

# ASSIGNMENT

## MODULE 1

### 1. What is SDLC?

**Ans:** The **Software Development Life Cycle (SDLC)** is a systematic process for developing software that ensures high quality and efficiency.

### 2. What is Software Testing?

**Ans:** Software Testing is a process of verifying and validating whether the Software Product is working as a requirement of client or not.

### 3. What is Agile methodology?

**Ans:** The Agile SDLC model combines step-by-step development and continuous improvements.  
It focuses on flexibility, customer needs, and quick delivery of working software.

### 4. What is SRS?

**Ans:** SRS is stands for **Software Requirements Specification (SRS)** .  
It serves as a guide for developers, testers, and clients to understand what the software should do.

### 5. Write SDLC phases with basic introduction

#### **SDLC Phases:**

- a) REQUIREMENT COLLECTION / GATHERING
- b) ANALYSIS
- c) DESIGN

d) IMPLEMENTATION

e) TESTING

f) DEPLOYMENT

g) MAINTENANCE

**a) Requirement Collection / Gathering**

Requirement Collection/Gathering is the process of identifying and documenting the needs and expectations of clients.

**b) Analysis**

The phase where you analyze the needs and expectation of client and users to ensure the final products meet their needs.

**c) Design**

The phase where the design/structure is prepared according to the requirements.

**d) Implementation**

The Implementation phase translated into functional software by writing code.

**e) Testing**

The phase where ensuring the requirements, function, correctly and free of defects.

**f) Deployment**

Deployment is the process of releasing a software application for users to access and use.

Google Play Store, Appstore, etc.

**g) Maintenance**

The phase where focusing on fixing bugs, improving performance and adapting to changing user needs after deployment software.

## 6. Explain Phases of the waterfall model

### Waterfall Model phases

- a) REQUIREMENT COLLECTION / GATHERING
- b) ANALYSIS
- c) DESIGN
- d) IMPLEMENTATION
- e) TESTING
- f) DEPLOYMENT
- g) MAINTENANCE

#### **a) Requirement Collection / gathering**

Requirement Collection/Gathering is the process of identifying and documenting the needs and expectations of clients.

#### **b) Analysis**

The phase where you analyze the needs and expectation of client and users to ensure the final products meet their needs.

#### **c) Design**

The Design phase the design/structure is prepared according to the requirements.

#### **d) Implementation**

The Implementation phase translated into functional software by writing code.

#### **e) Testing**

The Testing phase ensuring the requirements, function, correctly and free of defects.

**f) Deployment**

The Deployment phase marks the official release of the software to users.

**g) Maintenance**

Maintenance phase is software needs update and bug fixes after deployment.

**7. Write phases of Spiral Model**

a) PLANNING

b) RISK ANALYSIS

c) ENGINEERING

d) CUSTOMER EVALUATION

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

How Agile Works?

- Developers build the software in small sections.
- Each section is tested and reviewed before moving forward.
- Changes can be made anytime
- Teams and customers work together to improve the product.
- Software gets updated frequently, instead of a one-time release.

