What is SDLC

SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support.

Main Goals of SDLC:

- To understand customer requirements clearly
- To plan and manage the project efficiently
- To ensure quality through systematic testing
- To deliver a reliable, maintainable, and scalable product

Common SDLC Models:

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Agile Model

What is software testing?

Software testing is the process of identifying defects or errors in a software system to ensure it meets the specified requirements and produces the desired output.

Types of Testing:

- Unit Testing
- Integration Testing
- System Testing
- Acceptance Testing

What is agile methodology?

Agile methodology is an adaptive approach to software development where the project is divided into small, manageable units called **iterations** or **sprints**. Each iteration results in a working product that is reviewed and refined based on customer feedback.

What is SRS

SRS (Software Requirements Specification) is a formal document that provides a comprehensive description of a software system's functional (what it does) and non-functional (how it performs) requirements. It serves as a blueprint for developers and a formal agreement between stakeholders, ensuring everyone has a shared understanding of the project's goals, scope, and intended behavior, which is crucial for project success.

What is oops

OOPS (Object-Oriented Programming System) is a programming paradigm that organizes software design around **objects** (data) rather than functions and logic. It promotes reusability, scalability, and maintainability.

Main Goals:

- Model real-world entities
- Promote code reusability
- Improve software structure and maintenance

Write Basic Concepts of oops

- **Class** – Blueprint or template that defines data members and functions.
- **Object** – Instance of a class.
- **Encapsulation** – Wrapping data and code together.
- **Abstraction** – Hiding unnecessary details and showing only essential features.
- **Inheritance** – Deriving new classes from existing ones.
- **Polymorphism** – Ability to take multiple forms (method overriding and overloading).

What is object

An **object** is an instance of a class that represents a real-world entity with **state (data)** and **behavior (methods)**.
Example: `Car c = new Car();`

What is class

A class is a blueprint or template used to create objects.
It defines attributes and behaviors of the object.
Example:

```
class Student { String name; void study(){} }
```

What is encapsulation

To wrap code and data into a single unit is called encapsulation.

Example: private variables with getters/setters.

