1) What is SDLC ?

- SDLC stands for Software Development Life Cycle.

- In simple terms, it's a step-by-step process used to plan, build, test, and deliver software.

- Basic Steps in SDLC:

i. Requirement Gathering
ii. Analysis
iii. Design
iv. Implementation
v. Testing
vi. Maintenance

2) What is software testing ?

- SoftwareTesting is a process used to identify the correctness, completeness, and quality of developed computer software.

3) What is agile methodology ?

- Agile is a flexible way of developing software where the work is divided into small parts called iterations (1–3 weeks).

- In each iteration, a team plans, designs, codes, and tests a part of the product.

- After every cycle, a working version is shown to the customer for feedback.

- This helps teams quickly adapt to changes and deliver features step-by-step, keeping the customer involved and satisfied.

Pros:

- Customer Involvement:
  Regular feedback helps meet customer needs better.

- Faster Delivery:
  Working software is delivered in short cycles.

- Flexibility:
  Easy to adapt to changes at any stage.

- Improved Quality:
  Continuous testing and feedback improve product quality.

- Team Collaboration:
  Cross-functional teams work closely, boosting communication.

Cons:

- Less Predictable:
  Final product features or timelines may change often.

- Requires Experience:
  Teams need to be skilled and self-organized.

- Not Ideal for Big Documentation:
  Less focus on detailed documentation.

- Scope Creep Risk:
  Frequent changes can lead to unclear project scope.
- Customer Dependency:
  Needs active customer involvement throughout.

4) What is SRS ?

- An SRS is a clear and detailed document that explains what a software system should do.
- It includes all the requirements both functional (what the system should do) and non-functional (how it should perform).
- It acts like a blueprint for developers, testers, and clients to understand exactly what needs to be built.

5) What is oops ?

- OOPs stands for Object Oriented Programming System.
- It’s a programming paradiagm based on the concepts of objects, which can contain data and code.

6) Write Basic Concepts of oops.

- Class
- Object
- Encapsulation
- Polymorphism
- Abstraction

7) What is object ?

- An object is an instance of a class. It represents a real-world entity that contains both data and behavior.
- While a class is just a blueprint, an object is the actual thing created from that blueprint.

8) What is class ?

- Class is a structure in which we can have member function and member variable.
- In simple term

    " A blueprint for creating objects."

- It defines a type of object according to the data and methods it should have.

9) What is encapsulation ?

- Encapsulation is one of the fundamental principles of object oriented programming.
- To wrapping data into single unit is called encapsulation.

10) What is inheritance ?

