

Module – 1(Fundamental)

1. What is SDLC?

- **SDLC** stands for **Software Development Life Cycle.**
- Software development life cycle (SDLC) is a structured process that is used to design, develop, and test good-quality software.
- SDLC consists of a precise plan that describes how to develop, maintain, replace, and enhance specific software.

2. What is Software Testing?

- Software testing is the process of assessing the functionality of a software program. The process checks for errors and gaps and whether the outcome of the application matches desired expectations before the software is installed and goes live.

3. What is the agile methodology?

- The Agile methodology is a project management and software development approach that emphasizes flexibility, collaboration, and customer-centricity.
- It is the latest model used by major companies today like Facebook, Google, Amazon, etc.

4. What is SRS?

- **SRS** stands for **Software Requirement Specification.**
- As the name suggests, is a complete specification and description of requirements of the software that need to be fulfilled for the successful development of the software system.
- These requirements can be functional as well as non-functional depending upon the type of requirement.
- Depending upon information gathered after interaction, SRS is developed which describes requirements of software that may include changes and modifications that are needed to be done to increase quality of product and to satisfy customer's demand.

5. What is OOPS?

- **OOPS** stands for **Object Oriented Programming**.
- As the name suggests, Object-Oriented Programming or OOPs refers to languages that use objects in programming.
- Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc in programming.

6. Write basic concept of OOPS?

- There are four fundamental concepts of Object-oriented programming Class, Object, Inheritance, Encapsulation, Polymorphism, and Data Abstraction.

7. What is Object?

- In object-oriented programming (OOP), objects are the basic entities that actually exist in the memory.
- Each object is based on a blueprint of attributes and behaviours (variables and functions) defined as Class.
- **For example**, Iphone is an object with brand name Iphone 32GB RAM as properties and calling and texting as behaviors.

8. What is class?

- In object-oriented programming, a class is a template definition of the methods and variables in a particular kind of object.
- Thus, an object is a specific instance of a class; it contains real values instead of variables.
- **For example**, the animal type Dog is a class while a particular dog named Tommy is an object of the Dog class.

9. What is encapsulation?

- Encapsulation is one of the fundamental concepts in object-oriented programming (OOP).
- It describes the idea of wrapping data and the methods that work on data within one unit.
- **For example,** a car is class

Engine, Tires components

Moving, steering function

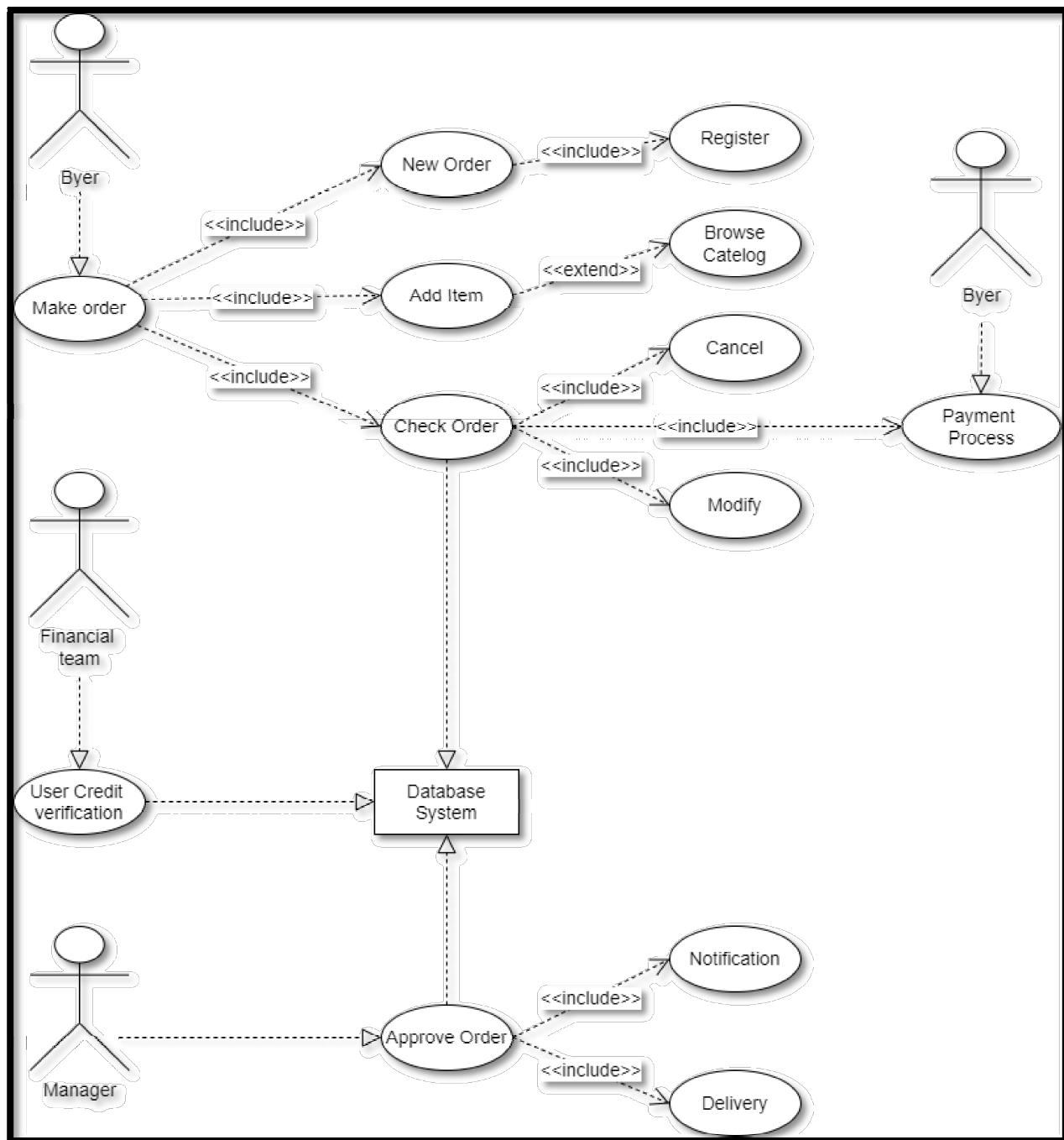
10.What is Inheritance?

