Question 1. What is SDLC?

=> SDLC is a structure imposed on the development of a software product that defines the process for planning, testing, Documentation, deployment, and on-going maintenance and support.

=> This Contains Six Phases Which Are hear under-

1) Requirement Gathering

2) Analysis

3) Design

4) Implementation

5) Testing

6) Maintenance

Question 2. What is software testing?

=> Software tasting is a process used to identify the correctness, completeness, and quality of developed Computer software.

=> In Other Words software testing is the process of evaluating a system or its components with the intent to find that whether it satisfies the specified requirements or not.

Question 3. What is agile methodology?

=> Agile SDLC model is combination of iterative and incremental process model with focus on process adaptability and customer satisfaction by rapid delivery of working software product.

=> In Agile the task are divide to time boxes 9 (small time frame) to deliver specific feature required by the Customer.

Pros:

1) It is a very realistic approach to software development.
2) Suitable for fixed or changing requirements.

Cons:

1) Not suitable for handling complex dependencies.
2) More risk of sustainability, maintainability and extensibility.

Question 4. What is SRS?

=> A software Requirements specification is a complete description of the behaviour of the system to be developed.

=> This Standard describes possible structure, desirable contents, and qualities of a software requirements specification.

Question 5. What is oops?

=> Full form of OOP is – OBJECT ORIENTED PROGRAMMING.

=> Identifying objects and assigning responsibilities to these objects.

=> In other words (OOP) is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behaviour.

Question 6. Write Basic Concepts of oops.

=> An object is like a black box.

=> There are some basic concepts that act as the building blocks of OOPs.

- Classes & Objects
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

Question 7. What is object?

=> An object represents an individual, identifiable item, unit, or entity, either real or abstract, with a well-defined role in the problem domain.

=> An Object can be defined as an entity that has a state and behaviour, or in other words, anything that exists physically in the world is called an object.

It can represent a dog, a person, a table, etc.

=> There Is two part of an object-

Object = Data + Method

Question 8. What is class?

=> Class can be defined as a blueprint of the object. It is basically a collection of objects which act as building blocks.

=> A class represents an abstraction of the object and abstracts the properties and behaviour of that object.

Question 9. What is encapsulation?

=> 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.

=> In other words the wrapping up of data and functions together in a single unit is known as encapsulation.

E.g. Capsule

Question 10. What is inheritance?

=> Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship.

=> Inheritance 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.

For example consider a Vehicle parent class and its child class Car.

Question 11. What is polymorphism?