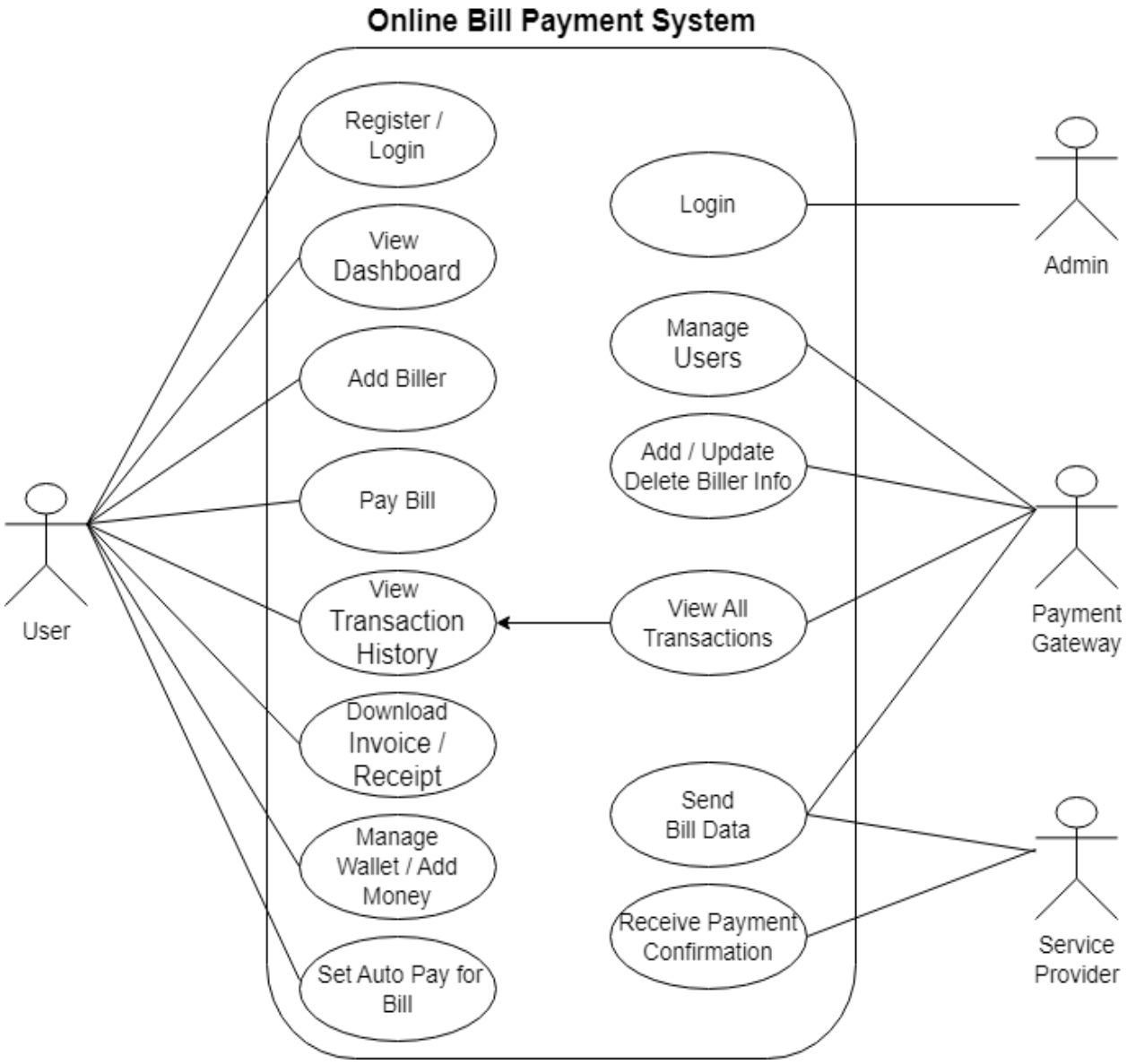
- To access propertyof one class to another class that is inheritance.
- 5 types of Inheritance :

i. Single Inheritance
ii. Multilevel Inheritance
iii. Multiple inheritance
iv. Hierarchical Inheritance
v. Hybride Inheritance

11) What is polymorphism ?

