# SOFTWARE TESTING ASSIGNMENT

## MODULE-1 (FUNDAMENTAL)

### Q.1) What is SDLC?

**Ans:** SDLC stands for Software Development Life Cycle. SDLC is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production and beyond.

SDLC has a method that outlines the various stages or phases involved in the development of software applications that are as below,

- **Requirements Gathering**: In this stage, the project requirements are collected to define the software's scope, goals, and features as per documentation process.

- **Analysis or Planning**: In this stage, the planning or analysis is takes place as per clients requirements and documentation.

- **Design:** In this phase, the designers creates designs based on the gathered requirements. It involves defining the overall structure, database design, user interface design, and other technical specifications.

- **Implementation:** Also known as the coding phase, this stage involves writing the actual code based on the design specifications. It includes programming, unit testing, and integration of different software modules.

- **Testing:** The software undergoes various levels of testing to identify and fix defects or bugs. This includes unit testing, integration testing, system testing, and user acceptance testing.

- **Deployment**: Once the software passes all the testing phases, it is deployed to the production environment. This involves installing the software on the target systems and making it available for end-users.

- **Maintenance:** After deployment, the software requires ongoing maintenance and support, This includes bug fixes, updates, etc.

**Q.2) What is software testing?**

**Ans**: Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

The primary goal of software testing is to find bugs, defects, and issues in the software and ensure that it performs as expected. By detecting and fixing these problems early in the development lifecycle, testing helps enhance the overall quality and reliability of the software.

Testing is a process rather than a single activity.

It has two types,

- Static testing: It can test and find defects without executing the code. Static testing is done during the verification process. This testing includes reviewing of the documents and static analysis.
- Dynamic testing: Dynamic testing is a type of software testing that involves executing the software and evaluating its behavior during runtime. It is also known as functional testing, as it focuses on testing the software’s functionality.

There are 7 key principles,

1. Testing shows presence of Defects

2. Exhaustive Testing is Impossible

3. Early Testing

4. Defect Clustering

5. The Pesticide Paradox

6. Testing is Context Dependent

7. Absence of Errors Fallacy.

**Q.3) What is SRS?**

**Ans:** SRS stands for software requirements specification.

software requirements specification is a complete description of the behavior of the system to be developed.

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

It is a document that describes what the software will do and how it will be expected to perform.

Types of requirements,

- Customer requirement
- Functional requirement
- Non Functional requirement

**Q.4 What is OOPS?**

**Ans:** OOPS stands for Object Oriented Programming System. 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.

OOPS is identifying objects and assigning responsibilities to these objects.

Everything in this world is an object. Eg. A flower, A tree, An animal, A student, A professor, etc.

**Q.5 Write basic concepts of OOPS.**

**Ans.** There are several concepts of OOP which are as below,

- **Object:** It is a part or example or representative of a class.
- **Class:** Class is a blueprint for an object. Class can contain many objects but objects can only relate to single class.
- **Encapsulation:** It is like wrapping up of data into single unit.
- **Inheritance**: Ability to adopt the property and behaviour of parent class.
- **Polymorphism**: It means having many forms or the ability to react the same thing in different way.

  **Types:** Overloading and Overriding
- **Abstraction:** It means to hide irrelevant data from user or hide the implementation from the user but shows only essential information to the user.

**Q.6 What is object?**

**Ans. :** It is a part or example or representative of a class. An object represents an individual, identifiable item, unit or entity, either real or abstract, with a well defined role in the problem domain.

objects are the things you think about first in designing a program and they are also the units of code that are eventually derived from the process.

This is the basic unit of Object oriented programming.

**Q.7 What is class?**

**Ans.** Class is a blueprint for an object. Class can contain many objects but objects can only relate to single class.

a template definition of the methods and variables in a particular kind of object. That’s why, an object is a specific instance of a class.

A class is a blueprint of an object. It is a user-defined data type. Inside a class, we define variables, constants, member functions, and other functionality.