- Inheritance is defined as the mechanism of inheriting the properties of the base class to the child class.
- Inheritance is a feature or a process in which, new classes are created from the existing classes.
- **For example,** The new class created is called “derived class” or “child class” and the existing class is known as the “base class” or “parent class”.

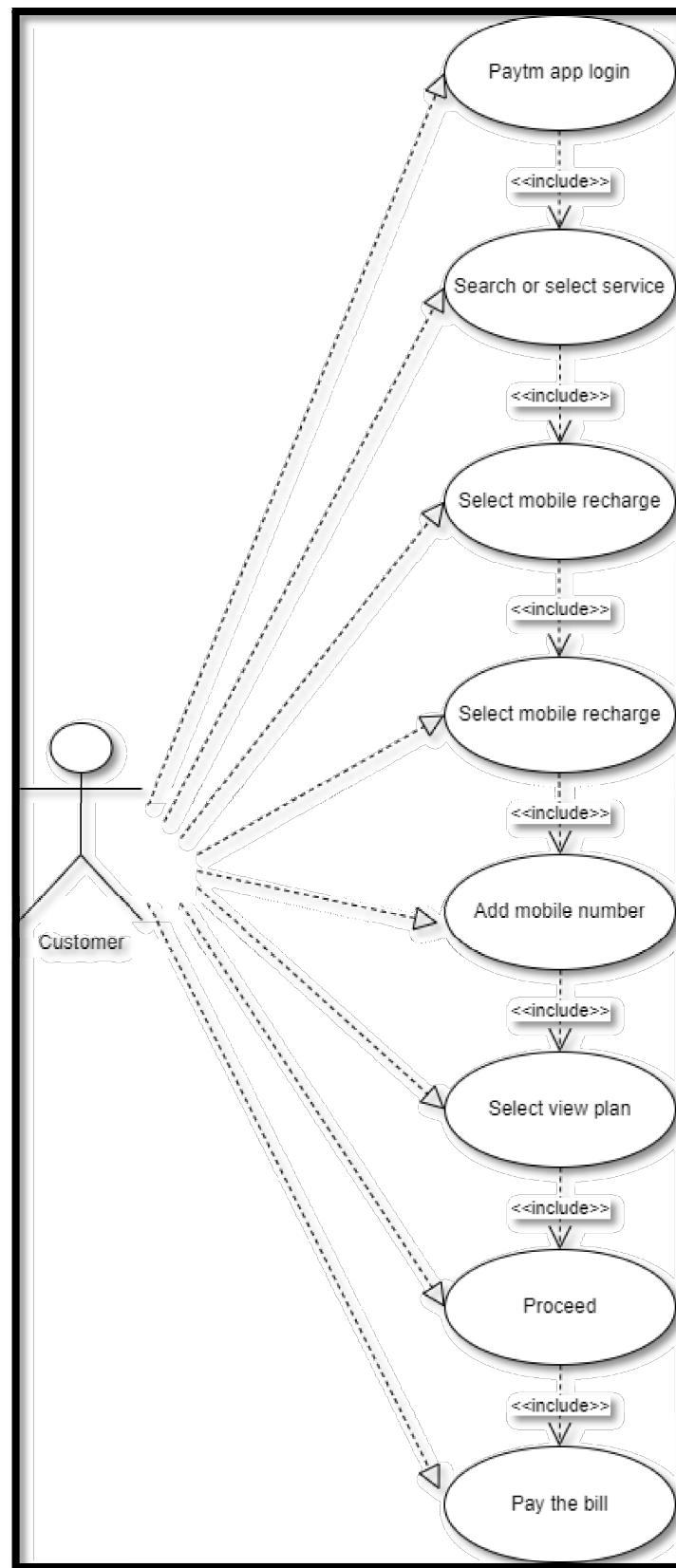
11.What is polymorphism?