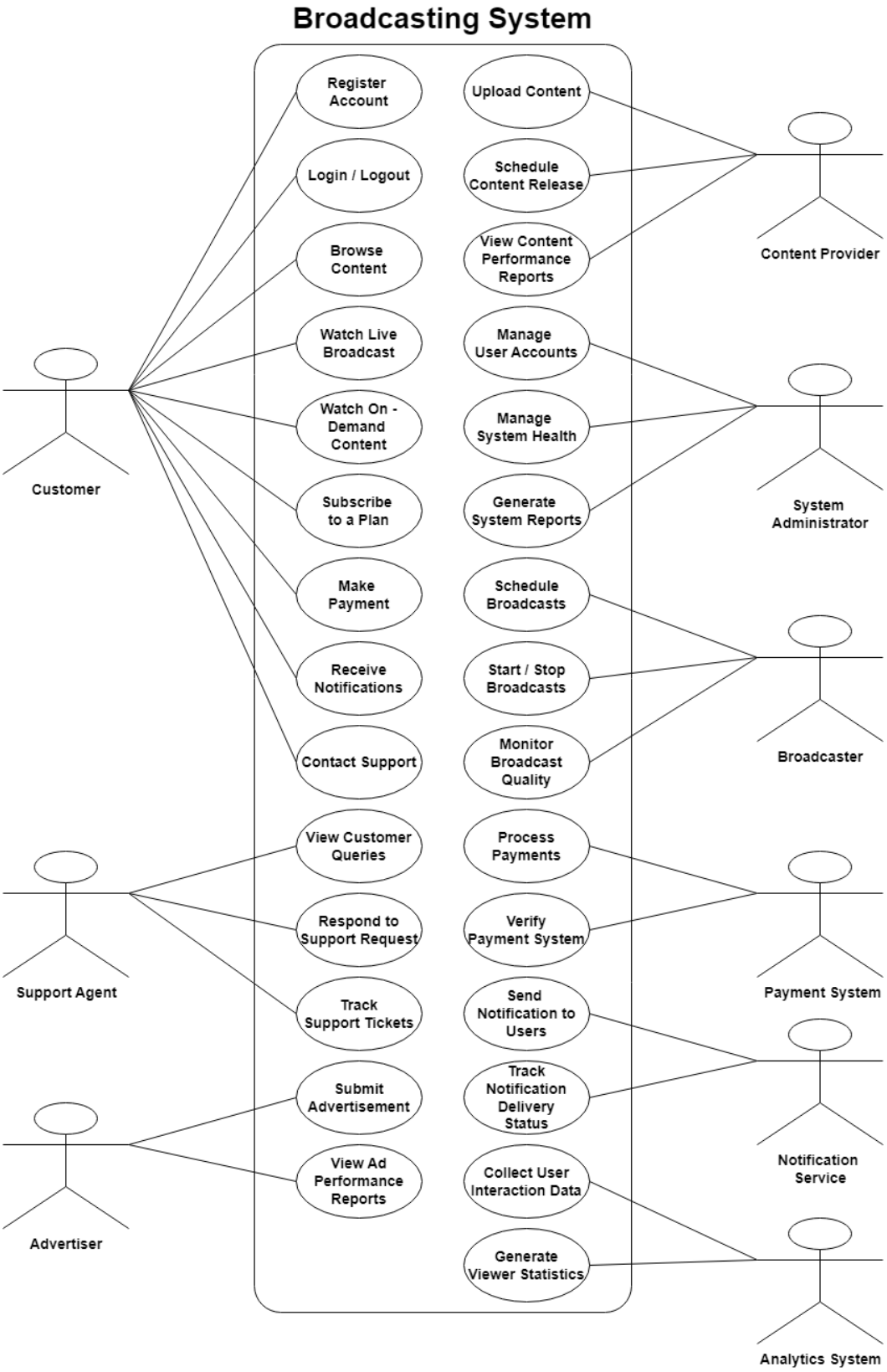
- Polymorphism is nothing but same function name but having different functionalities that called polymorphism.
- Polymorphism promotes code reusability, enhances flexibility and scalability, and enables loose coupling between classes.
- 2 types of Polymorphism :
    i. Overloading
    ii. Overriding

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



13)Draw Usecase on banking system for customers.

14) Draw Usecase on Broadcasting System.



15) Write SDLC phases with basic introduction.

- SDLC stands for Software Development Life Cycle.

- SDLC is a step-by-step process used to develop high-quality software in a structured, cost-effective, and timely manner.

SDLC Phases:

i. Requirement Gathering
ii. Analysis
iii. Design
iv. Implementation
v. Testing
vi. Maintenance

i. Requirement Gathering:

The team collects all the software requirements from the client or users.

ii. Analysis:

The gathered requirements are analyzed for clarity, feasibility, and risks.

iii. Design:

The system's structure is designed, including UI, database, and data flow.

iv. Implementation:

Developers write the code and build the software based on the design.

v. Testing:

Testers check the software for bugs and make sure it works properly.

vi. Maintenance:

After release, the software is updated, bugs are fixed, and improvements are made.