**Q.8 What is encapsulation?**

**Ans.** An encapsulation means binding variables and methods together into a single unit and preventing them from being accessed by other classes.

Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.

Encapsulation in java is the process of wrapping up of data and behavior of an object into single unit and unit here is class.

**Q.9 What is inheritance?**

**Ans.** Inheritance is a mechanism in which one class acquires the property of another class. For example, a child inherits the traits of his/her parents. With inheritance, we can reuse the fields and methods of the existing class.

It is an Ability to adopt the property and behaviour of parent class.

It describes the relationship between two classes. A class can get some of its characteristics from a parent class and then add unique features of its own.

**Q.10 What is polymorphins?**

**Ans.** Polymorphism means having many forms.

Polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. A real-life example of polymorphism is a person who at the same time can have different characteristics.

It allows different objects to respond to the same message in different ways, the response specific to the type of the object.

**Q.11 Write SDLC phases with basic introduction.**

**Ans.** SDLC stands for Software Development Life Cycle, it is a sequence vise approach followed by software development teams to plan, design, develop, test, deploy, and maintain software applications. It consists of several well-defined phases.

**Requirements Gathering**: In this phase, The client gives documents regarding to their requirements and desires. The goal is to understand the needs and expectations of the end users.

**Analysis and Planning:** The gathered requirements are analyzed, and a detailed plan is created, including project scope, timelines, resource allocation, and potential risks. This phase helps define the project's goals and deliverables.

**Design**: The design phase involves creating a blueprint for the software solution based on the requirements. It includes defining the software architecture, database structure, user interface design, and overall system design.

**Development**: In this phase, the actual coding and development of the software take place. Programmers write code according to the design specifications, and the development team follows coding standards and best practices.

**Testing:** Once the development is complete, the software undergoes various testing procedures. This phase ensures that the software functions as intended, is free of bugs and errors, and meets the specified requirements. It includes unit testing, integration testing, system testing, and user acceptance testing.

**Deployment:** After successful testing, the software is deployed to the production environment or made available to end users. This phase involves installing the software, configuring it, and providing necessary documentation and training to users.

**Maintenance:** Once the software is deployed, it requires regular maintenance and support. This phase includes bug fixing, updates, enhancements, and addressing any issues that arise during the software's lifespan. It ensures the software remains reliable, secure, and up-to-date.

**Q.12 Explain phases of waterfall model.**

**Ans.** The waterfall model is a sequential software development process we can say it is similar to SDLC, that is divided into phases. Each phase must be completed before moving on to the next one. Here are the typical phases of the waterfall model:

**Requirements Gathering:** In this phase, the project team collects the documents and all the requirements and expectations of the software to be developed.

**System Design:** Based on the requirements collected, the system design phase starts creating a high-level design of the software. This includes defining the architecture, components, and their interactions

**Implementation:** This phase involves the actual coding and programming of the software based on the design specifications. The development team translates the system design into executable code, creating individual software components and integrating them to build the complete system.

**Testing:** Once the implementation is complete, the testing phase begins. It involves various types of testing to ensure that the software functions as intended and meets the defined requirements. This includes unit testing, integration testing, system testing, and user acceptance testing.

**Deployment:** Once the software has successfully passed all testing stages, it is ready for deployment. The deployment phase involves installing the software in the target environment and making it available for users.

**Maintenance:** After the software is deployed, the maintenance phase begins. This involves providing ongoing support, bug fixes, updates, and enhancements based on user feedback and changing requirements.

**Q.13 What is agile methodology?**

**Ans**. The Agile methodology is an approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement.

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. At the end of the iteration a working product is displayed to the Customer.

Agile methodology is an iterative and incremental approach to project management and software development. It emphasizes flexibility, collaboration, and adaptability throughout the development process. Agile methodologies promote a more dynamic and responsive way of working, enabling teams to deliver high-quality products or services in a faster and more efficient manner.