- Polymorphism is the ability of objects to take on different forms or behave in different ways depending on the context in which they are used.
- In object-oriented programming (OOP), polymorphism is achieved through the use of inheritance, interfaces, and method overriding.
- **For example,** Consider a person, who can have multiple characteristics at a time, the person can be a friend, a colleague, and an employee at the same time.

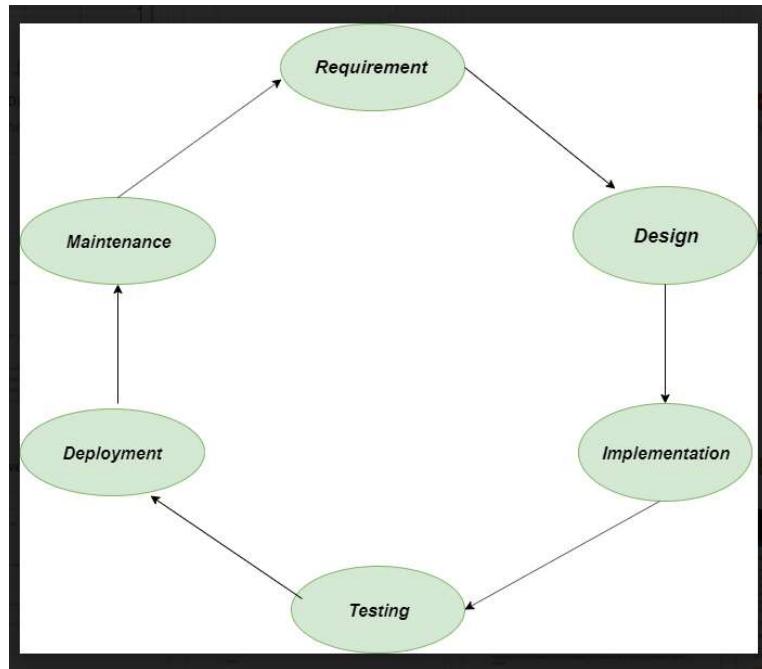
12. Draw the Usecase on online book shopping.



13. Draw the Usecase on online bill payment system(Paytm).



14. Write SDLC phases with the introduction.



- **Requirement Gathering and Analysis**
- **Design**
- **Implementation or coding**
- **Testing**
- **Deployment**
- **Maintenance**

➤ **Requirement Gathering and Analysis :-**

- During this phase, all the relevant information is collected from the customer to develop a product as per their expectation. Any ambiguities must be resolved in this phase only.
- For Example: A customer wants to have an application which involves money transactions. In this case, the requirement has to be clear like what kind of transactions will be done, how it will be done, in which currency it will be done, etc.

- Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

➤ **Desing :-**

- In this phase, the requirement gathered in the SRS document is used as an input and software architecture that is used for implementing systemdevelopment is derived.
- The Design team can now expand upon the information established in the requirement document. The requirement document must guide this decision process.

➤ **Implementation or coding :-**

- Implementation/Coding starts once the developer gets the Design document. The Software design is translated into source code. All the components of the software are implemented in this phase.

➤ **Testing :-**

- Testing starts once the coding is complete and the modules are releasedfor testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.
- There are many types of testing such as Regression testing, Internal testing, Unit testing, Application testing, Stress testing.

➤ **Deployment :-**

- Once the product is tested, it is deployed in the production environment or first **UAT (User Acceptance testing)** is done dependingon the customer expectation.
- In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided bythe customer to go live.

