

Software Testing Assignment-1

Modual-1 (Fundamental)

(1) What is SDLC

SDLC is a step by step approach to develop any software with height quality, low-cost in the shortest possible time.



(2) what is software testing

It is kind of process to verify and validate the developed software.

Software testing is a process, which is

Used to identify correctness, completeness, quality of the developed software.

(3) What is SRS

It is software Requirement Specification is a complete description of the behavior of the system to be developed.

There are three types of requirements

- (1) Customer requirement
- (2) Function requirement
- (3) Non-function requirement

(4) Write SDLC phases with basic introduction

There are six phases work in SDLC

(1)Planning (requirement) :

The planning phase is the foundation of the SDLC. In this phase, the project's goals, scope, resources, timeline, and budget are determined.

(2) Analysis :

This phase is follow the software requirement specification (SRS) and functional and non-functional documentation requirements .

(3) Designing :

The architecture and design of the software is created.

(A) Implementation :

In the implementation phase, the team builds the components either from scratch or by composition.

(B) Testing :

Performance and important feedback is generated and defects are revealed.

(5) Maintenance :

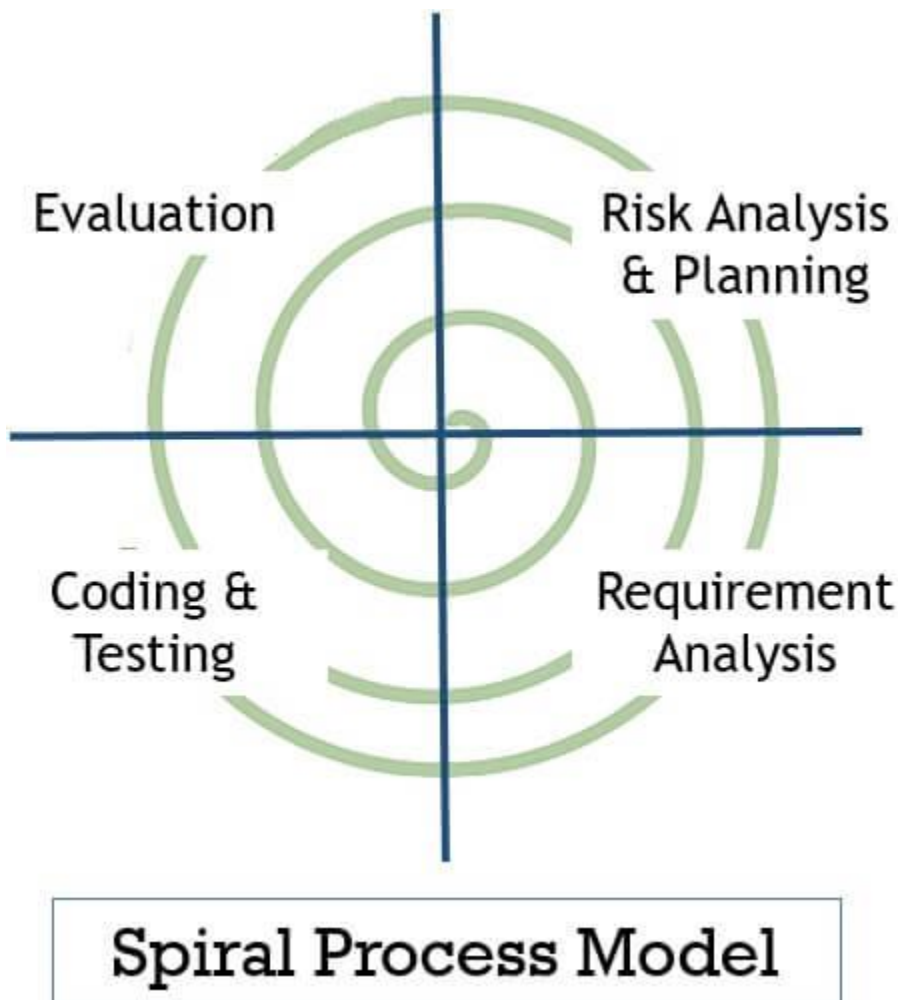
Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software release, as well as fixing defects.

In this three types of maintenance,

- (1) Corrective maintenance
- (2) Adapting maintenance
- (3) Perfective maintenance

(5) Write phases of spiral model

Spiral model is very widely used in the software industry as it is in sync 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 four phases of spiral model

(A) Planning

Determination of objective, alternatives and constraints .