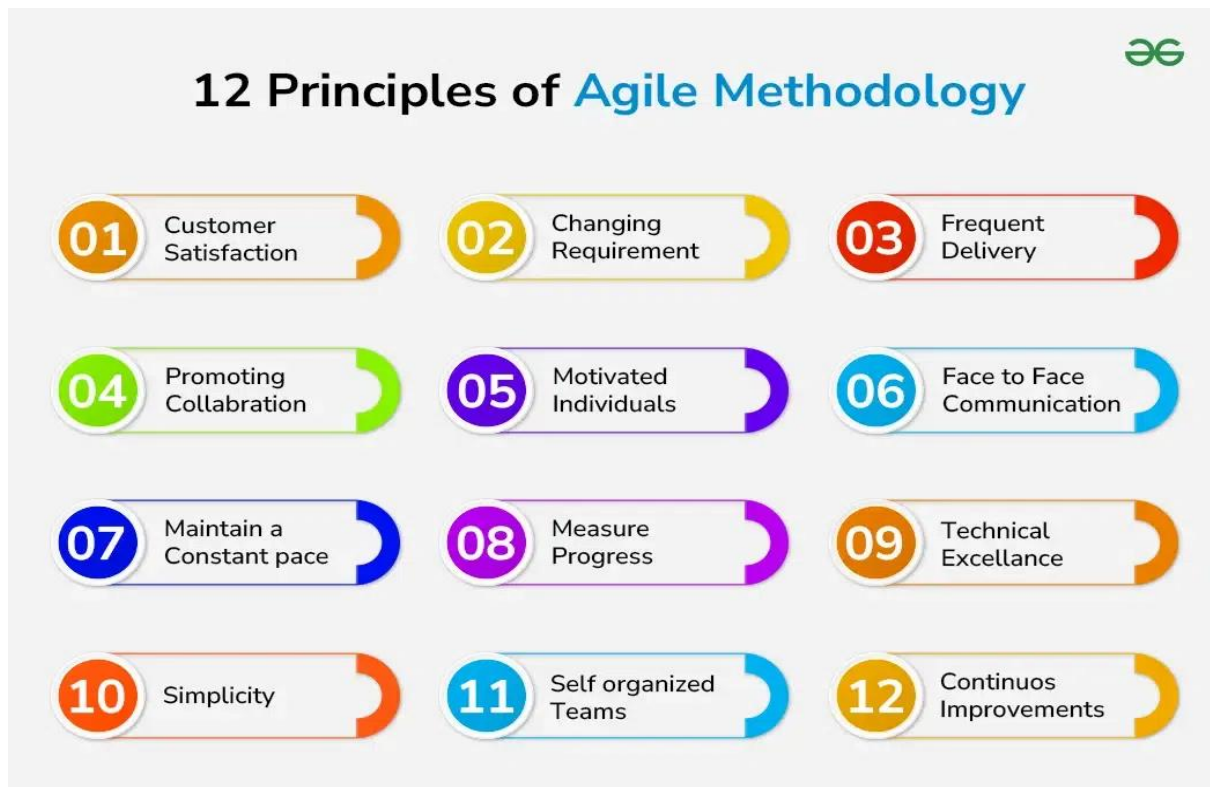
Pros:

- **Quick updates** – Faster improvements and bug fixes.
- **Flexible changes** – Easy to adjust features based on user needs.
- **Better teamwork** – Developers and customers work closely.
- **High customer satisfaction** – Users get what they actually need.

Cons:

- **Needs active teamwork** – Everyone must communicate regularly.
- **Less predictable** – Planning can change often.
- **Takes more effort** – Requires skilled developers to handle flexibility