**Q.14 Explain working methodology of agile model and also writes pros and cons.**

**Ans.** The Agile methodology is an approach that involves breaking the project into phases and emphasizes continuous collaboration and improvement.

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. At the end of the iteration a working product is displayed to the Customer.

**Pros:**

- It is a very realistic approach to software development
- Functionality can be developed rapidly

- Resource requirement are minimum
- Suitable for fixed or changing requirements
- Little planning required
- Easy to manage
- Gives flexibility to developers

**Cons:**

- Not suitable for handling complex requirements
- More risk of sustainability, maintainability
- Depends heavily on customer interaction
- Minimum documentation generated due to high individual dependency
- Transfer of technology to new team members maybe quite challenging due to lack of documentation
- Functionality to be delivered and adjustments to meet the deadlines

**Q.15 Write agile manifesto principles.**

**Ans.** The following are the principles of the Agile Manifesto:

- Meeting end user's needs with early and continuous delivery of work.

- Being open to changes in requirements even late in the project.
- Delivering completed work at regular intervals, preferably short ones.
- Working with the project team and business owners daily.
- Assembling a motivated team, providing them with the right environment and support, and trusting them.
- Communicating face-to-face regularly.
- Using completed work to measure progress.
- Creating processes that promote sustainable efforts and a constant pace of work.
- Requiring continuous attention to excellence through good design.
- Encouraging simplicity.
- Recognizing that the best work emerges from self-organized teams that deliver the best architectures and designs.
- Reflecting regularly on how the team can be more effective and fine-tuning and adjusting the approach.

**Q.16 Write phases of spiral model.**

**Ans.** The Spiral Model is a software development methodology that combines elements of both waterfall and iterative development models. It emphasizes risk management and allows for flexibility in accommodating changes throughout the development process.

The model consists of four main phases:

**Planning:** In this phase, project objectives, requirements, and constraints are identified. The development team evaluates alternative solutions, performs feasibility analysis, and creates a preliminary project plan. Risks are assessed, and strategies are formulated to manage them effectively.