(B) Risk Analysis

Analysis of alternative and identification and resolution of risk .

(C) Customer Evaluation

Assessment of the results of engineering .

(D) Engineering

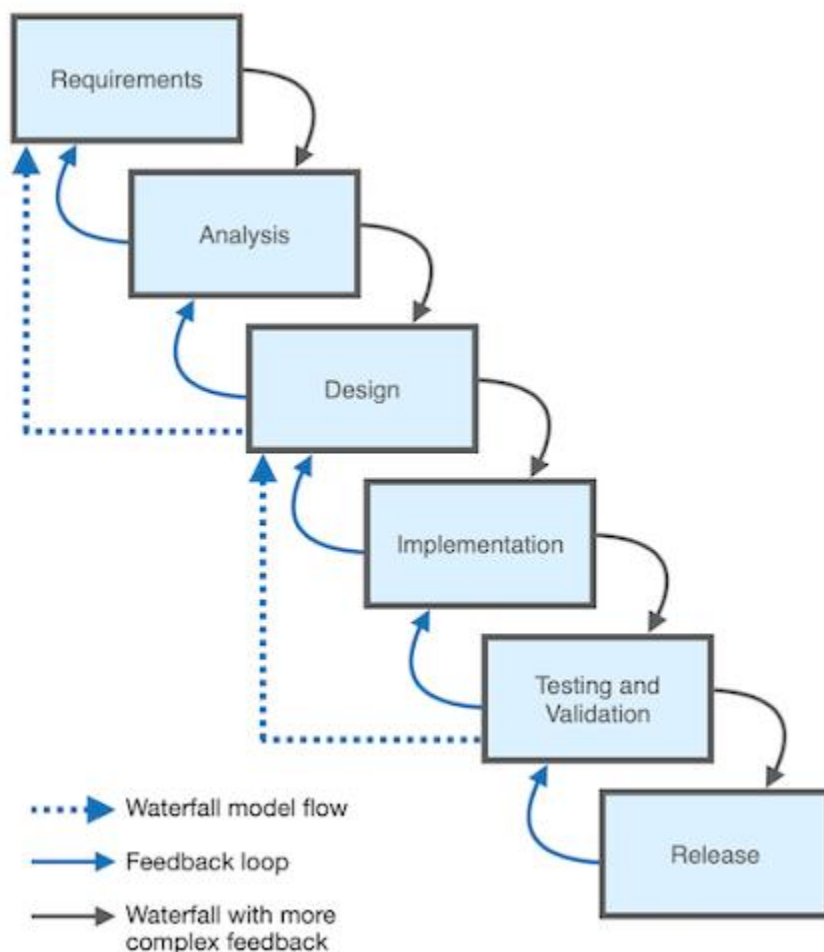
Development of the next level product .

(E) Customer Evaluation

Assessment of the results of engineering.

(6) Explain Phases of the waterfall mode

The waterfall model is a classical software lifecycle That models the software development as a step-by-step waterfall between the various development.



- Six phases of waterfall model

- (A) Requirements collection

- In this phases requirement of the well clear documents and clear information of in a software .

- (B) Analysis

- In this phase a project team analyze requirement And need of software and users.

- (C) Design

- Create design specifications based on software requirements.

- (D) Implementation

- Developers write and implements code according To software requirements .

- (E) Testing

- Test the developed software for defects and bugs.

- (F) Maintenance

- The final phase in maintenance in the product is Updated and released to the users.

(7) What is class

Class is a collection of a data member and member function with its behavior.

(8) What is oops

Object oriented programming is viewed as a collection of object.

It is used to structure the software program into simple reusable code. Here it is referred as functional testing or black box testing.

(9) Write Basic Concepts of oops

Oops in a basic six concepts here

(A) Class

Class is collection of a data member and member function with its behavior

(B) Object

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

(c) Encapsulation

Encapsulation is the process of **hiding the internal details** of an object and only exposing necessary information.

(D) Abstraction

Abstraction is the representation of the essential features of an object. Also called data hiding.

(E) Polymorphism

Polymorphism allows methods to **perform differently based on the object that invokes them.**

(a) Compile time polymorphism

(b) Run time polymorphism

(F) Inheritance

One class inherits the properties of another class.