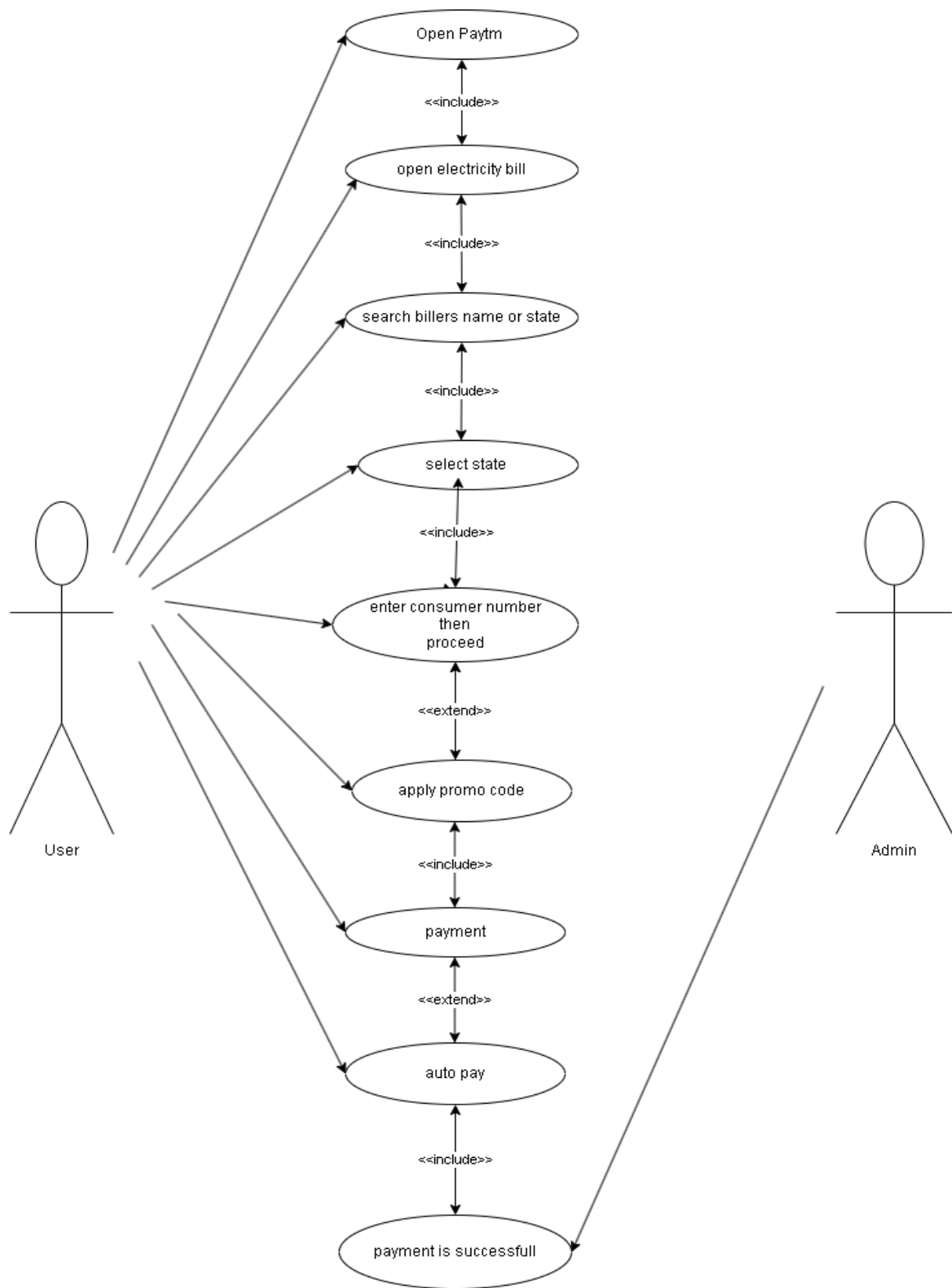
## 9. Write agile manifesto principles.



