

**Question 1** :- What is SDLC ?

**Answer** :- sdlc is a software development life cycle

:- 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

1 :- Requirements collection/gathering

2 :- Analysis

3 :- Design

4 :- Implementation

5 :- Testing

6 :- Maintenance

**Question 2** :- What is software testing ?

**Answer** :- Software testing is executing a system in order to identify any gaps, errors or missing requirements in contrary to the actual desire or requirements

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

:- Software testing is the process of validating and verifying that a software program or application or product.

**Question 3** :- What is agile methodology ?

**Answer** :- Agile 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.

:- In agile model the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

**Question 4** :- What is SRS ?

**Answer** :- SRS is a Software Requirement Specification.

:- SRS is a complete description of the behavior of the system to be developed.

:- SRS is contains non-functional (or supplementary) requirements.

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

**Question 5** :- What is oop ?

**Answer** :- oop is a object oriented programming.

:- Identifying objects and assigning responsibilities to these objects.

:- objects communicate to other objects by sending messages.

:- Messages are received by the methods of an object.

:- e.g = An object is like a black box ( The internal details are hidden )

:- object is derived from abstract data type.

:- object-oriented programming has a web of interacting objects, each house-keeping its own state.

:- objects of a program interact by sending messages to each other.

**Question 6** :- Write Basic Concepts of oops

**Answer** :- 1) Object

:- 2) Class

:- 3) Encapsulation

:- 4) Inheritance

:- 5) Polymorphism

A) Overriding

B) Overloading

:- 6) Abstraction

**Question 7            :- What is object ?**

**Answer        :- e.g =**

- > Tangible Things as a car, printer.**
- > Roles            as employee, boss.**
- > Incidents        as flight, overflow.**
- > Interactions    as contract, sale.**
- > Specifications as colour, shape.**

**Question 8            :- What is class ?**

**Answer                :- Class define a blueprint for an object.**

**:- Class represents an abstraction of the object and abstracts the properties and behavior of that object.**

**:- e.g = > car will be a blueprint or design.**

**> bike will be a blueprint or design.**

**Question 9            :- What is encapsulation ?**

**Answer                :- 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.**

**:- Encapsulate in plain English means to enclose or be enclosed in or as if in a capsule.**

**:- Encapsulation in Java is the process of wrapping up of data (properties) and behavior (methods) of an object into a single unit and the unit here is a Class (or interface).**

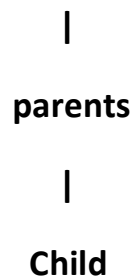
**Question 10**        :- What is Inheritance ?

**Answer**            :- Inheritance means that one class inherits the characteristics of another class.

e.g = > 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.