(10) What is Inheritance

One class inheritance the properties of another class.

- There are five types of inheritance

(A) Single inheritance

(B) Multiple inheritance

(C) Hierarchical inheritance

(D) Hybrid inheritance

(E) Multilevel inheritance

(11) What is Encapsulation

A wrapping up of data and function in to a single unit is called encapsulation. It hide private access of data members and member function.

(12) What is object

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

(13) what is polymorphism

An ability to take one name having many different forms.

- There are two types polymorphism
 - (A) Compile time polymorphism
 - (B) Run time polymorphism

(14) What is agile methodology?

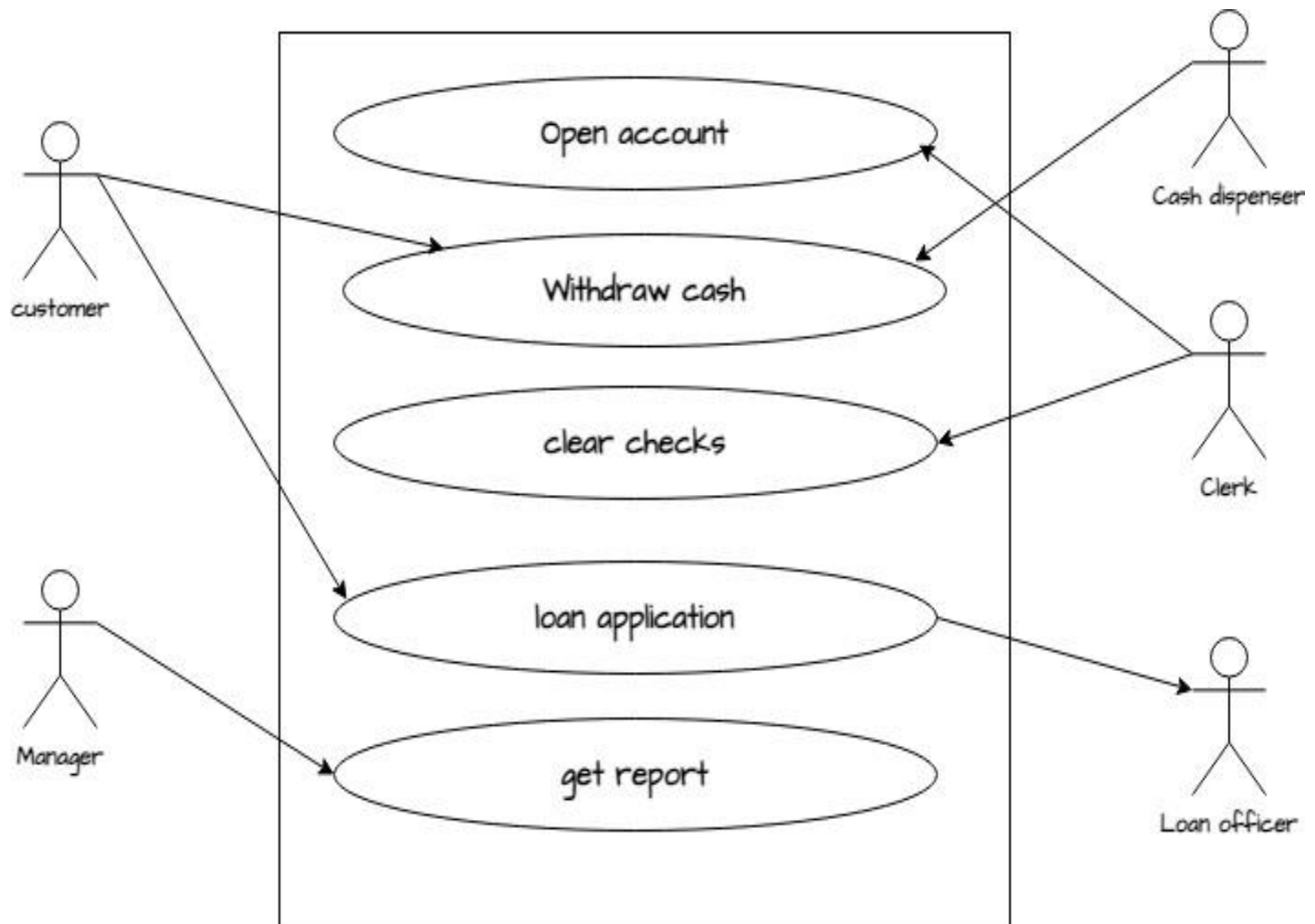
Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