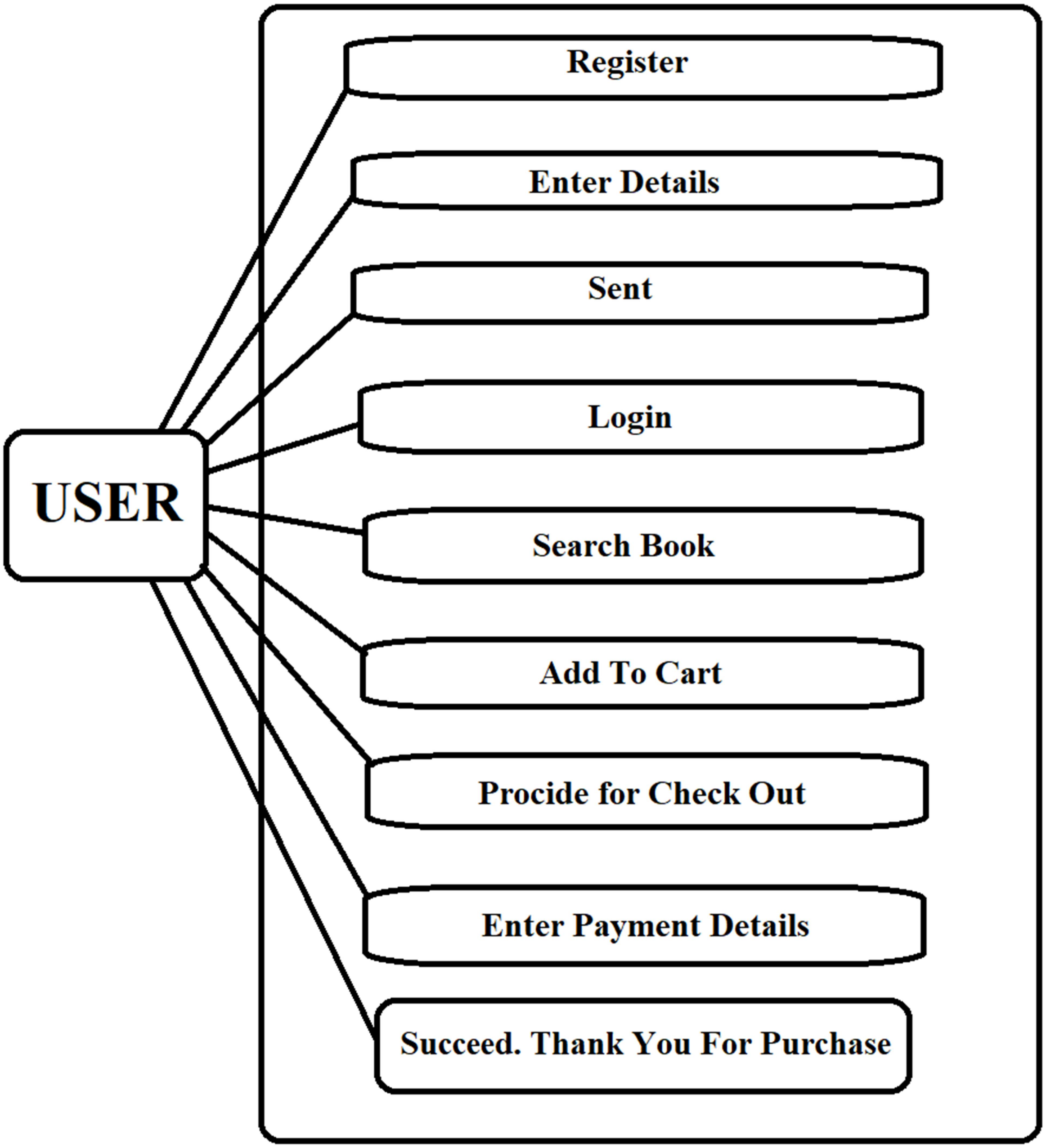
=> Polymorphism means having many forms.

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

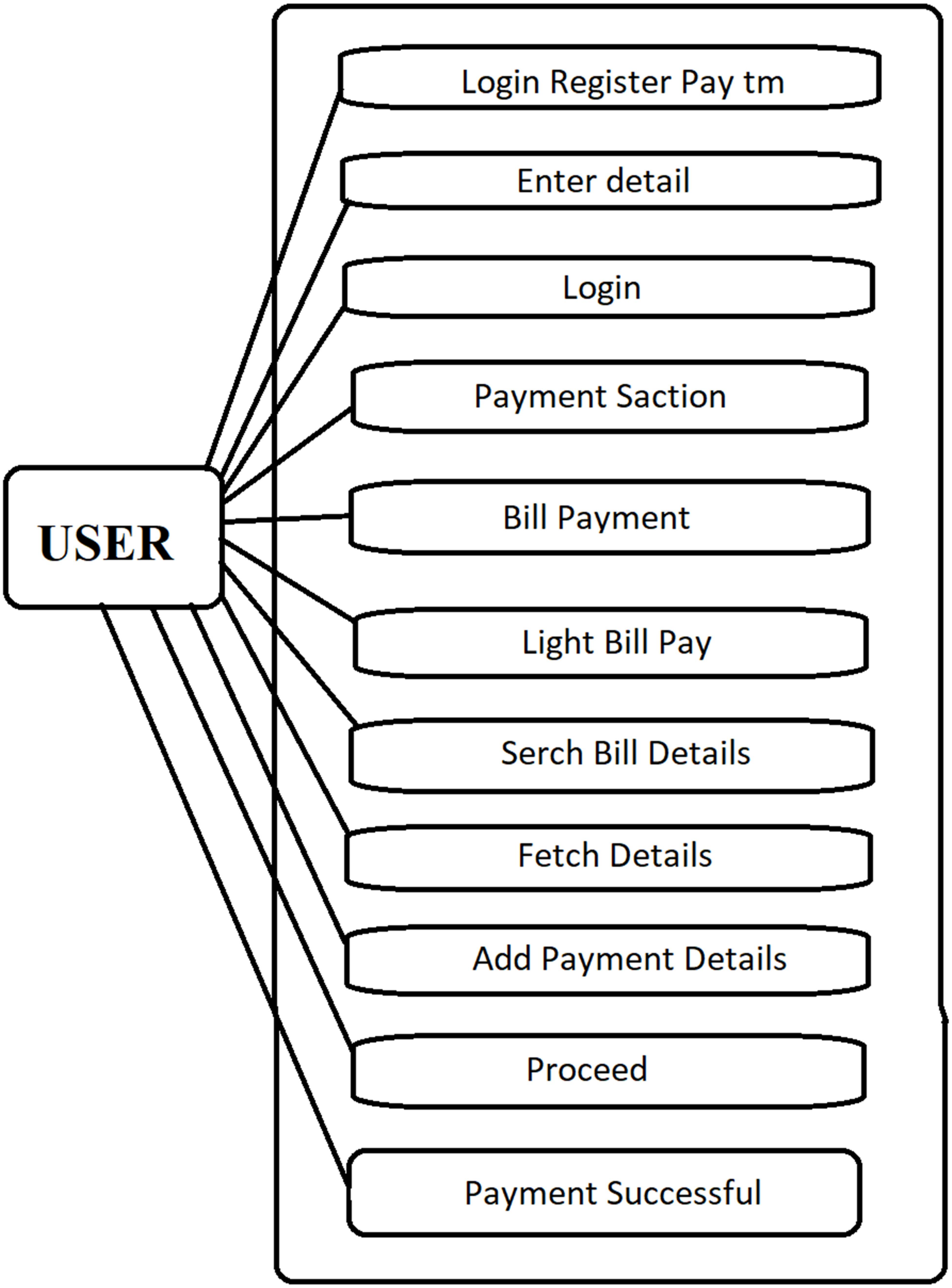
=> There is two types of polymorphism in Java