16) Explain Phases of the waterfall model.

- The Waterfall model is linear and sequential approach to software development. It is step-by-step process, where each phase must be completed before moving to the next.

i) Requirement Gathering:

⇒Collect and understand the needs of the client or end-user.

⇒Document the functional and non-functional requirements in the SRS (Software Requirement Specification) documents.

ii) Analysis:

⇒In this phase, the requirements are analysed to define "what" the system needs to achieve.

⇒Feasibility studies are conducted to assess technical and financial visibility.

⇒A Software Requirement Specification (SRS) document is prepared.

iii) Design:

⇒The systems architecture and design are planned based on requirements.

⇒This includes defining the system structure, database design, UI/UX design, and technical components.

⇒High level design (HLD) and Low-level design (LLD) documents and created.

iv) Implementation:

⇒Developers write the actual code based on the design specifications.

⇒This phase involves front-end, back-end and database development.

⇒Version control and coding best practices are followed.

v) Testing:

⇒This Testing Phase ensures the software works correctly and meets requirements.

⇒A separate team tests the system to find and fix errors. It includes different types of testing, such as:

1. Regression Testing
2. Internal Testing
3. Unit Testing
4. Application Testing
5. Stress Testing

vi) Maintenance:

⇒After deployment, the software is monitored for issues, bugs, and performance improvements.

⇒Updates, patches, and enhancements are made to adapt the system to new requirements and ensure smooth operation.

17) Write phases of spiral model.

- Spiral Model is very widely used in the software industry as it is in synch with the natural development process of any product i.e. learning with maturity and also involves minimum risk for the customer as well as the development firms.

i) Planning Phase:

⇒Requirements are gathered from the customer.

⇒Objectives, alternatives, and constraints are identified.

⇒A plan is created for the next phases.

ii) Risk Analysis Phase:

⇒Identify potential risks and uncertainties.

⇒Analyse and evaluate each risk.

⇒Purpose solutions or alternatives to reduce or eliminate risks.

⇒A prototype may be built to clarify requirements and reduce risk.

iii) Engineering:

⇒Actual development and testing of the product happens in this phase.

⇒Depending on the iteration, this can include designing, coding, testing, and integrating the software.

iv) Evaluation Phase:

⇒The customer evaluates the progress.

⇒Feedback is gathered.

⇒Decisions are made regarding the next iteration or necessary changes.

18)Write agile manifesto principles.

i. Individual and Interaction:

- People are more important than tools and processes.
- Teamwork, communication, and motivation matter a lot.
- Things like pair programming and working in the same place help better teamwork.

ii. Working Software:

- A running, working version of the software is the best way to show progress.
- Instead of only writing long documents, showing real software helps the customer understand better.

iii. Customer Collaboration:

- Its hard to know all the requirements at the beginning.
- So, keep talking to the customer regularly to understand what they want as the project moves forward.

iv. Responding to Change:

- In Agile, change is normal and welcome.
- If customer needs or market trends change, the team adjusts quickly and keeps improving the product.

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

- Agile model works in small, repeatable cycle called iterations or sprints. In each cycle, a small part of the software is planned, designed, developed, tested, and delivered.

i. Requirement gathering:

Client gives initial requirement (Not full, just the basic ideal).

ii. Planning:

The team breaks down work into small tasks and created a sprint plan.

iii. Design and Development:

Team design and codes the selected features for that sprint.

iv. Testing:

The software is tested during or right after development.

v. Delivery and Review:

A working part of the product is delivered. The customer gives feedback.

vi. Next Iteration:

Based on feedback, the next sprint starts with changes or new features.

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 and changing requirement.

⇒ Minimal rules, documentation easily employed.

⇒ Little or no planning requirement.

⇒ Easy to manage.

⇒ Gives flexibility to developers.