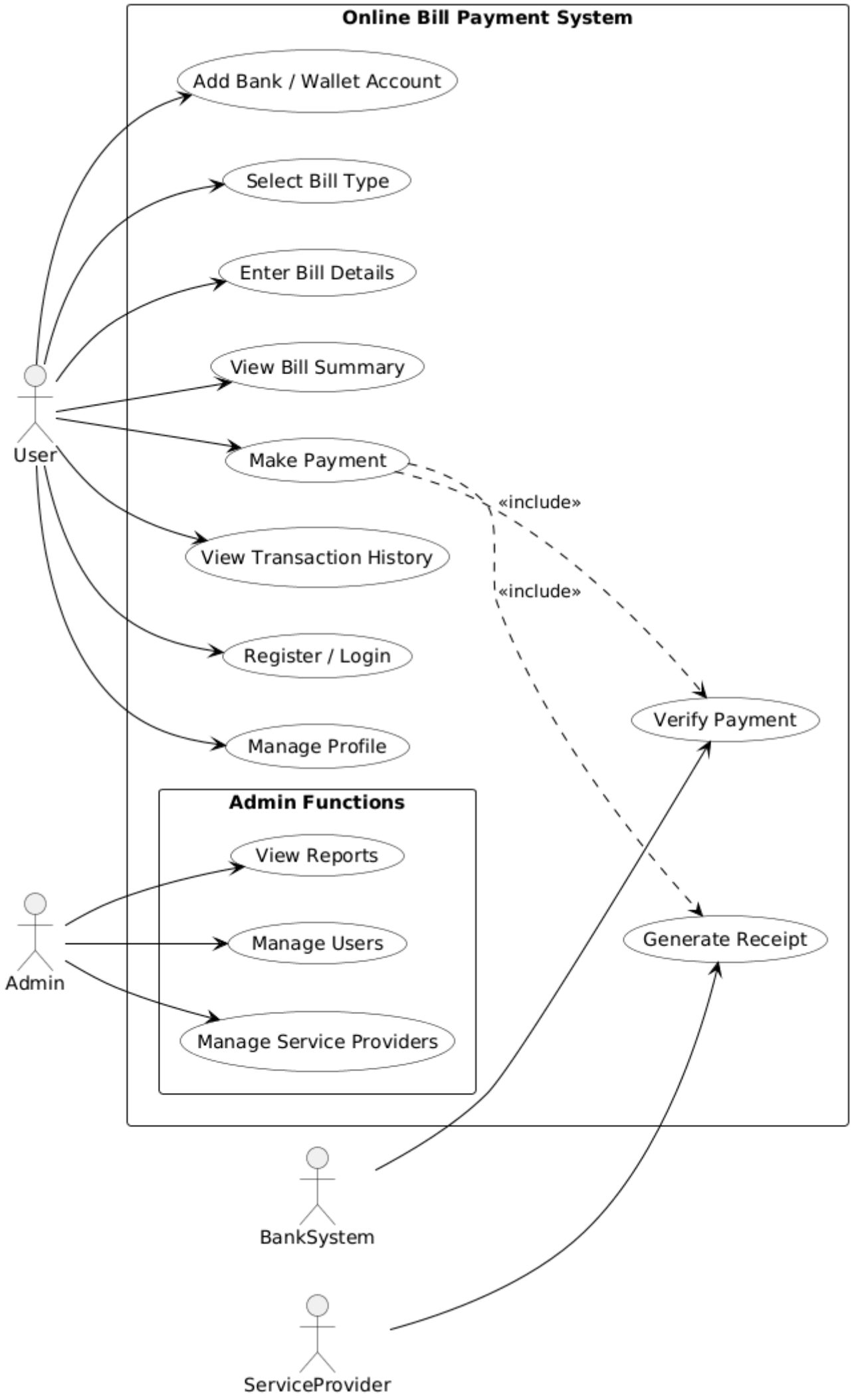
What is inheritance

The object of one class can acquire the properties of another class is called inheritance.
Creating a new class from an existing class is called an inheritance.