1. Compile time polymorphism (Overloading)
2. Runtime polymorphism(Overriding)

Question 12. Draw Usecase on Online book shopping.



Question 13. Draw Usecase on online bill payment system (paytm).



Question 14. Write SDLC phases with basic introduction?

=> 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. There are a number of different development models.

=> SDLC phases are hear under-

1. Requirement Collection/ Gathering
2. Analysis
3. Design
4. Implementation
5. Testing
6. Maintenance

1) Requirement Collection -

=>Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.

2) Analysis -

=> The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.

3) Design -

=> Design Architecture Document, Implementation Plan, Critical Priority Analysis Performance Analysis.

4) Implementation -

=> Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested, though there is still room for innovation and flexibility.

5) Testing -

=> The testing phase is a separate phase which is performed by a different team after the implementation is completed.
=> There is merit in this approach; it is hard to see one's own mistakes, and a fresh eye can discover obvious errors much faster than the person who has read and re-read the material many times.

6) Maintenance -

=> Software maintenance is one of the activities in software engineering, and is the process of enhancing and optimizing deployed software (software release), as well as fixing defects.

Question 15. Explain Phases of the waterfall model.

=> Phases of waterfall model is hear under -

1. **Requirement Gathering and analysis -**

   All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

2. **System Design -**

   The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.

3. **Implementation -**

   With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.

4. **Integration and Testing -**

   All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.

5. **Deployment of system -**

   Once the functional and non-functional testing is done; the product is deployed in the customer environment or released into the market.

6. **Maintenance -**

   There are some issues which come up in the client environment. To fix those issues, patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

Question 16. Write phases of spiral model.

=> Phases of spiral model are hear under

1) Planning
2) Risk Analysis
3) Engineering
4) Customer evolution

1) Planning -

=> This phase starts with gathering the business requirements in the baseline spiral. In the subsequent spirals as the product matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

=> Base on requirement determination of objective, alternatives and constrain done on planning phase.

2) Risk Analysis -

=> Risk Analysis includes identifying, estimating and monitoring the technical feasibility and management risks, such as schedule slippage and cost overrun.

=> In Risk analysis determine that is something that will delay project or increase its cost. That means go or not to go decision can be taken.

3) Engineering -

=> The Engineering phase refers to production of the actual software product at every spiral. In the baseline spiral, when the product is just thought of and the design is being developed a POC (Proof of Concept) is developed in this phase to get customer feedback.

=> Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to the customer for feedback.

=> Product tasting Also done in this phase.

4) Customer evaluation -

=> After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

Question 17. Write agile manifesto principles.

=> There are four manifestos in agile method –

1) Individuals And Interaction
2) Working Software
3) Customer collaboration
4) Responding to change

1) Individuals and interactions -

=>In Agile development, self-organization and motivation are important, as are interactions like co-location and pair programming.