e.g = > grandparants



**Question 11**        :- What is polymorphism ?

**Answer**            :- Polymorphism means “having many forms”.

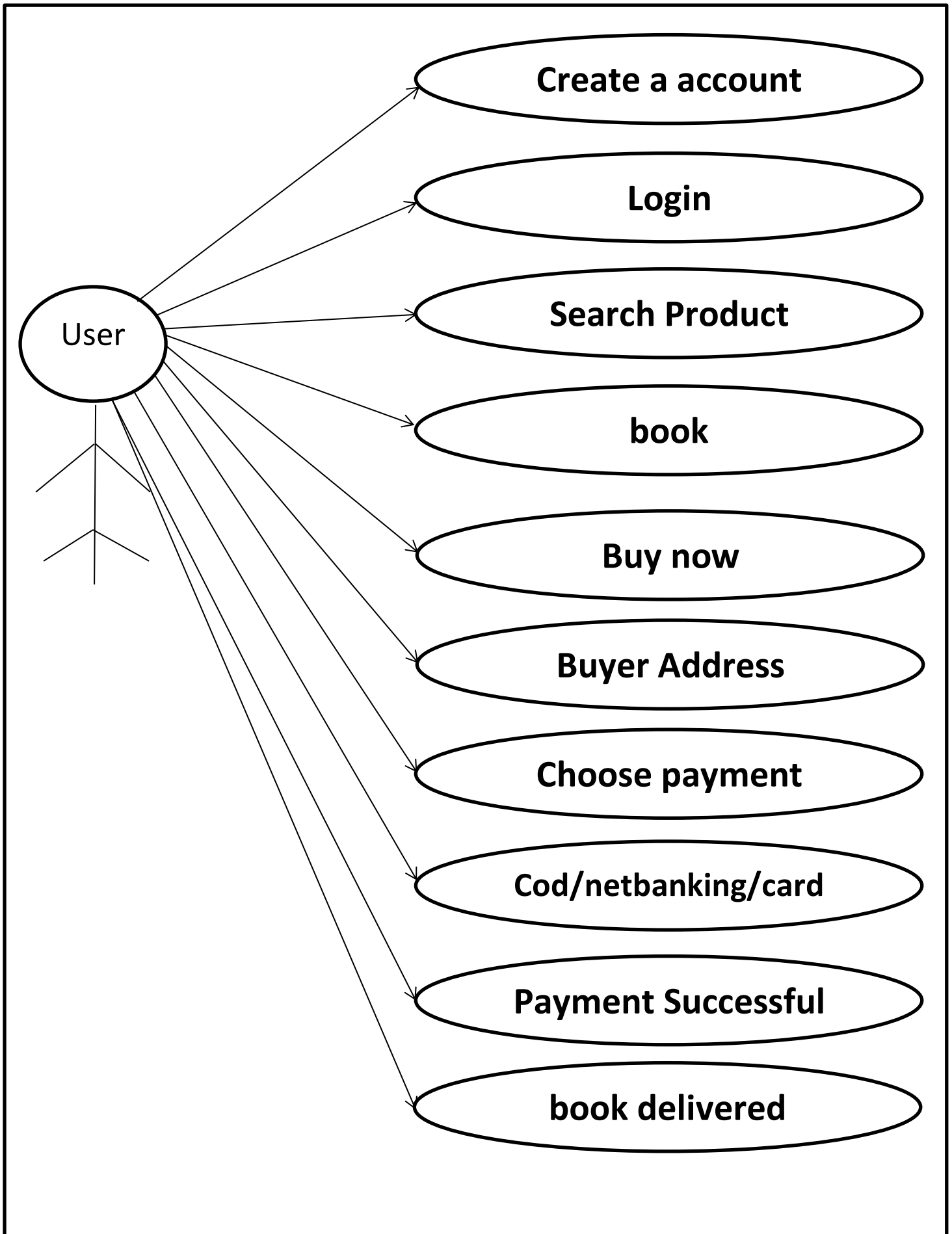
                         :- It allows different objects to respond to the same message in different ways, the response specific to the type of the object.