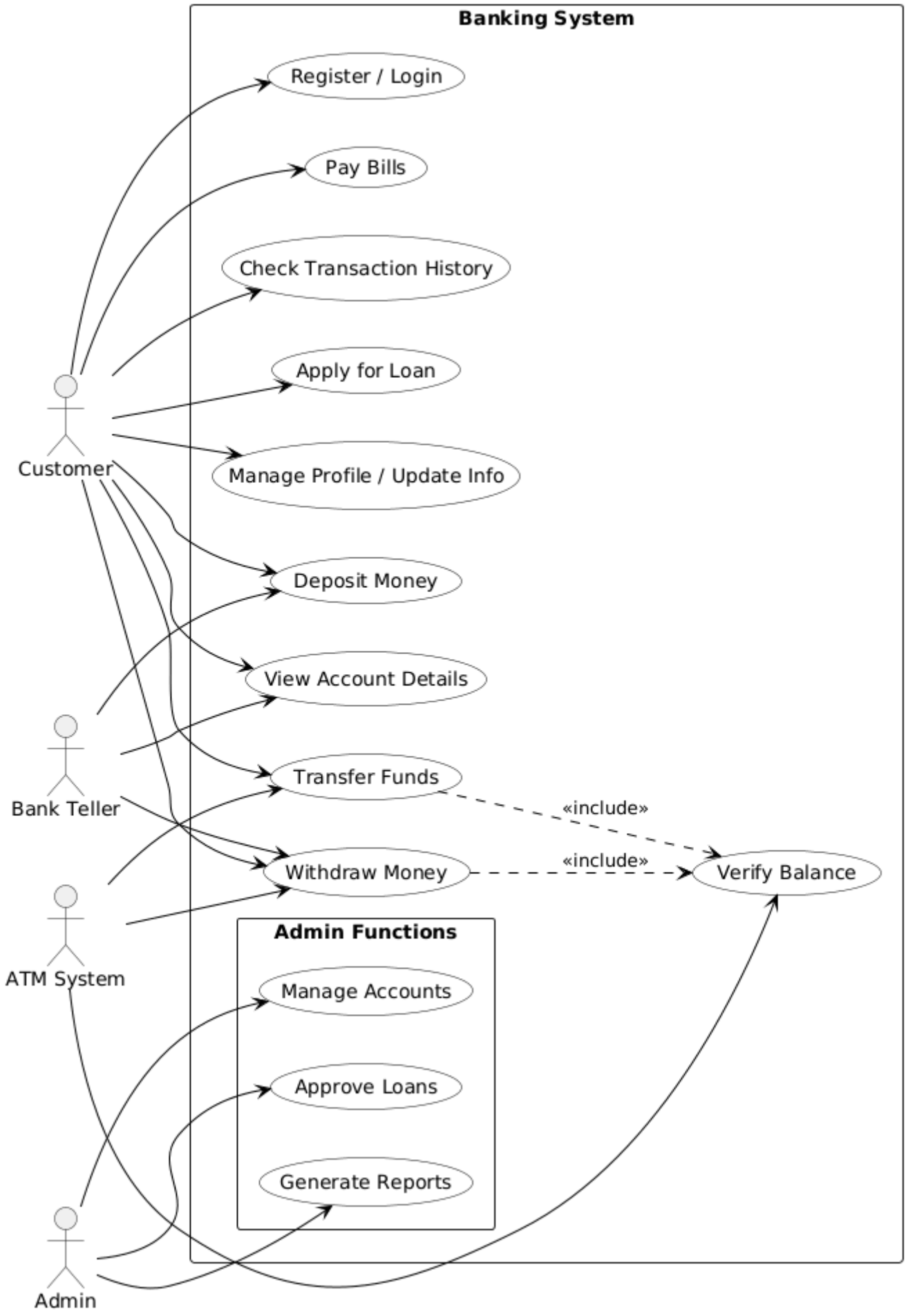
- **What is polymorphism**

Polymorphism means one function behaves differently for different objects.
It allows **method overloading** and **method overriding** in OOP.

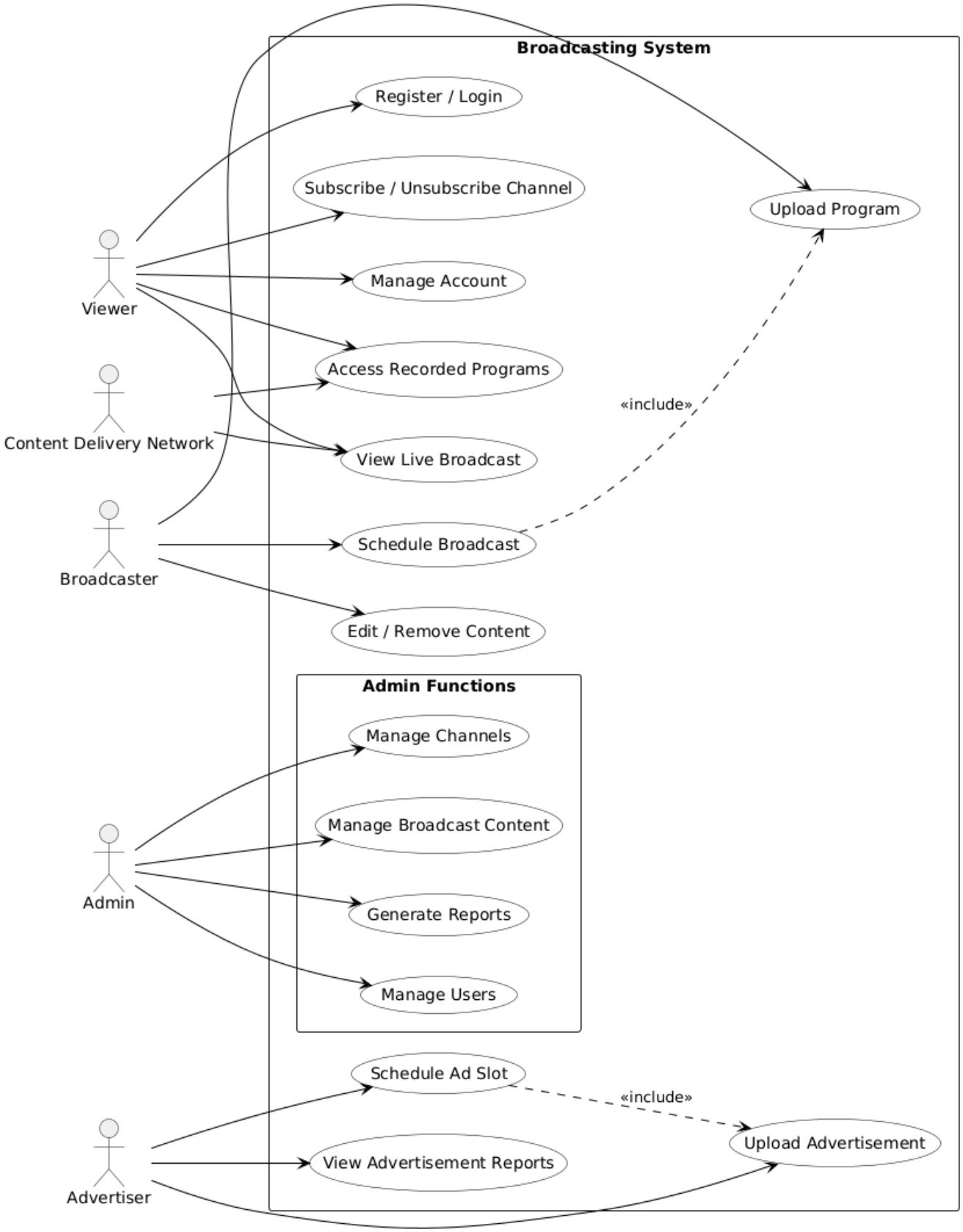
- **Draw Use case on online bill payment system (paytm)**