(15) Agile manifesto principles

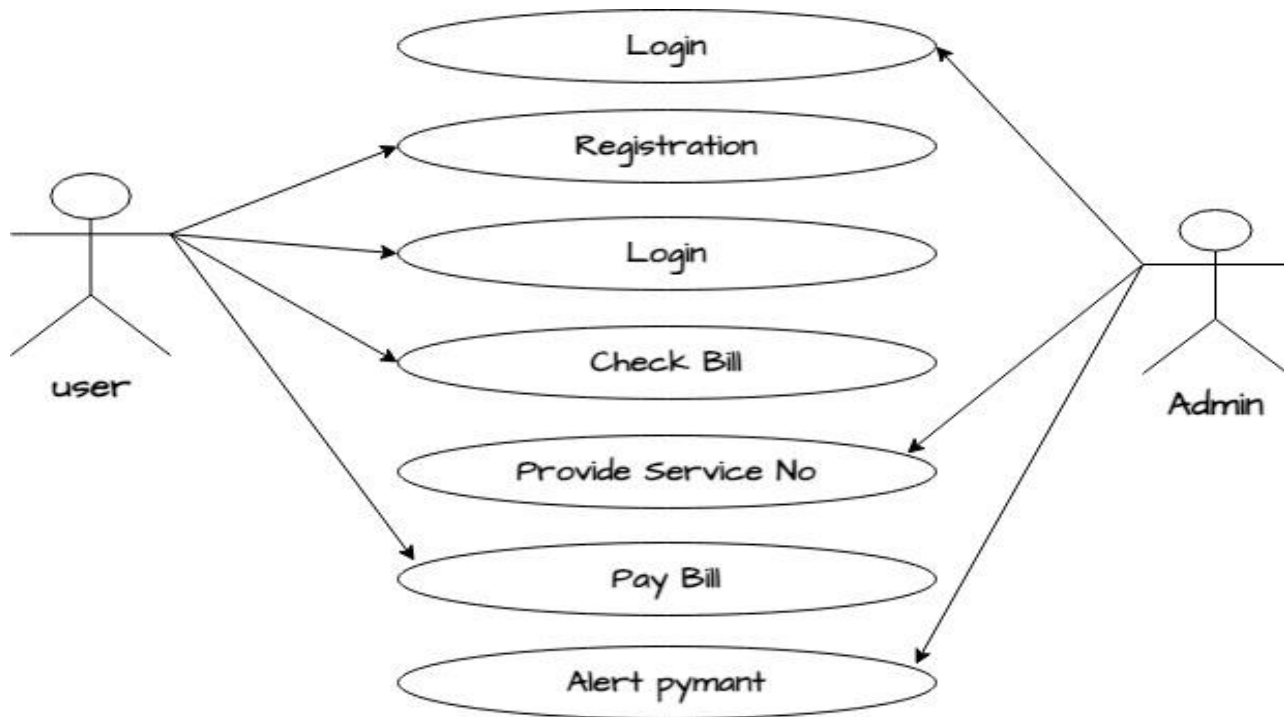
There are four manifesto principles in agile model

- (A) Individuals and interactions over process and tools
- (B) Working software over comprehensive documentation
- (C) Customer collaboration over contract negotiation
- (D) Responding to change over following a plan