Cons:

⇒ Not suitable for handling complex dependencies.

⇒ More risk of suitability, maintainability and extensibility.

⇒ Needs Strong Leadership

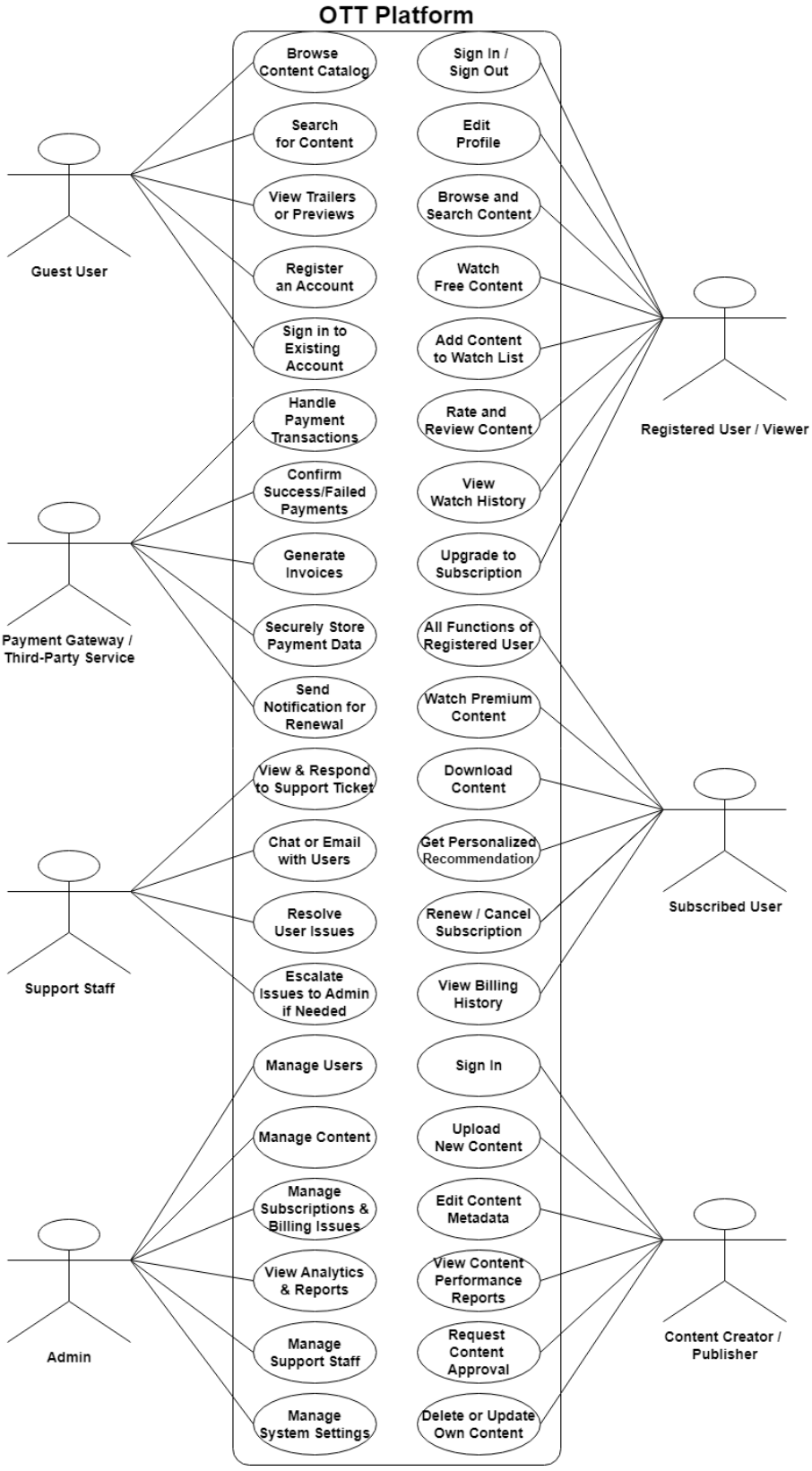
⇒ Strict Deadlines

⇒ High Customer Dependency