Online Bill Payment System
Add Bank / Wallet Account
Select Bill Type
Enter Bill Details
View Bill Summary
Make Payment
View Transaction History
Register / Login
Manage Profile
«include»
«include»
Verify Payment
Generate Receipt
Admin Functions
View Reports
Manage Users
Manage Service Providers
User
Admin
BankSystem
ServiceProvider

- **Draw Use case on banking system for customers**



- **Draw Use Case on Broadcasting System.**



● Write SDLC phases with basic introduction

SDLC (Software Development Life Cycle) is a process used to design, develop, test, and maintain software systematically.
It includes the following main phases:

- **Requirement Analysis:**
  Collect and analyze the needs of the customer and document them as SRS (Software Requirement Specification).
- **System Design:**
  Prepare design documents like architecture, data flow, and database design based on requirements.
- **Implementation (Coding):**
  Developers write code using suitable programming languages following coding standards.
- **Testing:**
  The developed software is tested to find and fix defects ensuring quality and correctness.
- **Deployment:**
  The tested product is delivered or installed in the client environment for actual use.
- **Maintenance:**
  After deployment, software is updated to fix issues, improve performance, or add new features.

● Explain Phases of the waterfall model

The Waterfall Model is a linear and sequential approach to software development.
Each phase must be completed before moving to the next one.

Phases:

1. **Requirement Gathering and Analysis:**
   Collect all functional and non-functional requirements from the customer.
2. **System Design:**
   Prepare software architecture, data design, and interface design.
3. **Implementation (Coding):**
   Actual code is written according to the design documents.
4. **Integration and Testing:**
   Different modules are integrated and tested as a complete system.
5. **Deployment:**
   Software is installed and made operational in the user's environment.
6. **Maintenance:**
   After delivery, necessary changes, bug fixes, and updates are done.

Features:

- Simple and easy to understand.
- Suitable for small and well-defined projects

● Write phases of spiral model

The **Spiral Model** is a risk-driven software development model that combines iterative development with systematic risk analysis.

Phases of Spiral Model:

1. **Planning Phase:**
   Objectives, alternatives, and constraints of the project are identified.
2. **Risk Analysis Phase:**
   Risks are identified, analyzed, and strategies are planned to reduce them.
3. **Engineering Phase:**
   Software is developed, tested, and validated in incremental versions.
4. **Evaluation Phase:**
   The customer evaluates the output, and feedback is used to plan the next iteration.

Use:
Best suited for large, complex, and high-risk projects.