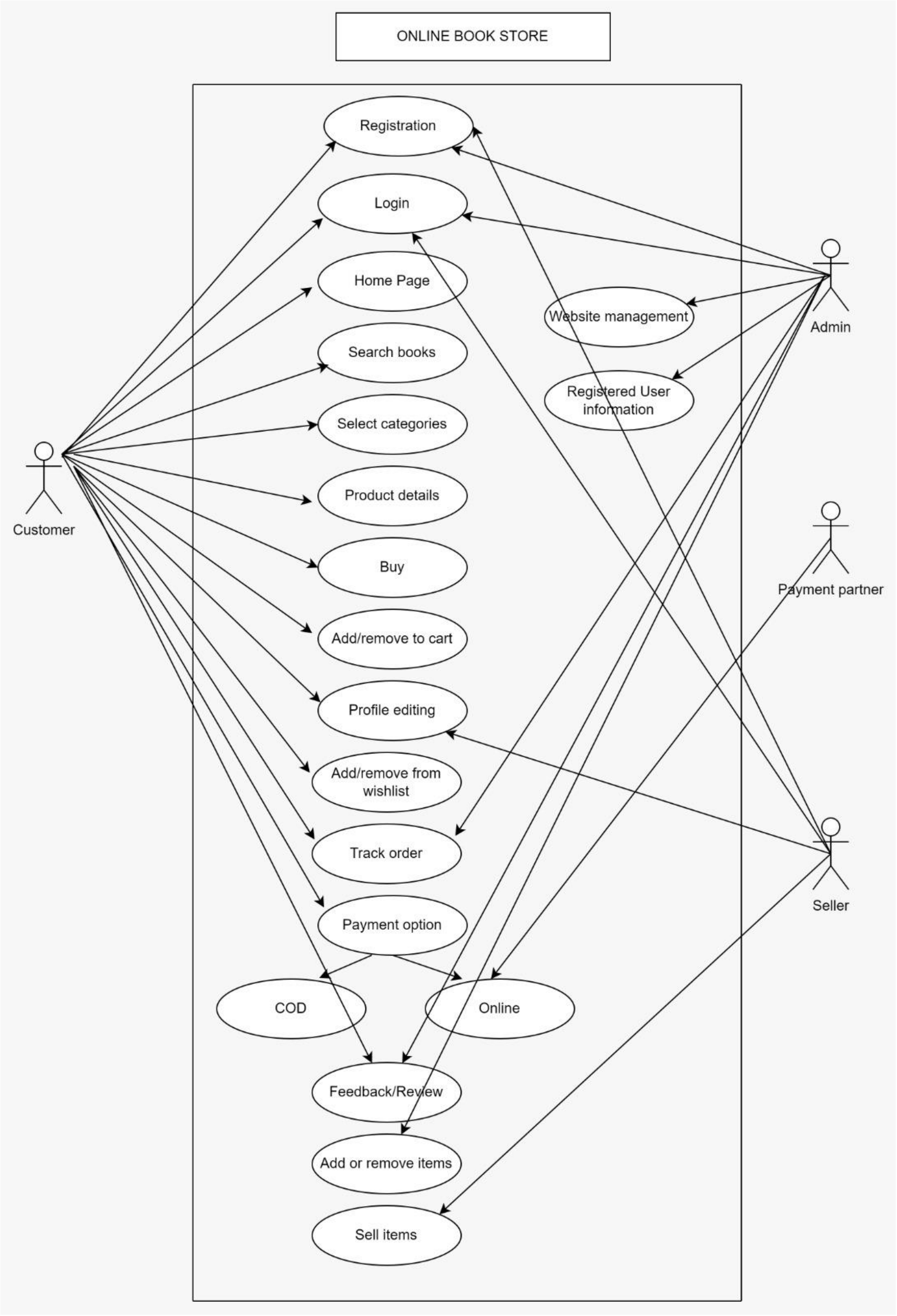
**Risk Analysis:** This phase focuses on identifying, analysing, and mitigating risks associated with the project. The development team assesses potential risks and develops strategies to address them. This involves a thorough evaluation of technical, schedule, and cost risks. Prototypes or proof-of-concept models may be created to explore potential solutions and mitigate risks.

**Engineering:** The engineering phase involves the actual development of the software. It follows an iterative approach where the development team builds and tests the software incrementally. The project requirements are divided into smaller modules or components, and each iteration involves designing, coding, and testing these components. Feedback from users and stakeholders is collected and incorporated into subsequent iterations.

**Evaluation:** This phase involves a comprehensive review of the software and its performance. The software is evaluated against the project objectives and requirements defined in the planning phase. User feedback and testing results are collected and analyzed to identify areas for improvement. Based on the evaluation, the project may proceed to the next iteration or be closed if the objectives have been met.