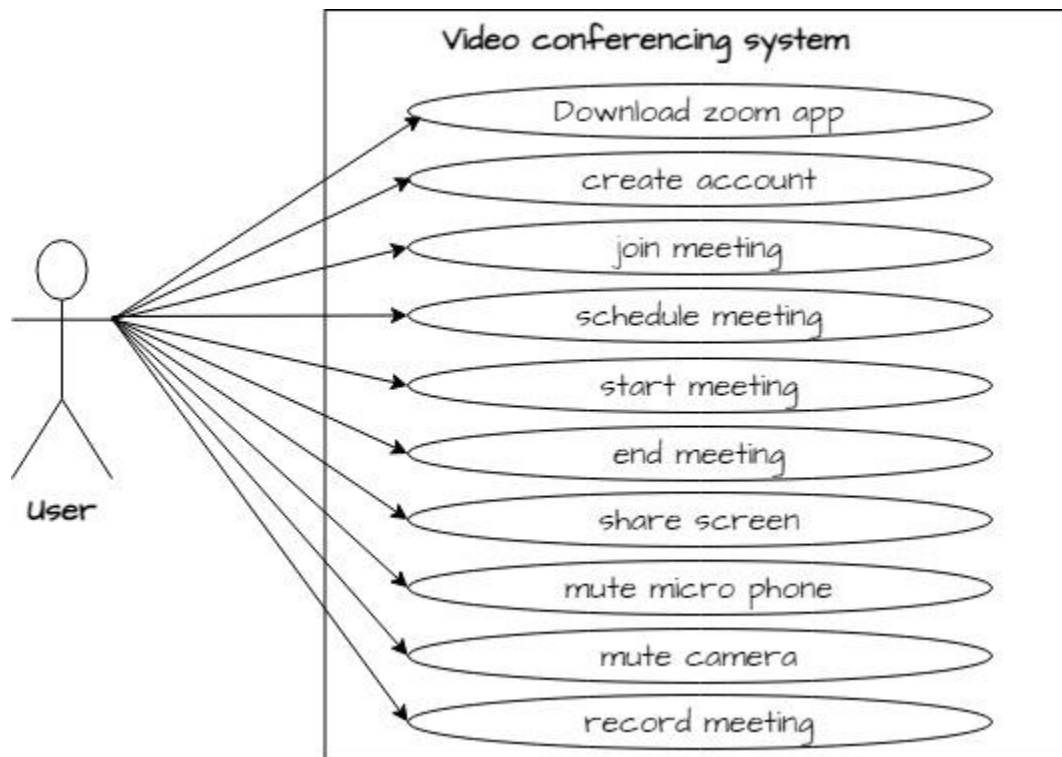
(16) Draw Use case on banking system for customers.



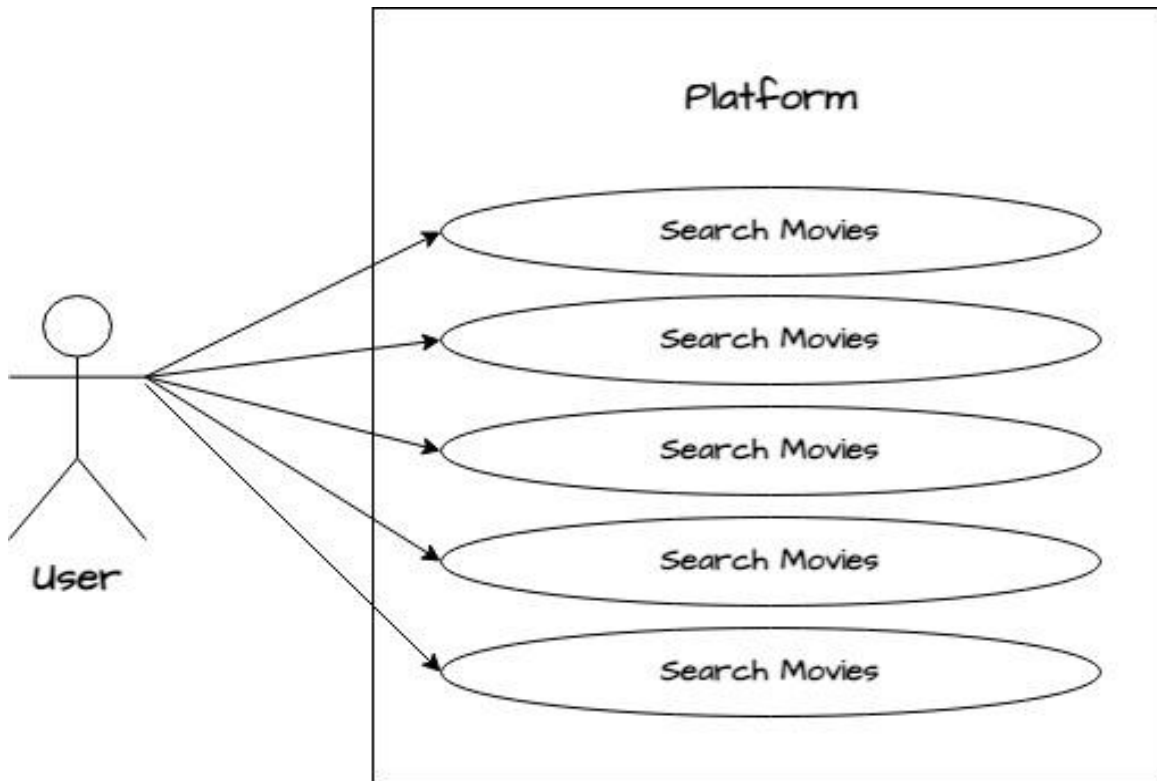
(17) Draw Use case on online bill payment system (paytm)