● Write agile manifesto principles

The **Agile Manifesto** consists of **12 key principles** that guide agile software development:

- Customer satisfaction through early and continuous delivery
- Welcome changing requirements
- Deliver working software frequently
- Close collaboration between business and developers
- Build projects around motivated individuals
- Face-to-face communication is most effective
- Working software is the main measure of progress
- Maintain a sustainable development pace
- Continuous attention to technical excellence
  Simplicity – maximize the amount of work not done
- Self-organizing teams produce best results
- Regular reflection and improvement by the team

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

Working Methodology:

- The project is divided into short development cycles called **sprints** (2–4 weeks).
- Each sprint involves **planning, design, coding, testing, and review**.
  Teams hold **daily stand-up meetings** to discuss progress.
- At the end of each sprint, a **working product increment** is delivered to the client.
- Customer feedback is taken for continuous improvement.

Advantages (Pros):

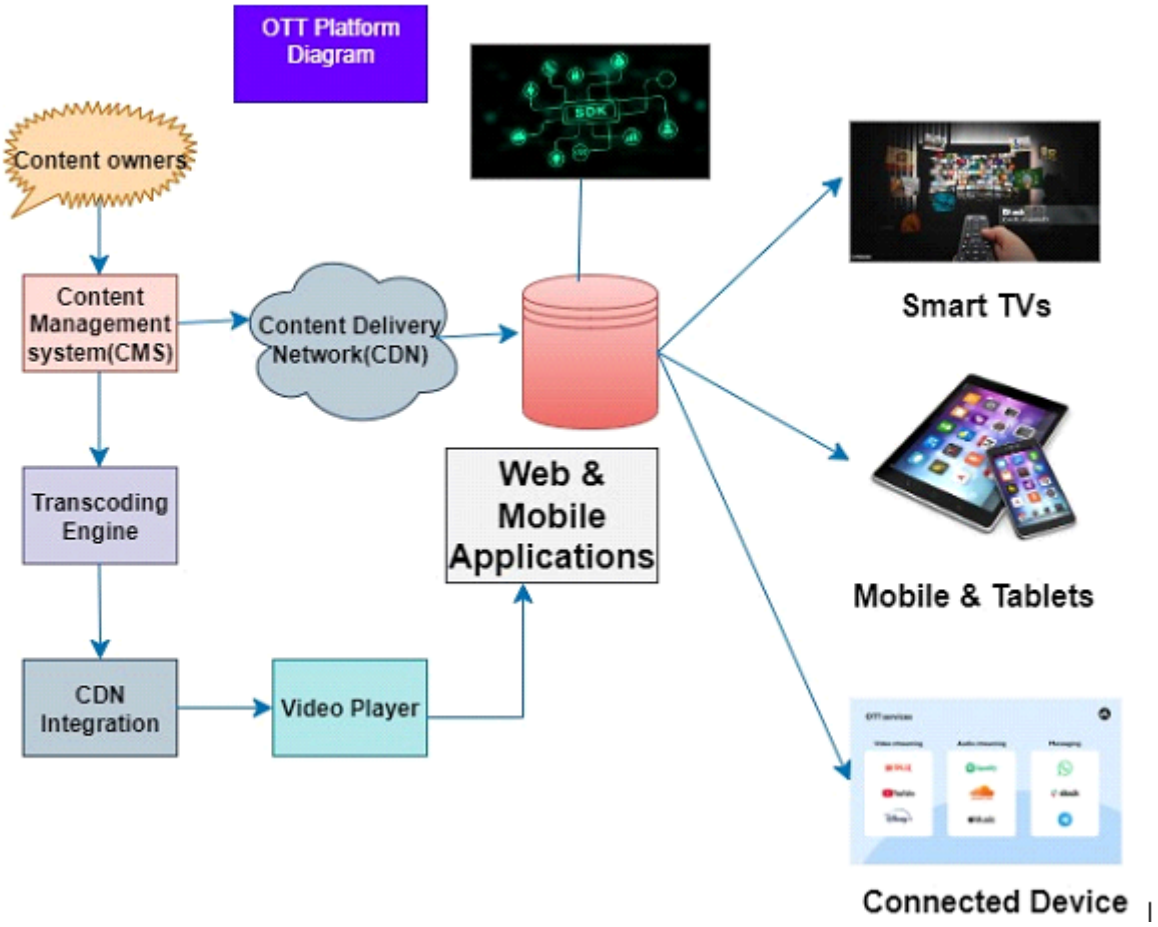
- Flexible to changing requirements
- Early and continuous delivery of working software

- Encourages teamwork and customer satisfaction
- Detects problems early due to frequent testing

