements of the software to be developed are captured in this phase.
- Requirements must be clear and unambiguous. They must be fixed as they cannot change throughout the lifecycle.

Analysis:

- Captured requirements from the previous phase are documented in SRS document.
- Documentation of requirements must be very clear and well documented.

Design:

- Overall system architecture is designed using the SRS document from the previous phase.
- This involves creating a detailed design document that outlines the software architecture, user interface, and system components.
- This is documented in Design Architecture Document.

Implementation:

- Software is developed with reference to SRS and Design Architecture Document from previous phases.
- Each module is developed and tested for its functionality which is referred to as Unit Testing done by the development team.

Testing:

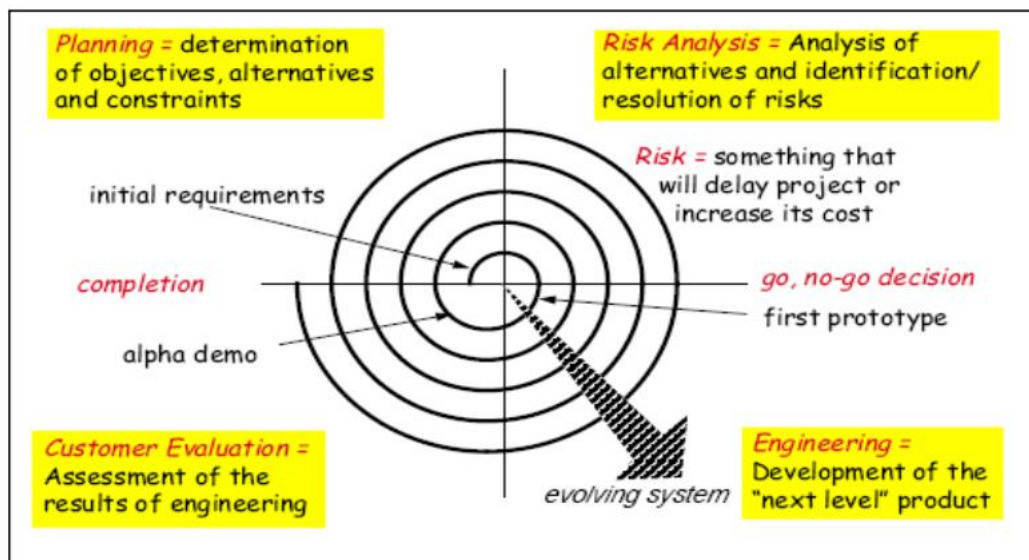
- Developed software product is integrated to perform integration testing and system level testing.
- In the testing phase, the software is tested as a whole to ensure that it meets the requirements and is free from defects.
- Results must be well documented.
- Once the software has been tested and approved, it is deployed to the production environment.

Maintenance:

- The final phase is maintenance, which involves fixing any issues that arise after the software has been deployed and ensuring that it continues to meet the requirements over time.
- Maintenance can be corrective, perfective, or adaptive.

Q17. Write Phases of the Spiral model

A17.



Spiral Model is very widely used in the software industry as it is in synch with the natural development process of any product learning with maturity also involves minimum risk for the customer as well as the development firms. It is a combination of the waterfall model and the iterative model. This Model is suitable for medium to high risk projects where requirements are complex and needs evaluation to get clarity. However, it has limitations such as project timeline estimation is difficult as the number of phases is unknown at the start of the project.

Planning Phase:

- Each cycle of spiral begins with Planning Phase.

- Requirements are gathered from the customers and the objectives are identified, elaborated, and analysed at the start of every phase.
- This phase involves determination of objectives, identifying alternative solutions and constraints of the portion of the product.

Risk Analysis Phase:

- This step is to evaluate the risk involved with alternatives and constraints identified from the previous phase to select the best possible solution.
- At the end of this quadrant, the Prototype is built for the best possible solution.

Engineering Phase:

- During the third quadrant, the identified features are developed and verified through testing.
- At the end of the third quadrant, the next version of the software is available.