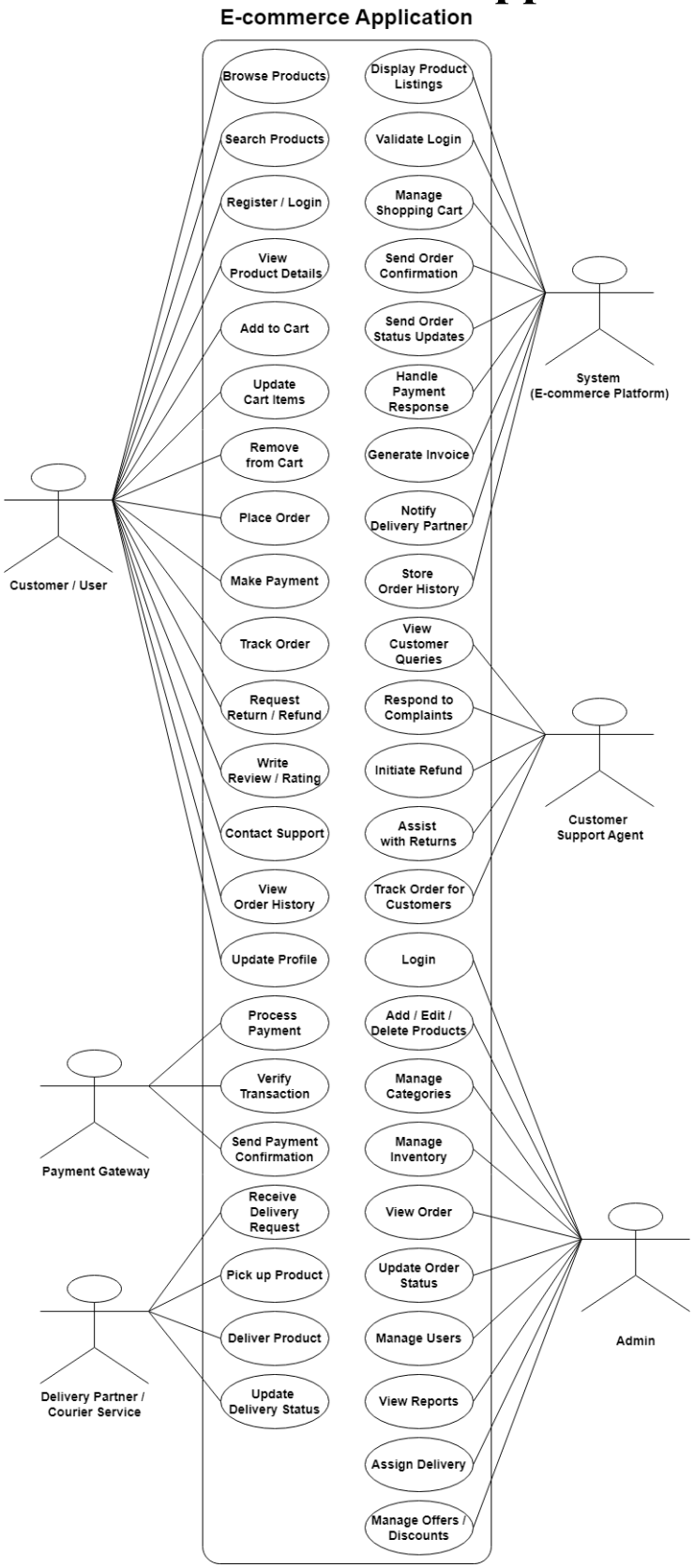
⇒ Less Documentation

⇒ Difficult Knowledge Transfer

20) Draw usecase on OTT Platform.



21) Draw usecase on E-commerce application.



21) Draw usecase on Online shopping product using payment gateway.