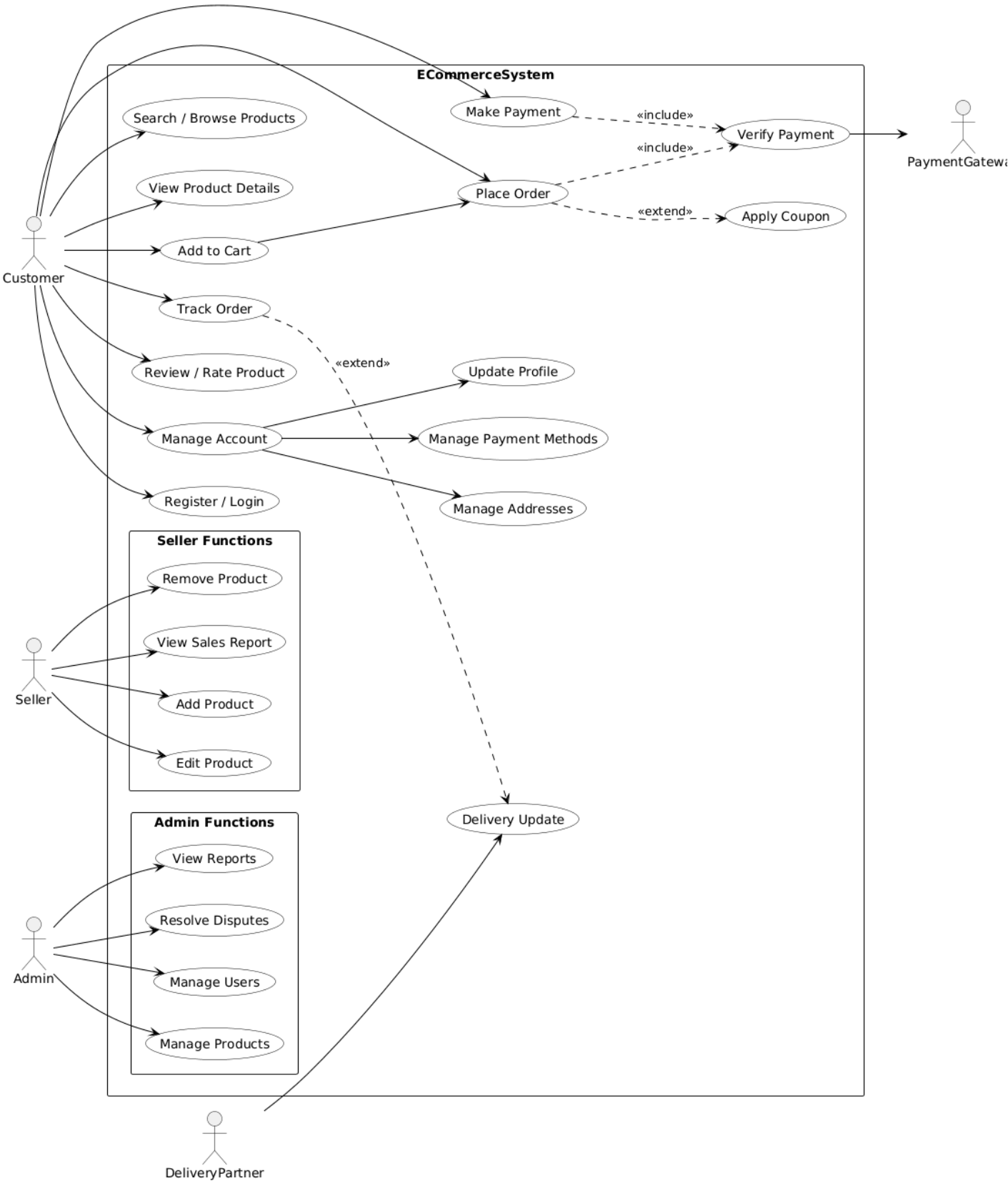
Disadvantages (Cons):

- Requires experienced and skilled team
- Less focus on documentation
- Hard to estimate time and cost accurately
- Not ideal for very large or fixed projects