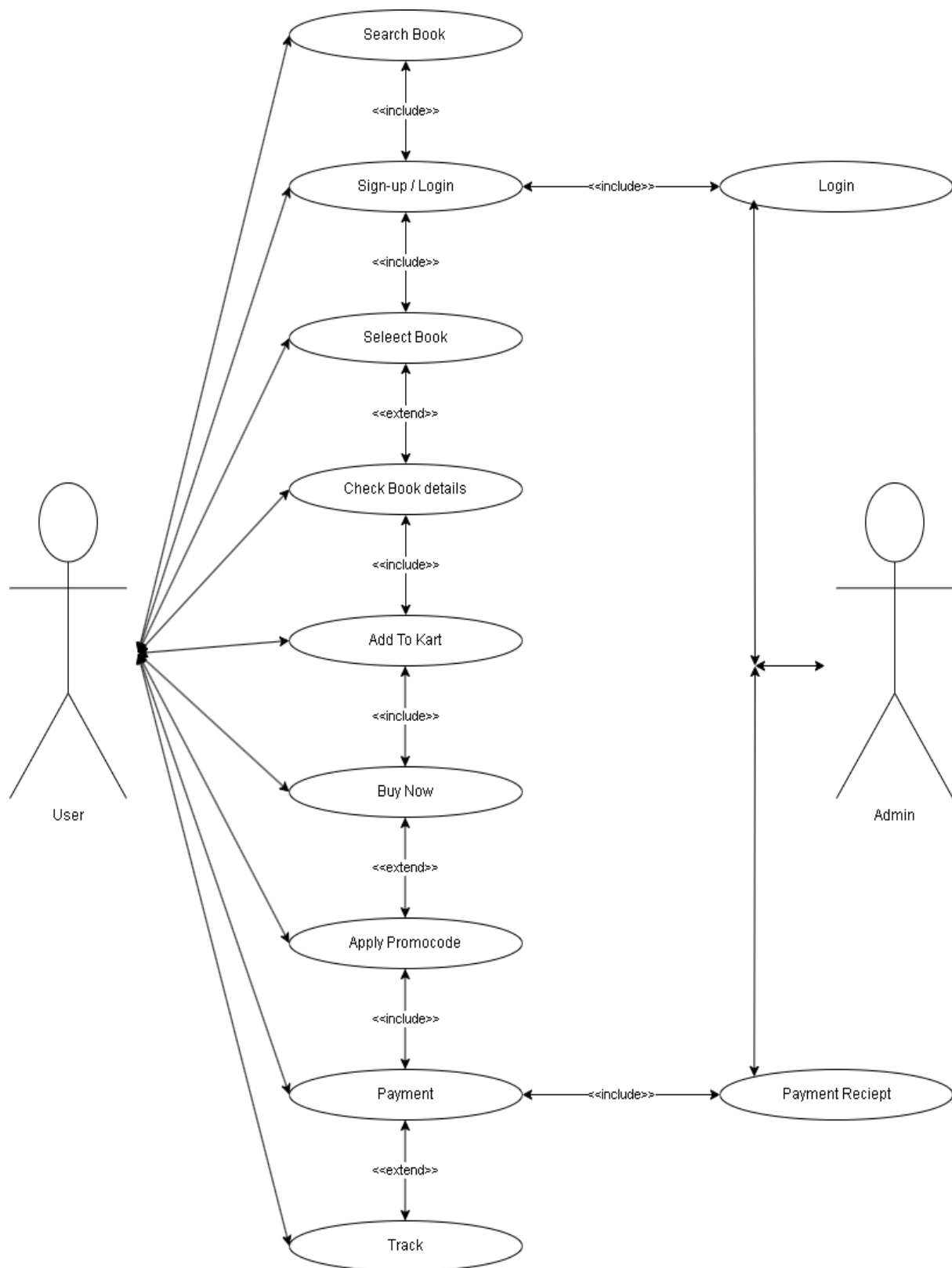
- I. **Customer satisfaction** – Deliver what the customer needs as quickly as possible.
- II. **Changing Requirement** – Be open to changing requirements, even late in the process.
- III. **Frequent Delivery** – Release small, working versions frequently, not just one big final version.
- IV. **Promoting collaboration** – Developers and business people should communicate and collaborate daily.

- V. **Trust and motivation** – Give teams the support they need and trust them to do their job well.
- VI. **Face-to-face communicate** – Talking directly is the most effective way to communicate.
- VII. **Measure progress** – The product should always function properly.
- VIII. **Sustainable development** – Keep a pace that can be maintained over time without burnout.
- IX. **Continuous attention to quality** – Focus on good design and technical excellence.
- X. **Simplicity** – Do only what's necessary; avoid overcomplicating things.
- XI. **Self-organizing teams work best** – Teams should make their own decisions on how to do the work.
- XII. **Continuous improvement** – Teams should review their work and improve continuously.