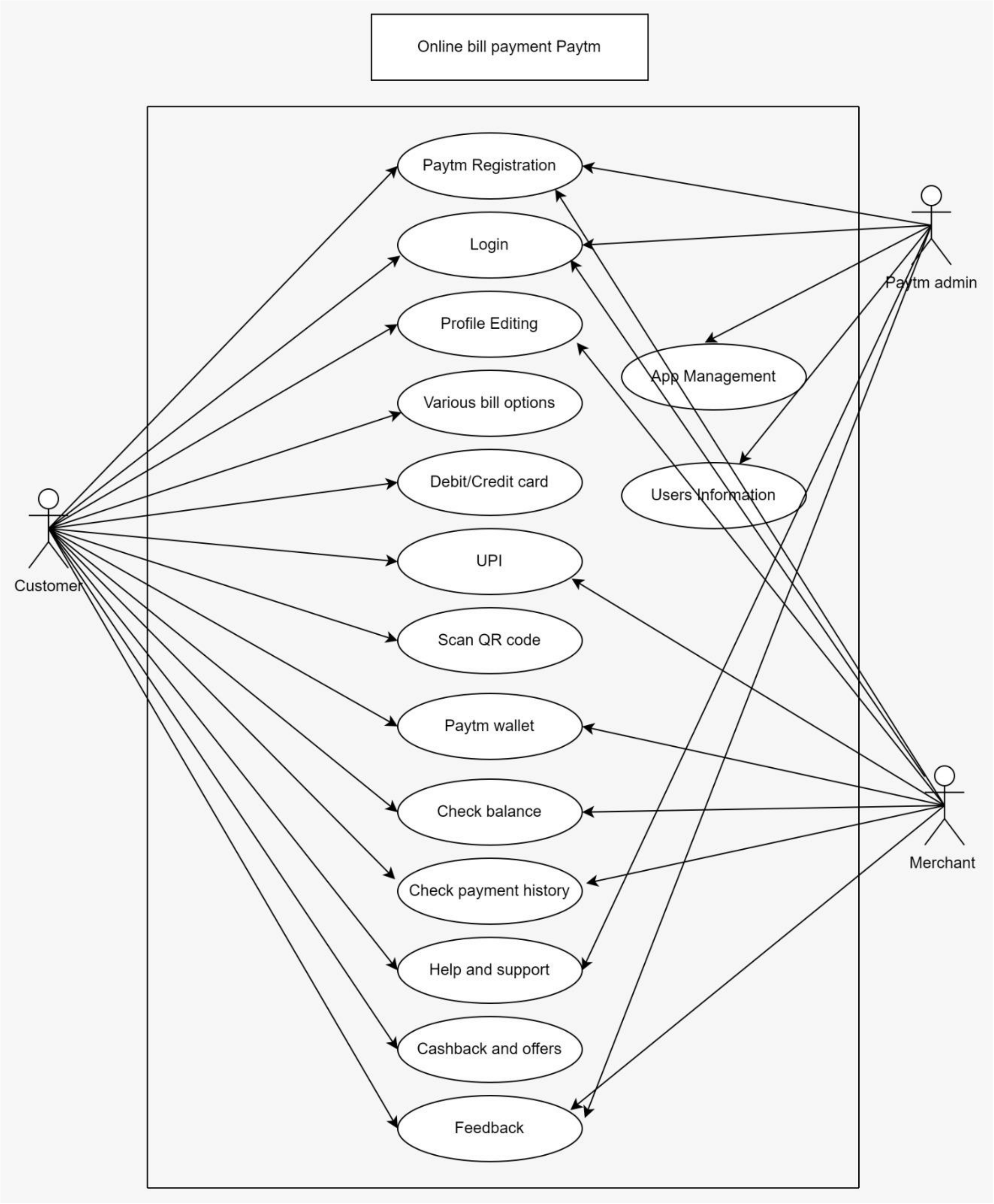
**Q.17 Draw usecase on online book shopping.**

**Ans.**


ONLINE BOOK STORE
Registration
Login
Home Page
Search books
Select categories
Product details
Buy
Add/remove to cart
Profile editing
Add/remove from wishlist
Track order
Payment option
COD
Online
Feedback/Review
Add or remove items
Sell items
Website management
Registered User information
Customer
Admin
Payment partner
Seller