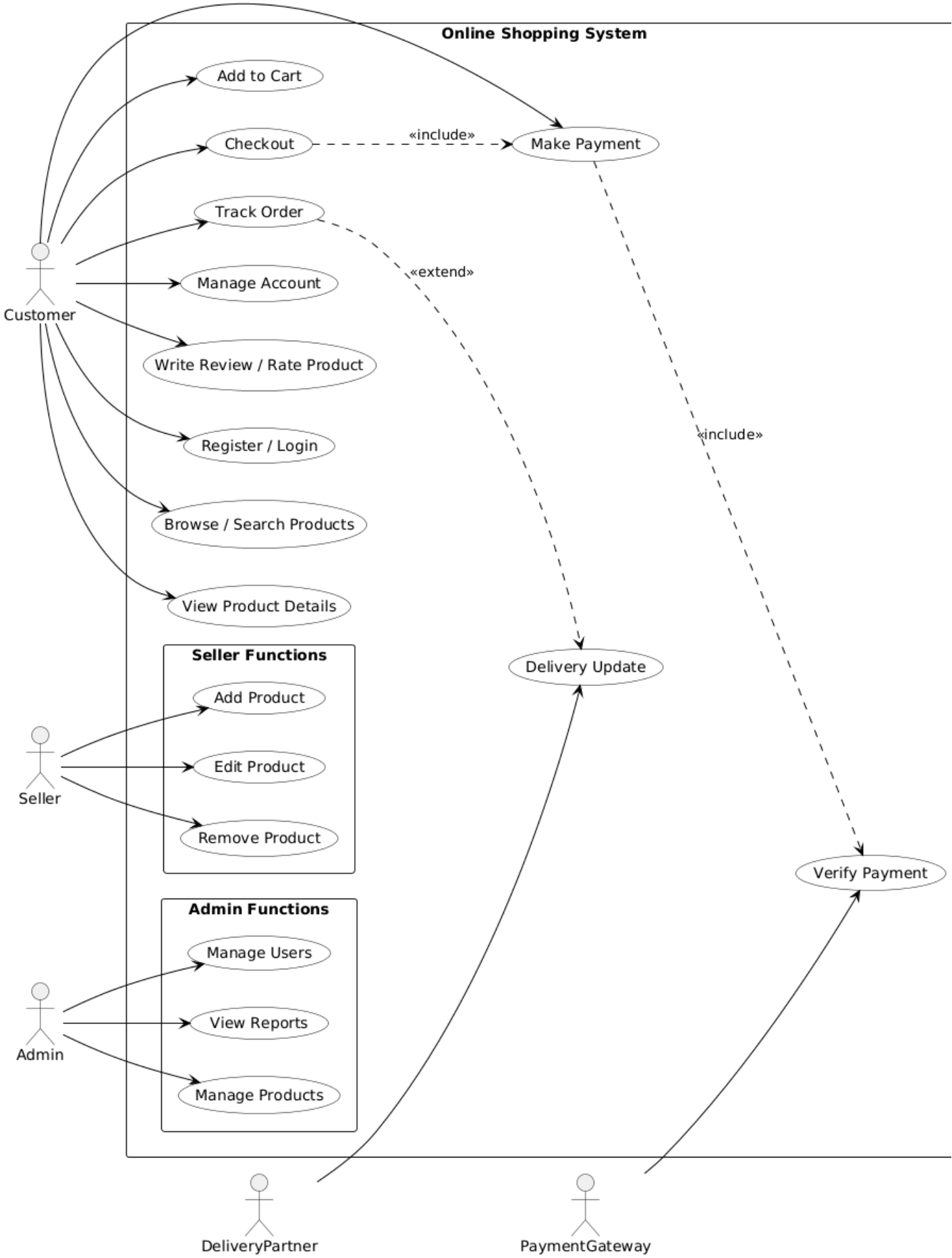
- **Draw use case on OTT Platform.**



- **Draw use case on E-commerce application**



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


Online Shopping System
Add to Cart
Checkout
«include»
Make Payment
Track Order
«extend»
Manage Account
Customer
Write Review / Rate Product
Register / Login
«include»
Browse / Search Products
View Product Details
Seller Functions
Delivery Update
Add Product
Edit Product
Seller
Remove Product
Verify Payment
Admin Functions
Manage Users
Admin
View Reports
Manage Products
DeliveryPartner
PaymentGateway