Customer Evaluation Phase:

- In this phase, assessment of the results of engineering is done.
- In the fourth quadrant, the customers evaluate the so-far developed version of the software.
- In this phase, the software is evaluated to determine if it meets the customer's requirements and if it is of high quality.
- The next iteration of the spiral begins with a new planning phase, based on the results of the evaluation.

Q18. Write agile manifesto principles

A18. The 4 values of Agile Manifesto Principles are as follows:

- Individuals and interactions over processes and tools
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

Q19. Explain working methodology of agile model and also write pros and cons.

A19. Working Methodology of Agile model:

- Agile SDLC model is a combination of iterative and incremental model with focus on process adaptability and customer satisfaction by rapid delivery of working software product.
- Agile Methods break the product into small incremental builds.
- Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.
- Each iteration typically lasts from about one to three weeks.
- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- At the end of the iteration a working product is displayed to the customer and important stakeholders.

Pros of Agile model:

- Very realistic approach
- Rapid delivery.
- Functionality can be developed rapidly
- Resource requirements are minimum.
- Little or no planning required

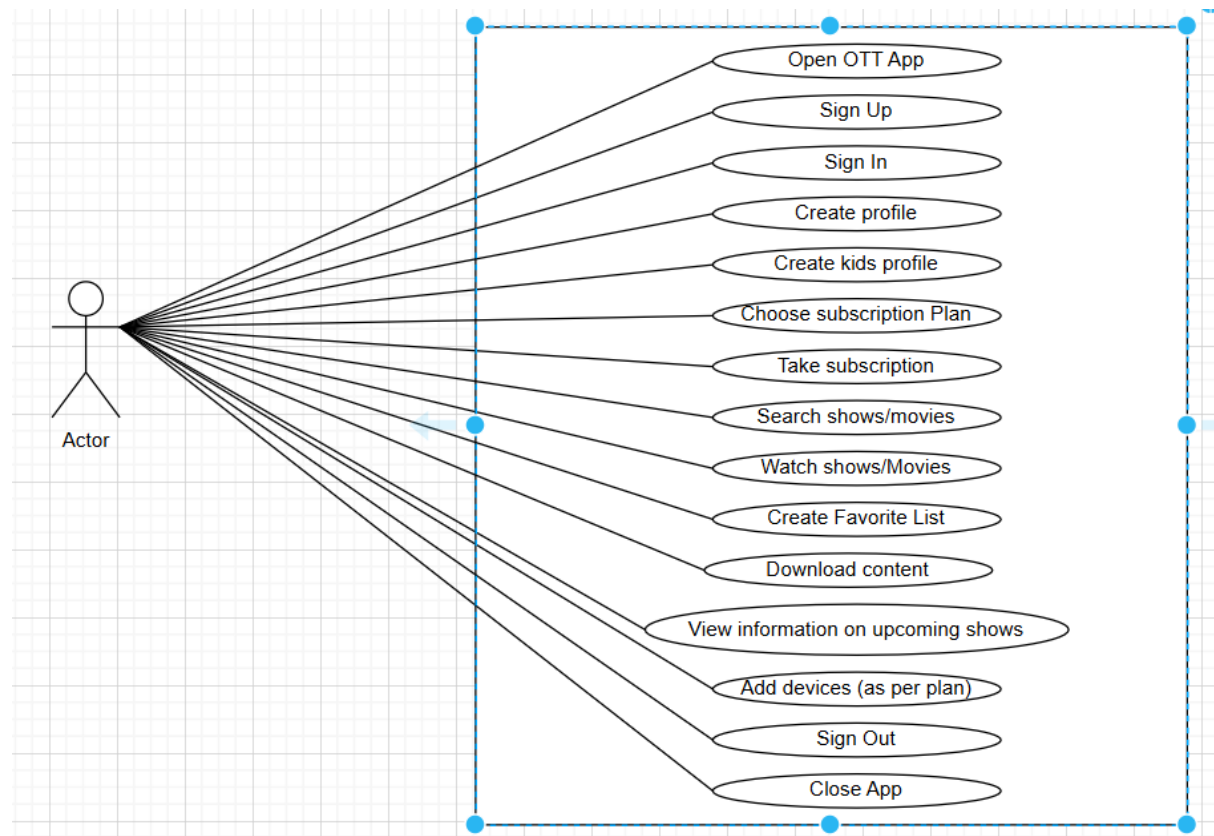
- Promotes teamwork and cross training.
- Suitable for fixed or changing requirements
- Gives flexibility to developers

Cons of Agile model:

- More risk of sustainability, maintainability and extensibility.
- Depends heavily on customer interactions.
- Very high individual dependency.
- Minimum documentation generated.
- Not useful for small projects.
- Not suitable for handling complex dependencies.