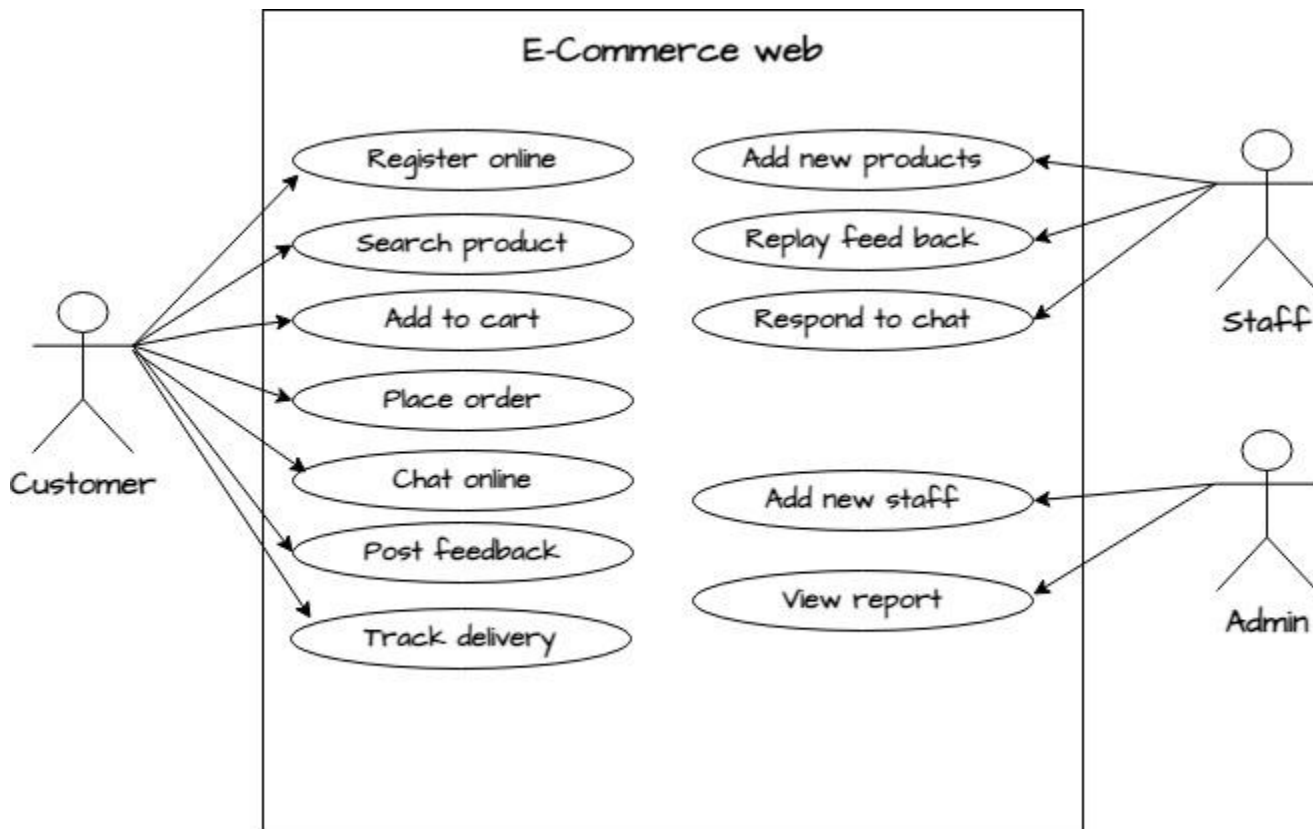
(18) Draw Use case on Broadcasting System