➤ **Maintenance :-**

- Software maintenance is one of the activities in software engineering and the process of enhancing and optimizing deployed software fixing defects.
- There are three types of maintenance....
 - Corrective maintenance
 - Adaptive maintenance
 - Perfective maintenance.

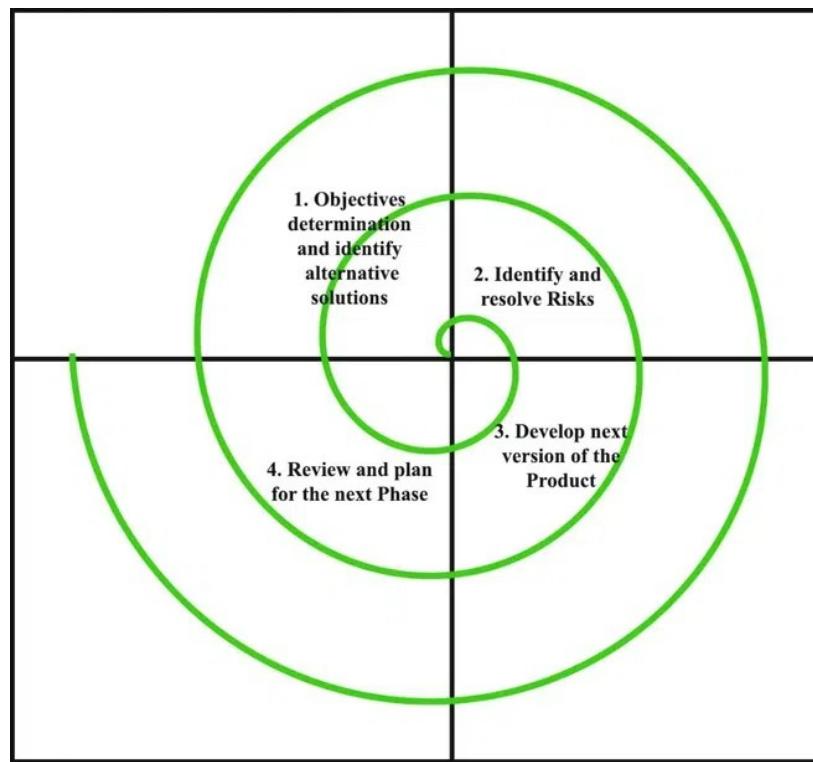
15.Explain phases of the waterfall model.



- **Waterfall Model** is the very first model that is used in SDLC. It is also known as the linear sequential model.
- First, Requirement gathering and analysis is done. Once the requirement is freeze then only the System Design can start. Herein, the SRS document created is the output for the Requirement phase and it acts as an input for the System Design.

- In System Design Software architecture and Design, documents which act as an input for the next phase are created i.e. Implementation and coding.
- In the Implementation phase, coding is done and the software developed is the input for the next phase i.e. testing.
- In the testing phase, the developed code is tested thoroughly to detect the defects in the software. Defects are logged into the defect tracking tool and are retested once fixed. Bug logging, Retest, Regression testing goes on until the time the software is in go-live state.
- In the Deployment phase, the developed code is moved into production after the signoff is given by the customer.
- Any issues in the production environment are resolved by the developers which come under maintenance.

16. Write phases of spiral model.



- The Spiral Model is a risk-driven model, meaning that the focus is on managing risk through multiple iterations of the software development process.

- It consists of the following phases:
 - **Planning :-**
 - The first phase of the Spiral Model is the planning phase, where the scope of the project is determined and a plan is created for the next iteration of the spiral.
 - **Risk analysis :-**
 - In the risk analysis phase, the risks associated with the project are identified and evaluated.
 - **Engineering :-**
 - In the engineering phase, the software is developed based on the requirements gathered in the previous iteration.
 - **Customer evaluation :-**
 - In the evaluation phase, the software is evaluated to determine if it meets the customer's requirements and if it is of high quality.
- The Spiral Model is often used for complex and large software development projects, as it allows for a more flexible and adaptable approach to software development. It is also well-suited to projects with significant uncertainty or high levels of risk.
- The Radius of the spiral at any point represents the expenses(cost) of the project so far, and the angular dimension represents the progress made so far in the current phase.

17. Write agile manifesto principles.

- **Individuals and interactions**
- **Working software**
- **Customer collaboration**
- **Responding to change**
- It consists of the following phases:

➤ **Individuals and interactions :-**

- In agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

➤ **Working software :-**

- Demo working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.

➤ **Customer collaboration :-**

- As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.

➤ **Responding to change :-**

- Agile development is focused on quick responses to change and continuous development.

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

- 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.
- Agile Methods break the product into small incremental builds. These builds are provided in iterations.
- 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.

