Software Testing Assignment Module–1(Fundamental)

1. What is SDLC

SDLC (Software Development Life Cycle) is a structured process for developing software, consisting of stages like Planning, Analysis, Design, Implementation, Testing, Deployment, and Maintenance. It ensures quality, efficiency, and alignment with user requirements.

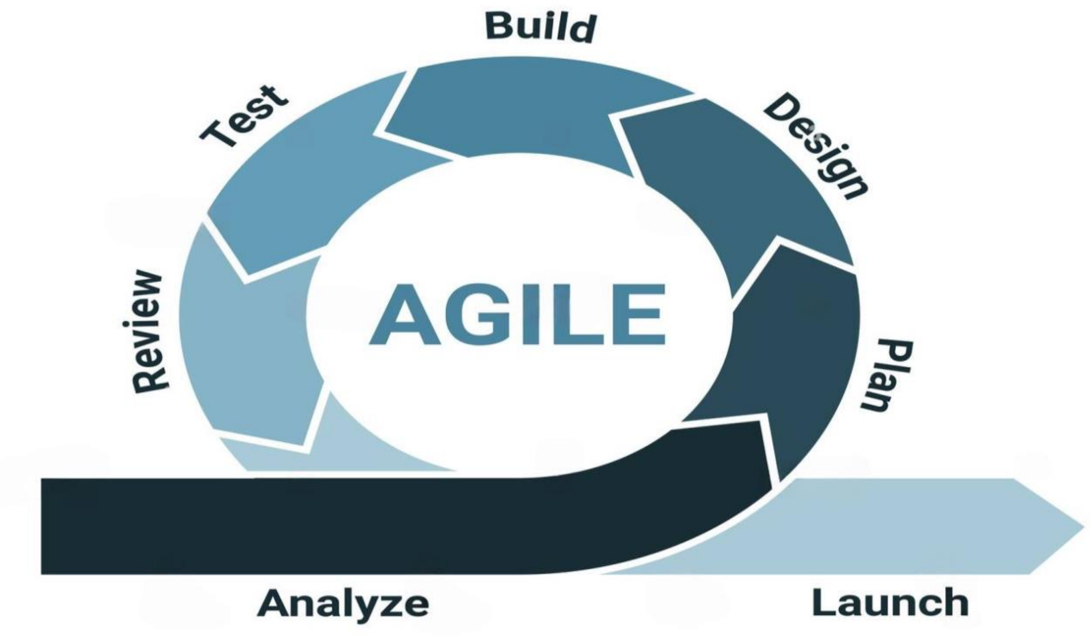
2. What is software testing?

Software testing is a process which is used to identify the correctness, completeness and quality of the developed software.

Software testing is the process of verifying that the actual software product matches the expectations requirements and ensure that the software product is error-free.

3. 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.



Prepared By: Nishi Patel

4. What is SRS

The Software Requirements Specification (SRS) document is a framework for software development. It describes functional and non-functional requirements, limitations and how the software is intended to be used

5. What is oops

Object Oriented Programming is considered as a collection of objects. They are used software program in simple reusable code. Here it is called Functional testing or Black Testing the box.

6. Write Basic Concepts of oops

1. Class

2. Object

3. Encapsulation

4. Abstraction

5. Polymorphism

6. Inheritance

7. What is object

The object is an individual unit of OOP which can be assessed with the properties called data member and member function. It creates the memory of the class.

8. What is class

A class is a group of variables and data members that work together to determine behavior. A class is a blueprint or template that outlines an object's characteristics and actions.

9. What is Encapsulation

A wrapping up of data and functions into a single unit is called Encapsulation. IT hide/include private access of data member & member function.

10. What is Inheritance

In object-oriented programming, inheritance allows a class (child class) to inherit characteristics and behaviors (attributes and methods) from a parent class. In addition to introducing its own special features or changing the inherited ones, it enables the child class to reuse code from the parent class. It's similar to how a youngster develops their own uniqueness while simultaneously acquiring characteristics from their parents.

11. What is Polymorphism

In OOP, polymorphism refers to the ability of a same function or method to operate differently based on the object it is operating upon. It is similar to how a square or a circle can do the same operation, such as "drawing," in different ways. Simply put, polymorphism enables the use of the same name or method for many object types, each of which carries out the task in a unique manner.