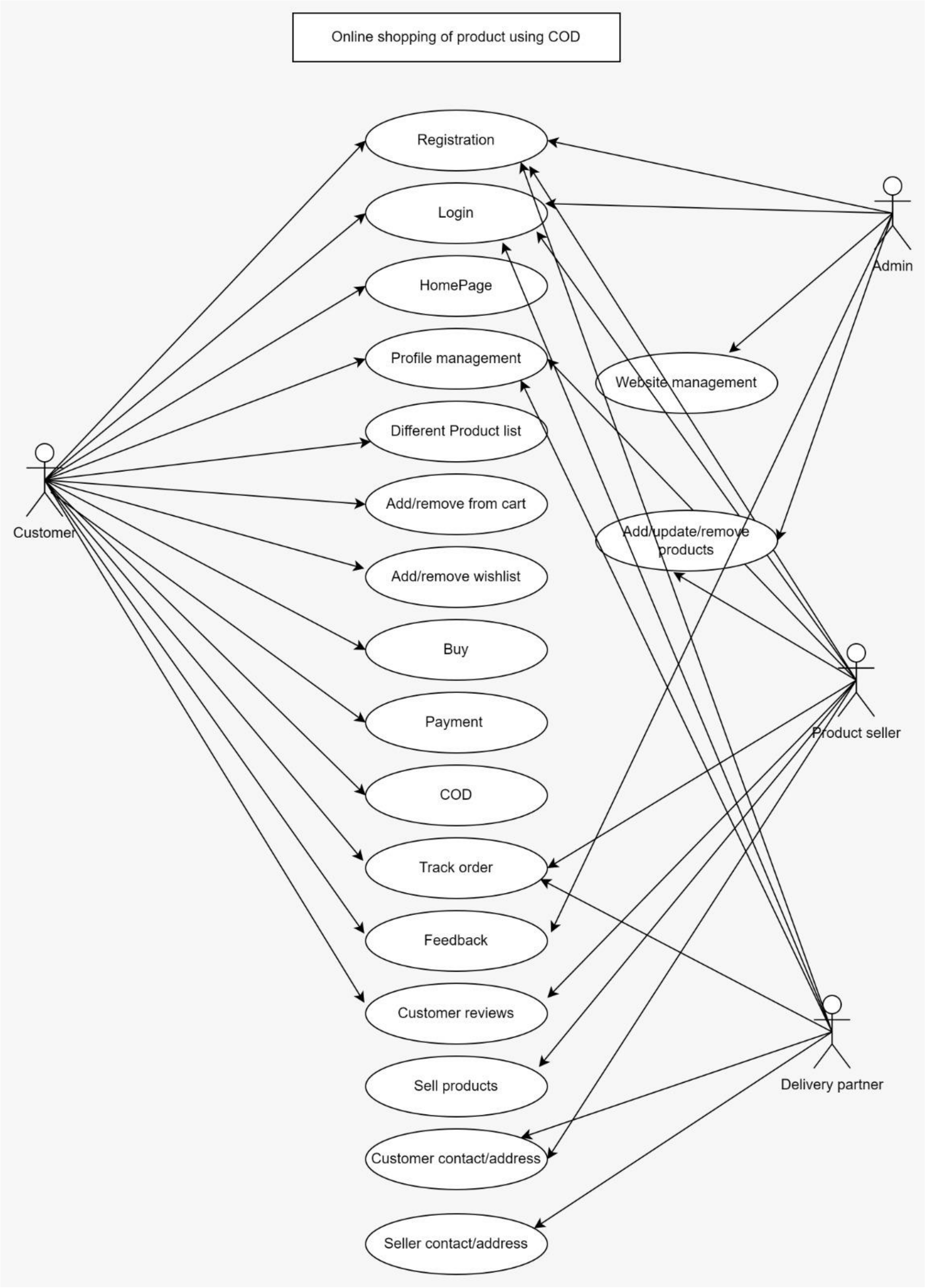
**Q.18 Draw usecase on online bill payment system Paytm.**