2) Working software -

=> Demo working software is considered the best means of communication with the customers to understand their requirements, instead of just depending on documentation.

3) Customer collaboration -

=> As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction is very important to get proper product requirements.

4) Responding to change -

=> Agile Development is focused on quick responses to change and continuous development.

Question 18. Explain working methodology of agile model and also write pros and cons.

=> Working methodology of Agile Model -

=> 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.

=> Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

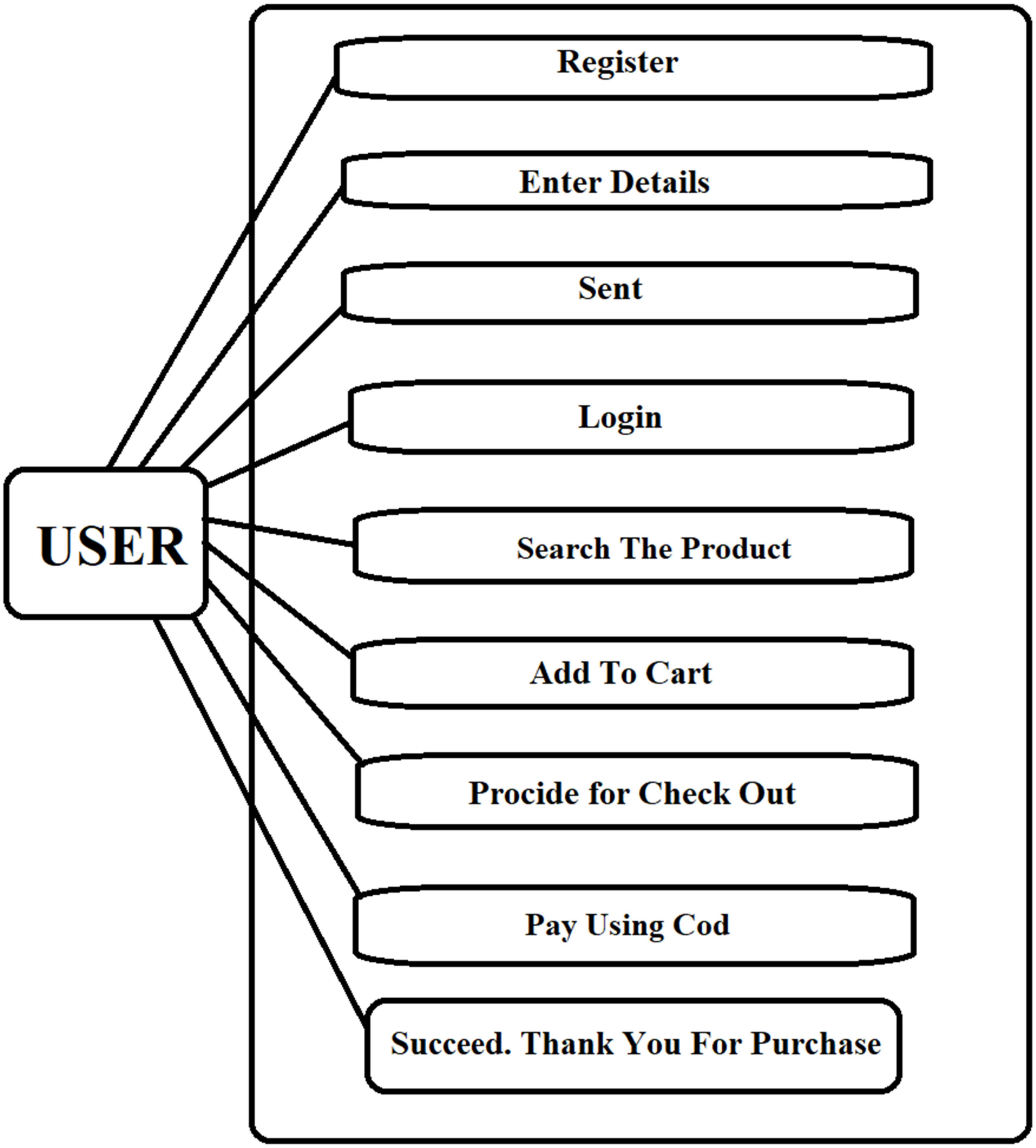
=> **PROS:-**

1. Is a very realistic approach to software development.
2. Functionality can be developed rapidly and demonstrated.
3. Resource requirements are minimum.
4. Suitable for fixed or changing requirements
5. Little or no planning required.