12. Write SDLC phases with basic introduction

1. Requirements Collection/Gathering
- Establish Customer Needs

2. Analysis
- Model And Specify the requirements-"What"

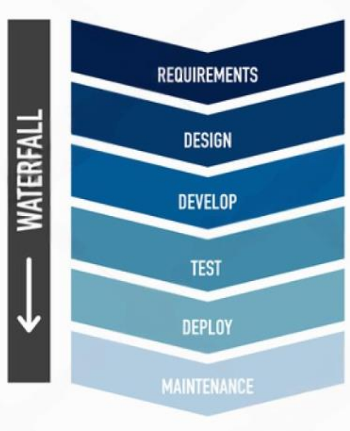
3. Design (Low Level Design & High Level Design)
- Model And Specify a Solution – "Why"

4. Implementation / Coding
- 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



1. Requirement Gathering and Analysis:
-In this phase, all the requirements for the software are collected from the stakeholders. The focus is on understanding what the system should do and documenting all functional and non-functional requirements
-This phase ends with a detailed requirement specification document.
2. System Design:
- Based on the gathered requirements, the system architecture and design are created. This includes both high-level design (overall structure) and detailed design (specific components and modules).
- The design phase helps in creating a blueprint for the system.

3. Implementation (Coding):
- During this phase, the actual code for the software is written according to the design specifications. Developers build the system component by component.
- After coding, the individual units are tested for functionality.
4. Integration and Testing (Verification):
- Once the system components are implemented, they are integrated to form a complete system. Comprehensive testing is performed to find and fix any defects, ensuring that the software meets the specified requirements.
- Testing includes unit testing, integration testing, and system testing.

5. Deployment (Installation):
- After successful testing, the software is deployed to the production environment where it becomes available for users. This is the phase where the system is installed and made operational.

6. Maintenance:
- After deployment, the software enters the maintenance phase. This involves providing ongoing support, fixing any bugs that arise, and making updates or improvements as needed based on user feedback.
- Maintenance ensures the software continues to operate smoothly and adapt to changing needs.

Since each step in the Waterfall model is unique and the procedure is inflexible, it is challenging to return to a prior stage (such as switching from coding to requirements). Projects with clear, consistent criteria are the greatest candidates for this strategy.

14. Write phases of spiral mode

1. Planning Phase (Objective Setting)
2. Risk Analysis and Assessment
3. Engineering Phase (Development and Testing)
4. Evaluation Phase (Customer Feedback)
5. Next Iteration (Planning for the Next Spiral)

15. Write agile manifesto principles

- Individuals and interactions
- Working software
- Customer collaboration
- Responding to change

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

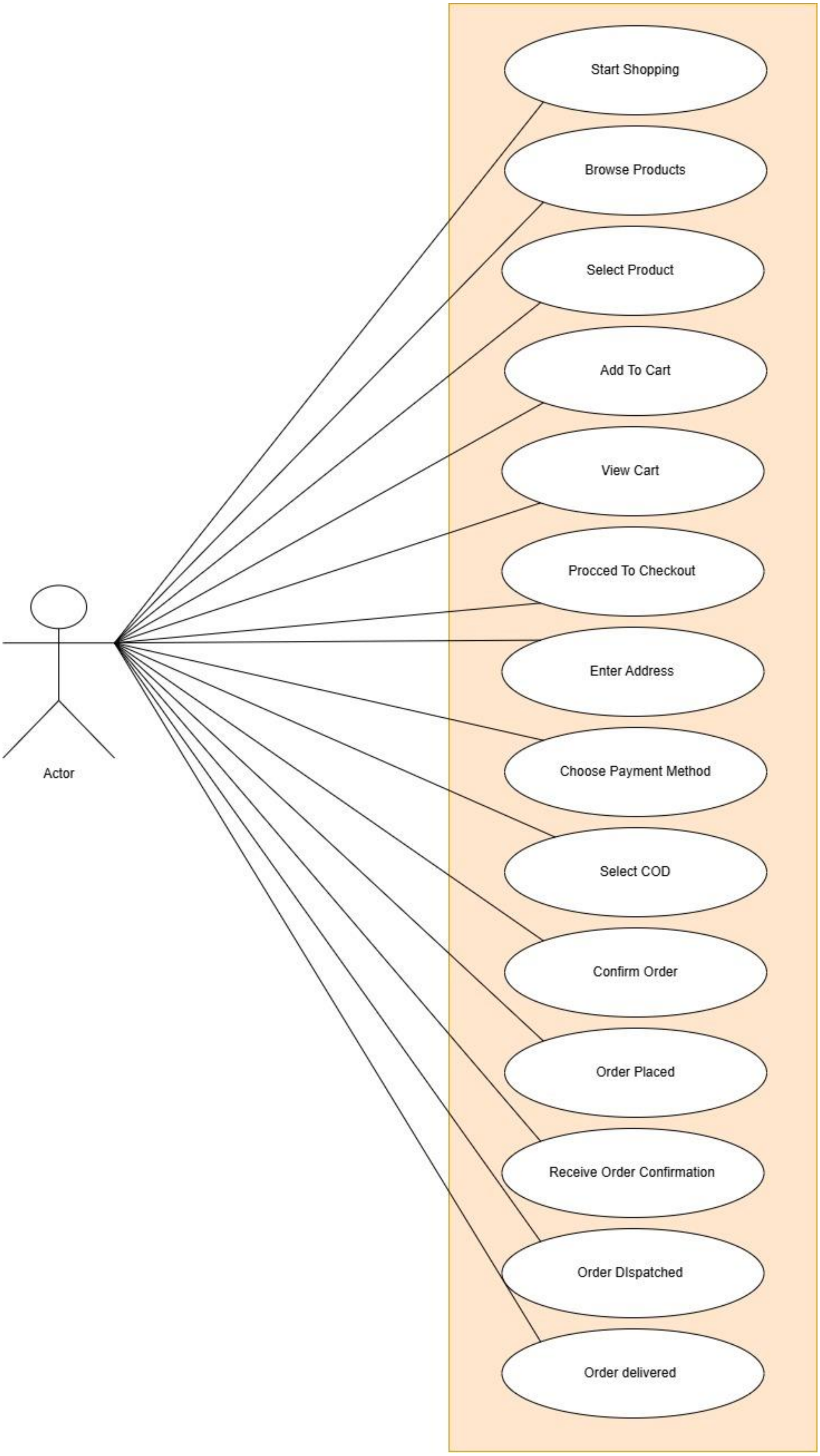
Pros:

1. Very realistic approach
2. Fast delivery
3. Functionality can be built quickly
4. Resource requirements are minimal
5. Minimal planning required.
6. Encourages teamwork and cross-training
7. Suitable for both fixed and variable requirements
8. Provides flexibility to developers

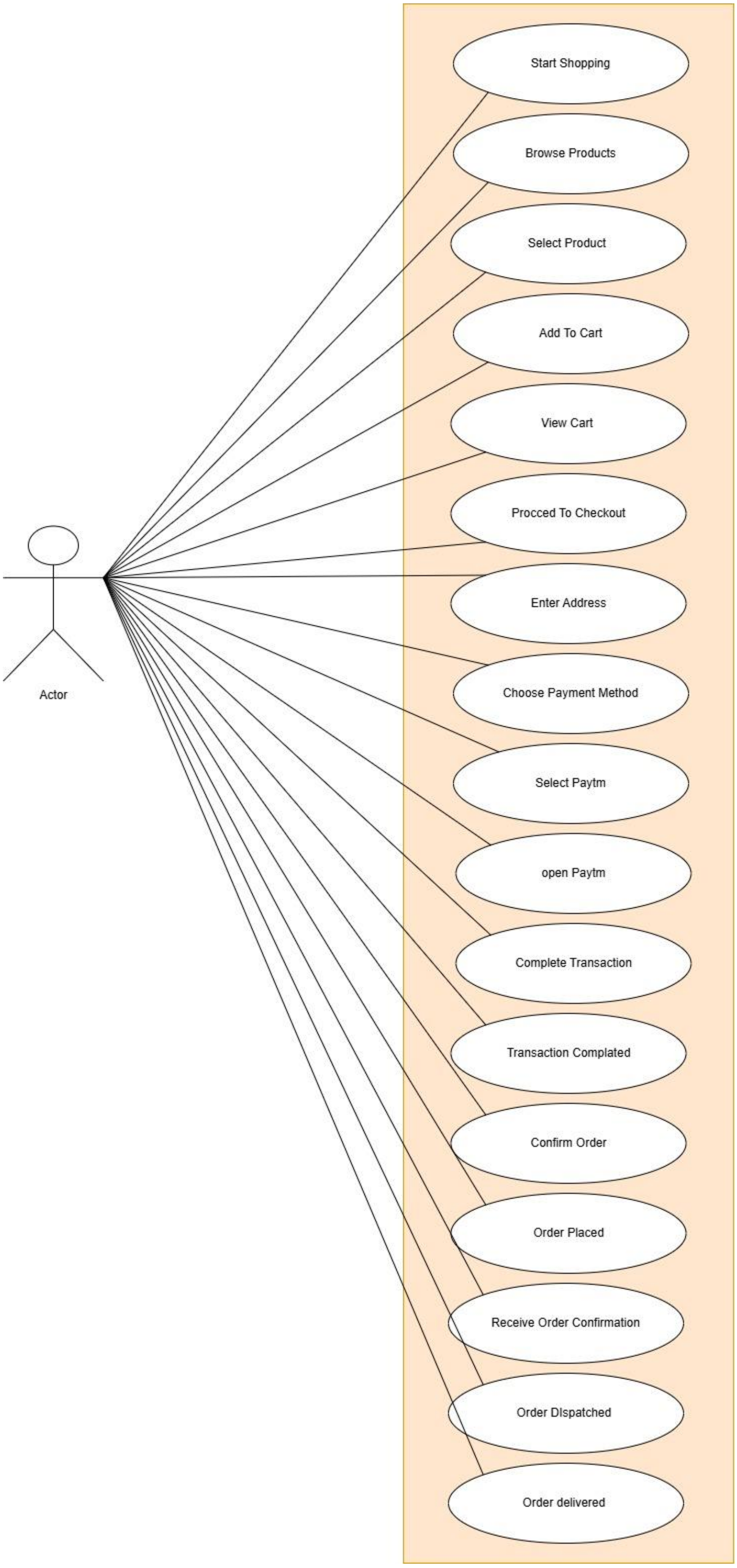
Cons:

1. Increased risk of sustainability, maintainability, and extensibility
2. Rely primarily on customer interactions
3. Individual reliance is really high
4. The minimum amount of documentation is generated
5. Not suitable for minor projects
6. Unsuitable for managing complicated dependencies

17. Draw usecase on Online shopping product using COD

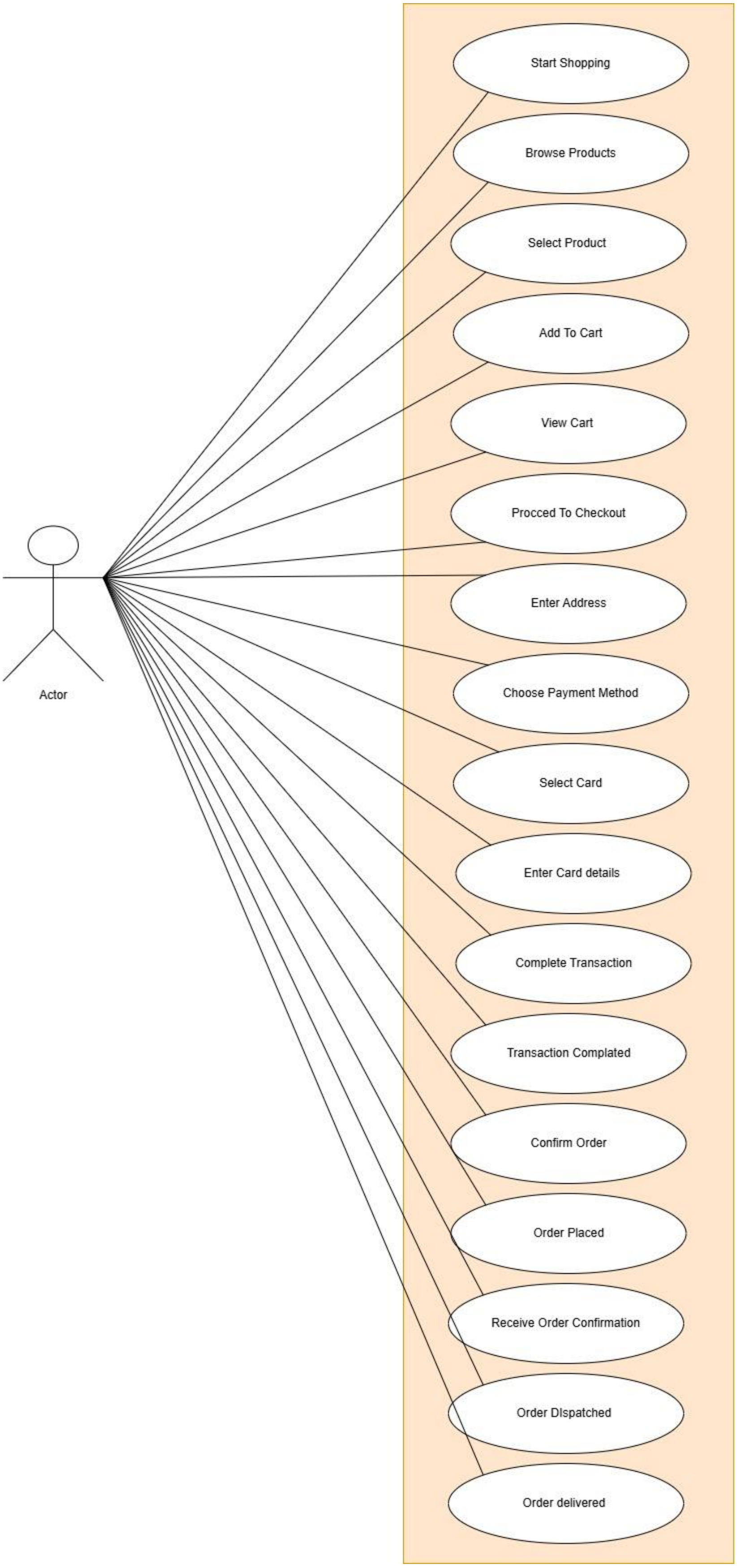


18. Draw Usecase on online bill payment system (paytm)

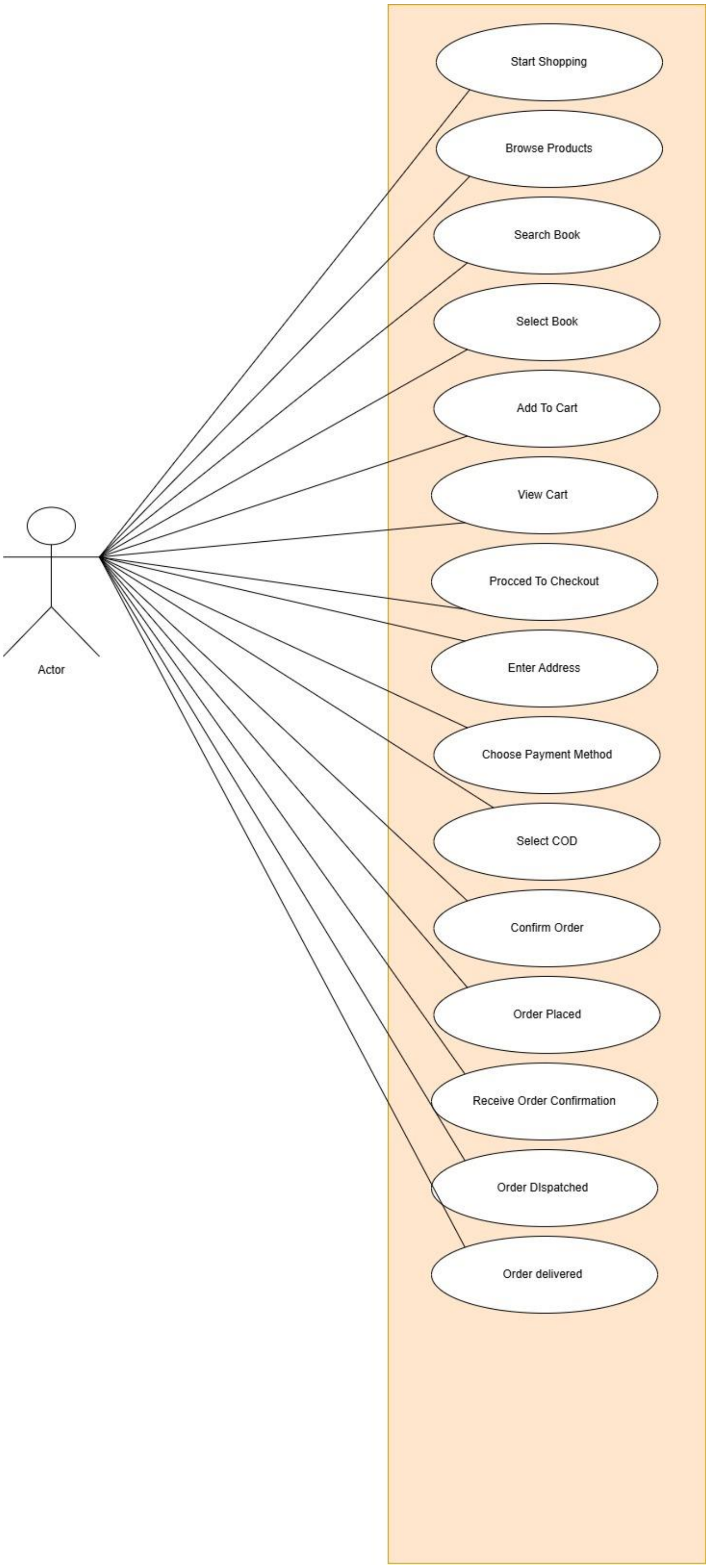


Prepared By: Nishi Patel

19. Draw usecase on Online shopping product using payment gateway



20. Draw Usecase on Online book shopping



Prepared By: Nishi Patel

Prepared By: Nishi Patel