**Ans.**



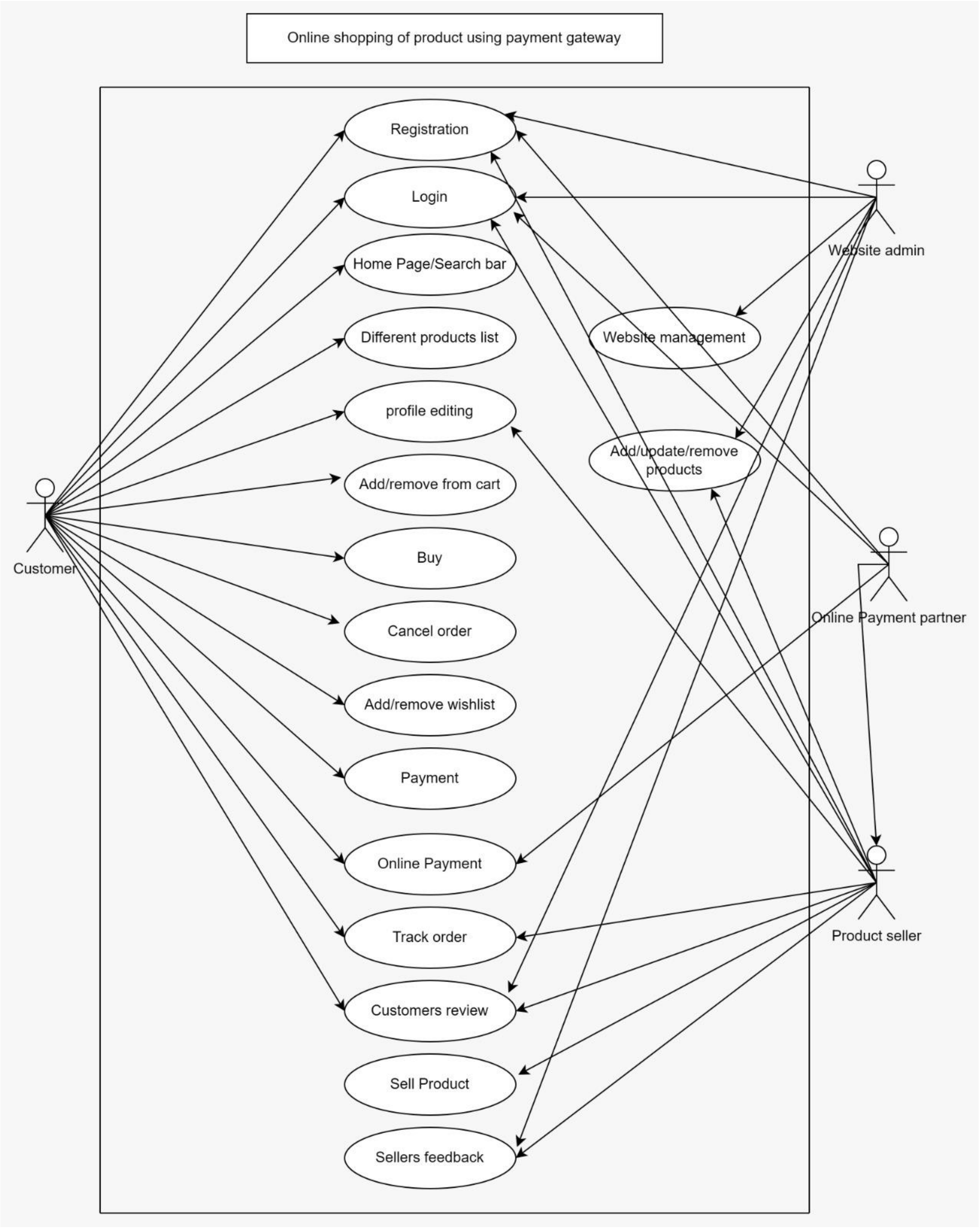
**Q.19 Draw usecase on online shopping product using COD.**

**Ans.**


Online shopping of product using COD
Registration
Login
HomePage
Profile management
Different Product list
Add/remove from cart
Add/remove wishlist
Buy
Payment
COD
Track order
Feedback
Customer reviews
Sell products
Customer contact/address
Seller contact/address
Website management
Add/update/remove products
Customer
Admin
Product seller
Delivery partner

**Q.20 Draw usecase on online shopping product using payment gateway.**

**Ans.**


Online shopping of product using payment gateway
Registration
Login
Home Page/Search bar
Different products list
profile editing
Add/remove from cart
Buy
Cancel order
Add/remove wishlist
Payment
Online Payment
Track order
Customers review
Sell Product
Sellers feedback
Website management
Add/update/remove products
Customer
Website admin
Online Payment partner
Product seller