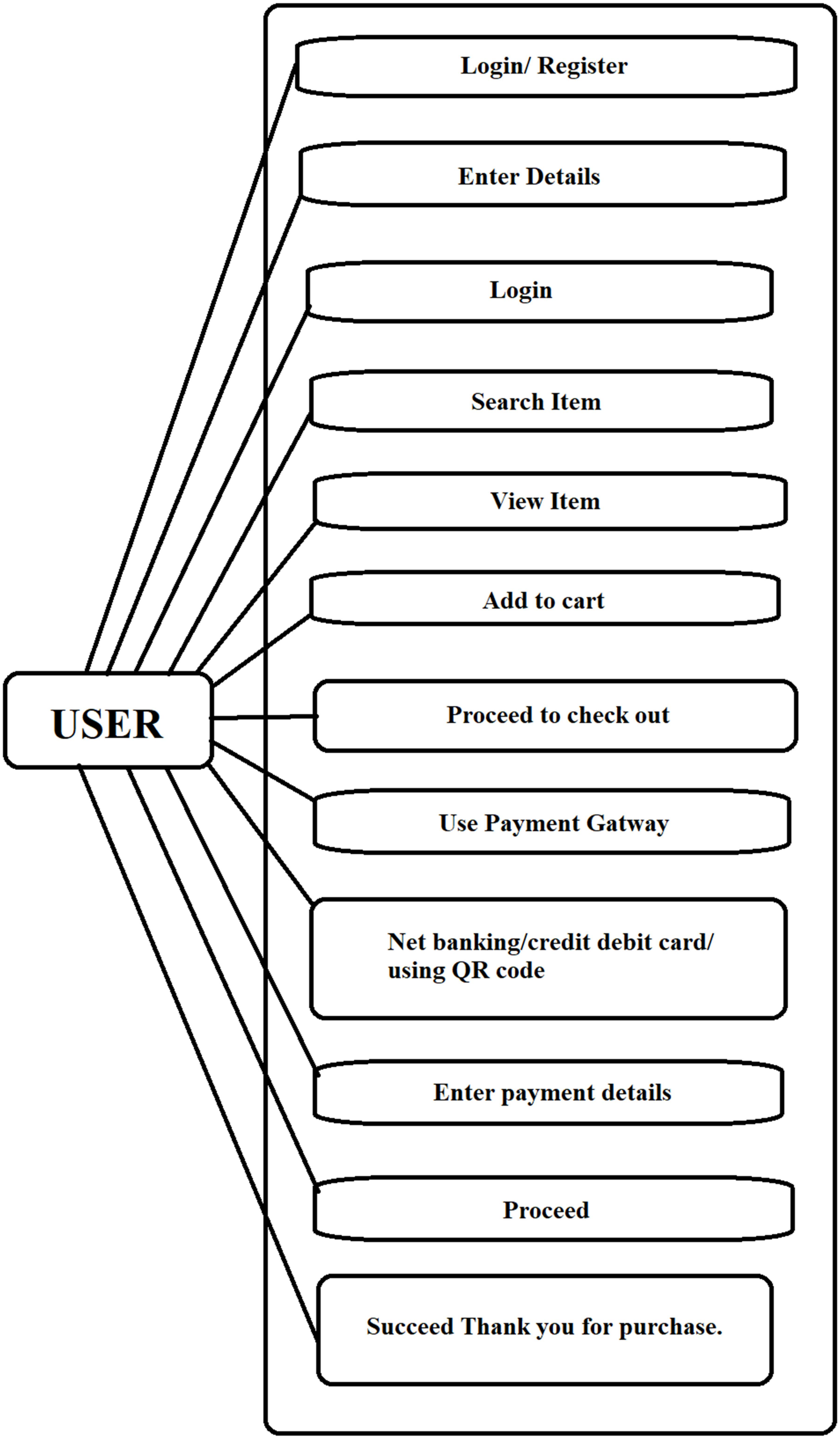
=> **CONS:-**

1. Not suitable for handling complex dependencies.
2. More risk of sustainability, maintainability and extensibility.
3. Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.

Question 19. Draw usecase on Online shopping product using COD.



Question 20. Draw usecase on Online shopping product using payment gateway.


USER
Login/ Register
Enter Details
Login
Search Item
View Item
Add to cart
Proceed to check out
Use Payment Gatway
Net banking/credit debit card/ using QR code
Enter payment details
Proceed
Succeed Thank you for purchase.