### 10. Draw Use-case on online bill payment system (Paytm)

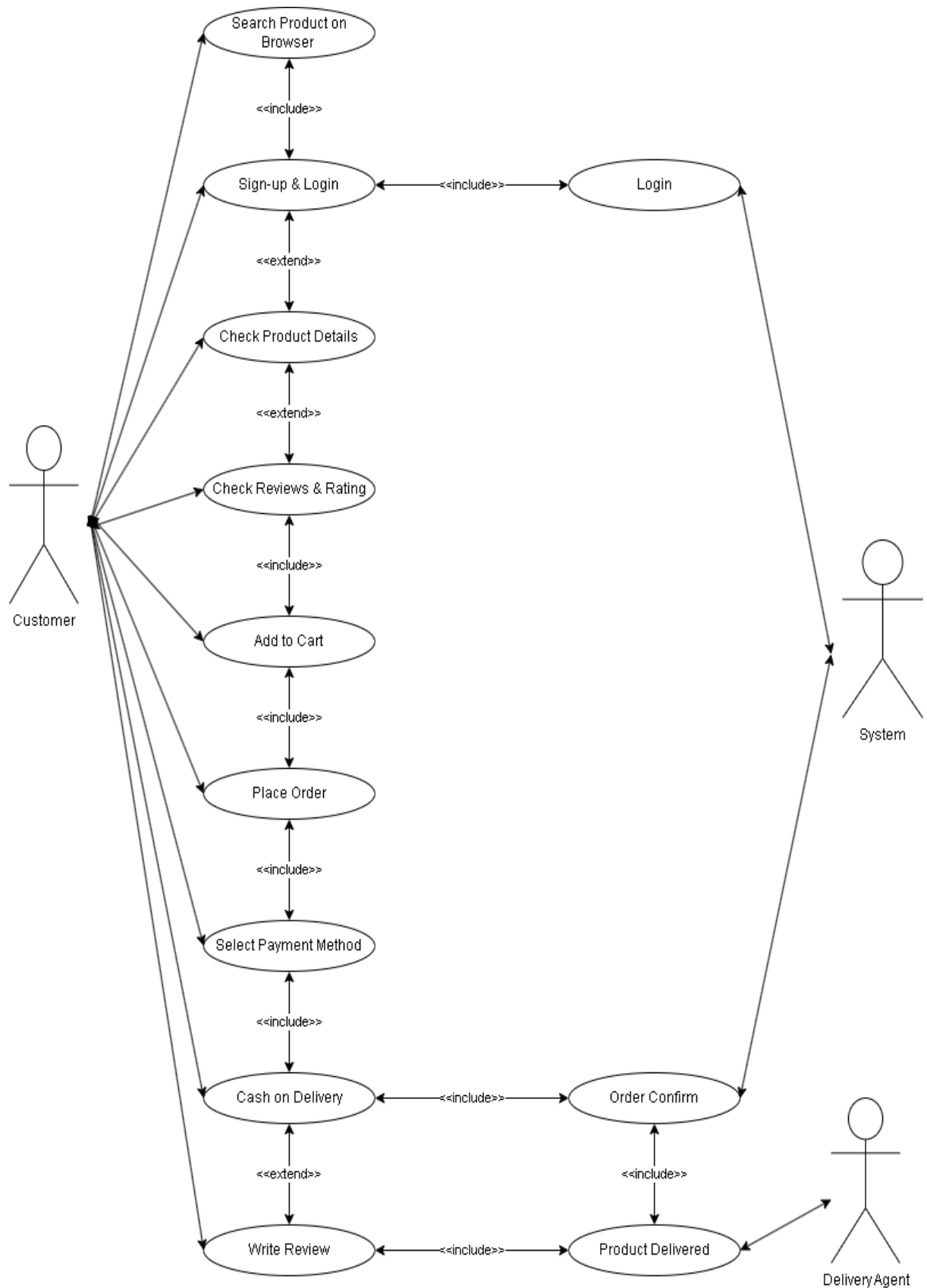


### 11. Draw Use-case on Online book shopping.

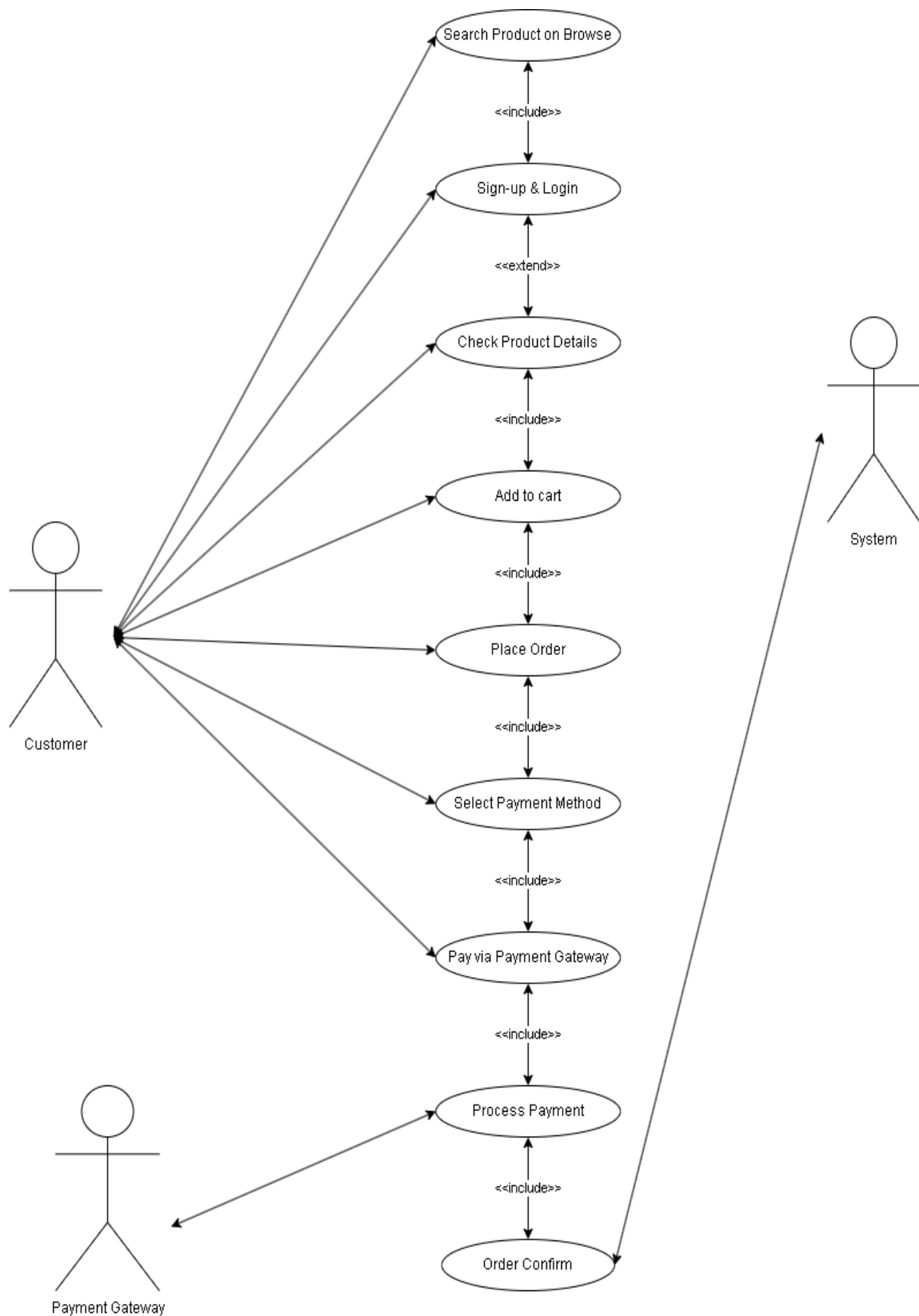




## 12. Draw use-case on Online shopping product using COD.



**13. Draw use case on Online shopping product using payment gateway.**



#### **14. What is OOPS?**

**Ans:** Object-Oriented Programming System (OOPS) is a programming model that organizes code using objects, which combine data and behavior.

#### **15. Write Basic Concepts of OOPS.**

- a) OBJECT
- b) CLASS
- c) ENCAPSULATION
- d) ABSTRACTION
- e) POLYMORPHISM
- f) INHERITANCE

#### **16. What is Class?**

**Ans:** It is collection of data member and member function.

#### **17. What is Object?**

**Ans:** Object gives permission to access functionality.

#### **18. What is Encapsulation?**

**Ans:** Encapsulation is as the wrapping up of data under a single unit.

#### **19. What is Inheritance?**

**Ans:** Transfer inherited code from base class to derived class.

#### **20. What is Polymorphism?**

**Ans:** polymorphism as the ability of a message to be displayed in more than one form.