➤ **Pros :-**

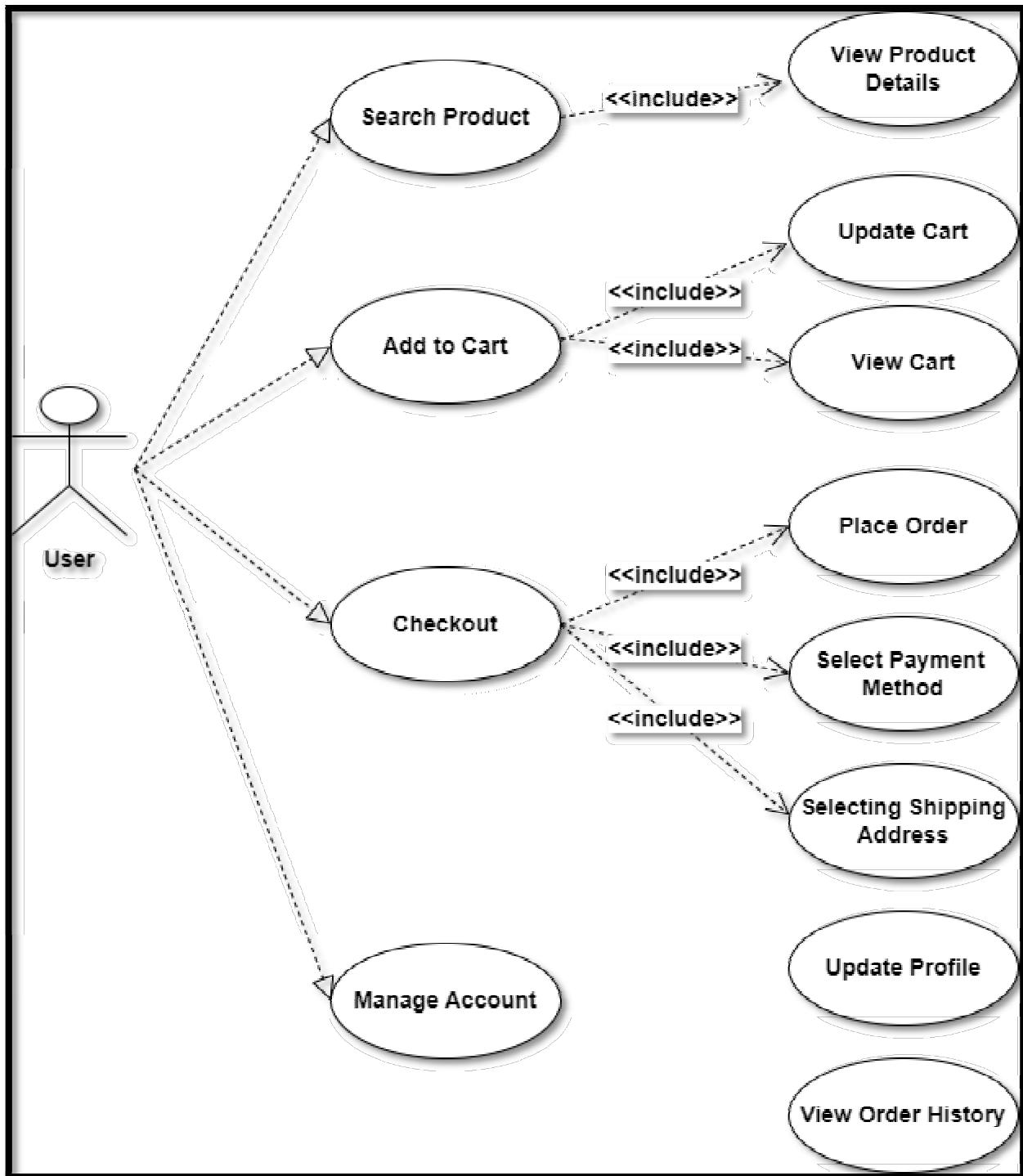
- Is a very realistic approach to software development.
- Promotes teamwork and cross training.
- Functionality can be developed rapidly and demonstrated.
- Resource requirements are minimum.
- Suitable for fixed or changing requirements.

- Minimal rules, documentation easily employed.
- Little or no planning required.
- Easy to manage.

➤ **Cons :-**

- There is very high individual dependency, since there is minimum documentation generated.
- Transfer of technology to new team members may be quite challenging due to lack of documentation.

19. Draw the Usecase on online shopping product using COD.



20. Draw the Usecase on online shopping product using payment gateway.