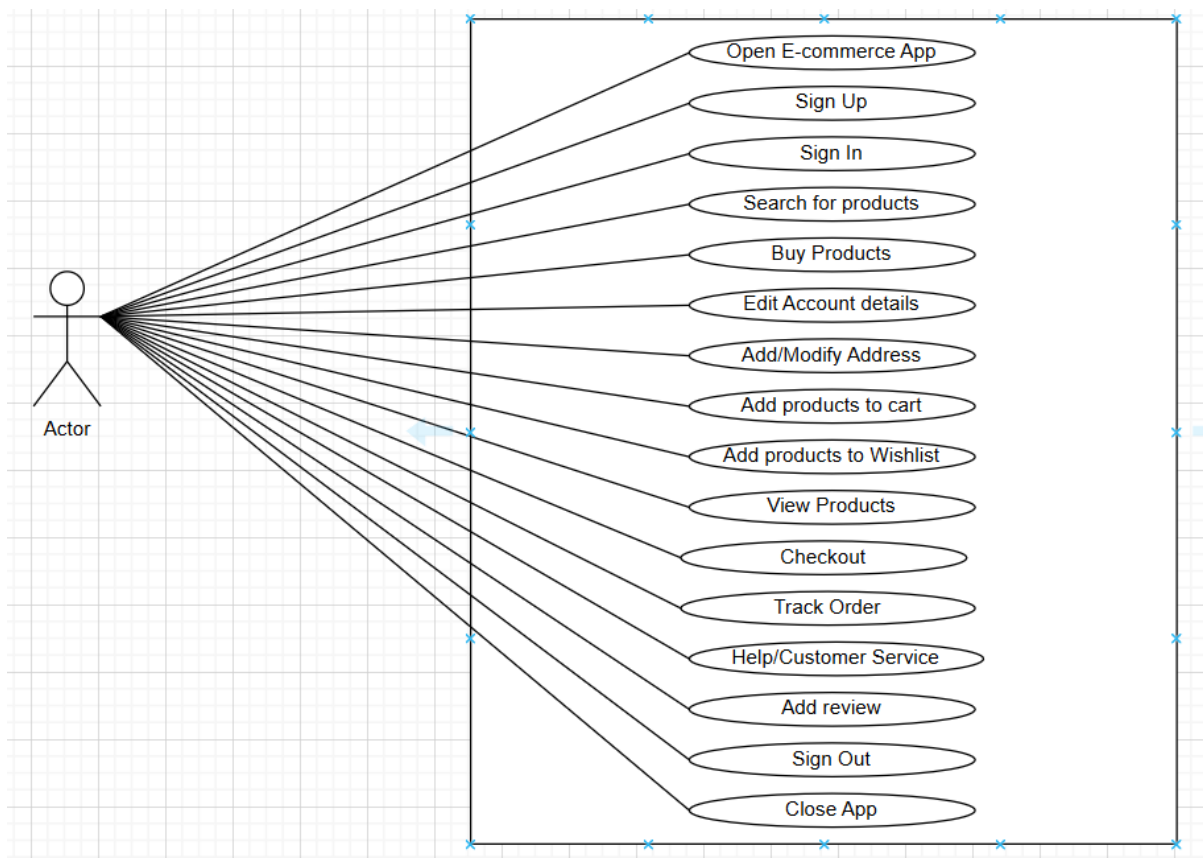
Q20. Draw usecase on OTT Platform.

A20.



Q21. Draw usecase on E-commerce application.

A21.



Q22. Draw usecase on Online shopping product using payment gateway..

A22.