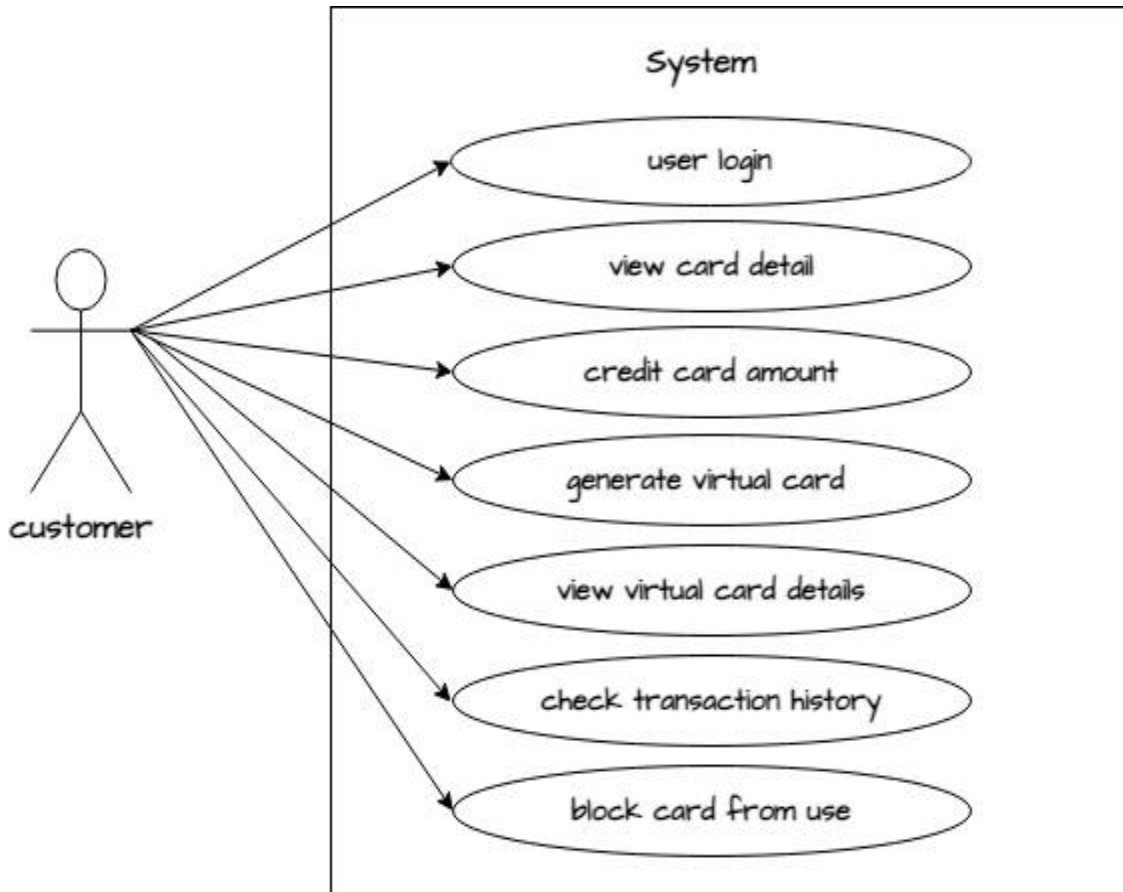
(19) Draw use case on OTT Platform.



(20) Draw use case on E-commerce application



(21) Draw use case on online shopping product using payment gateway.



(22) Explain working methodology of agile model and also write pros and cons.

The Agile model works on the four manifesto principles.

- (1) Individuals and interactions over process and tools.
- (2) Working software over comprehensive document.
- (3) Customer collaboration over contract negotiation.
- (4) Responding to change over following a plan.

▪ Pros:

- (1) Very realistic approach
- (2) Rapid delivery
- (3) Functionality can be developed rapidly
- (4) Resource requirements are minimum
- (5) Little or no planning required
- (6) Promotes teamwork and cross training
- (7) Suitable for fixed or changing requirements
- (8) Gives flexibility to developers

▪ Cons:

- (1) More risk of sustainability, maintainability and Extensibility
- (2) Depends heavily on customer interactions
- (3) Very high individual dependency
- (4) Minimum documentation generated
- (5) Not useful for small projects
- (6) Not suitable for handling complex dependencies