                         :- There is two types of polymorphism

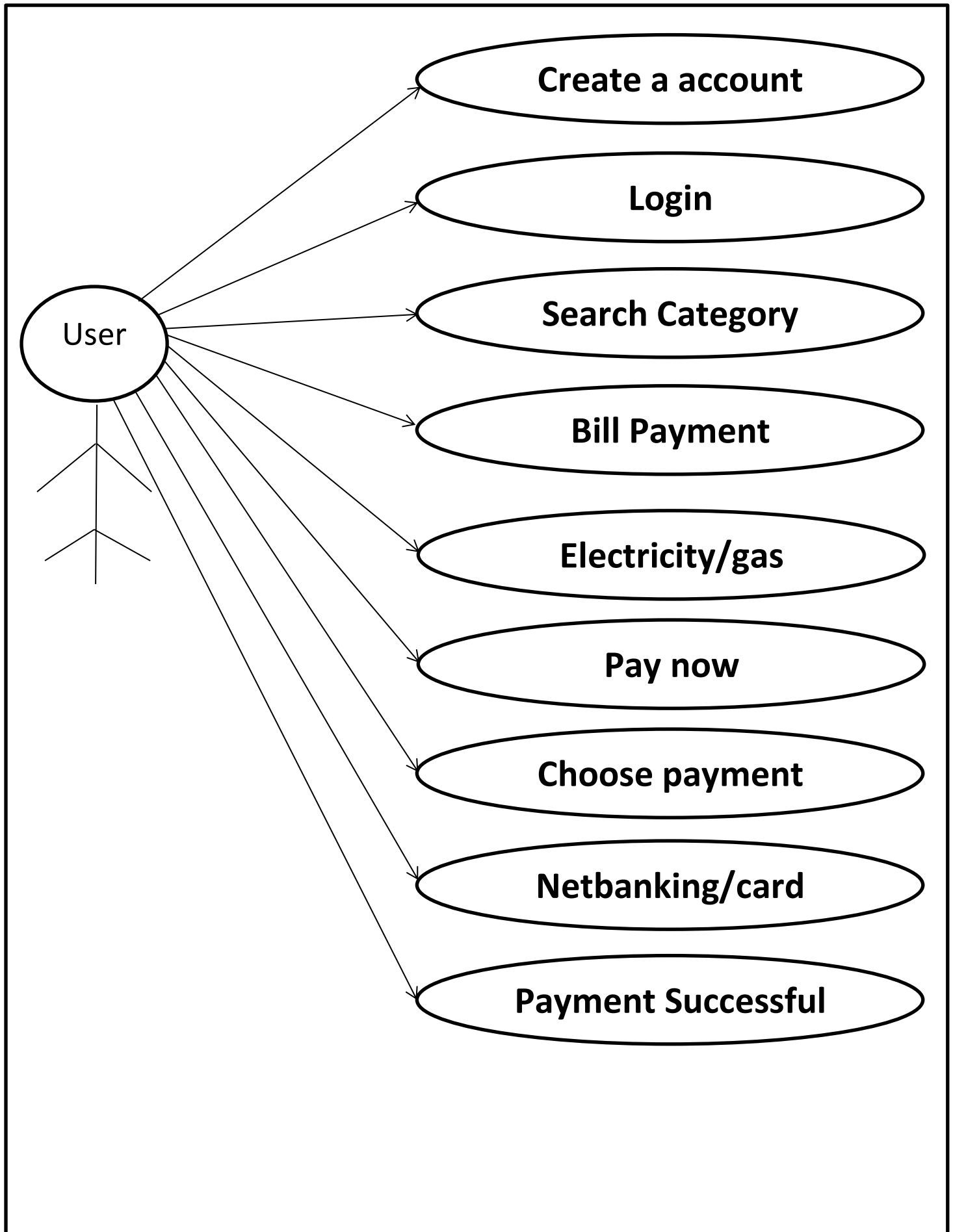
1) Overloading

2) Overriding

**Question 12 :- Draw use case on Online book Shopping.**



**Question 13 :- Draw use case on Online Bill Payment System (Paytm)**



**Question 14**                    :- Write SDLC phases with basic introduction

**Answer**                    :- 1) Requirements Collection / Gathering

                                 => Requirements definitions usually consist of natural language, supplemented by (e.g., UML) diagrams and tables.

                                 => Three types of problems can arise:

                                      1 => Lack of clarity

                                      2 => Requirements confusion

                                      3 => Requirements Amalgamation

                                 => Types of Requirements:

                                      1 => Functional Requirements

                                      2 => Non-Functional Requirements

## **2) Analysis**

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

                                 => This analysis represents the “what” phase.

                                 => This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture.

                                 => This phase represents the “how” phase

## **3) Design**

                                 => Design Architecture Document

                                 => Implementation Plan

                                 => Critical Priority Analysis

                                 => Performance Analysis

                                 => Test Plan

#### **4) Implementation**

**=> In the implementation phase, the team builds the components either from scratch or by composition.**

**=> e.g = a component may be narrowly designed for this particular system, or the component may be made more general to satisfy a reusability guideline.**

**=> The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.**

#### **5) Testing**

**=> Simply stated, quality is very important. Many companies have not learned that quality is important and deliver more claimed functionality but at a lower quality level.**

**=> It is much easier to explain to a customer why there is a missing feature than to explain to a customer why the product lacks quality.**

**=> A customer satisfied with the quality of a product will remain loyal and wait for new functionality in the next version.**

#### **6) Maintenance**

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

**=> configuration and version management.**

**=> updating all analysis, design and user documentation.**

**=> Three types of problems can arise:**

**1 => Corrective maintenance**

**2 => Adaptive maintenance**

**3 => Perfective Maintenance**



**Question 15** :- Explain Phases of the waterfall model

**Answer** :-

- 1 => Requirements Collection / Gathering
- 2 => Analysis
- 3 => Design
- 4 => Implementation
- 5 => Testing
- 6 => Maintenance

**Question 16** :- Write phases of spiral model

**Answer** :-

- 1 => Planning
- 2 => Risk Analysis
- 3 => Engineering
- 4 => Customer Evaluation

**Question 17** :- Write agile manifesto principles

**Answer** :- Four Manifesto in Agile:

- 1 => Individual and Interactions
- 2 => Working Software
- 3 => Customer Collaboration
- 4 => Responding to changes

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

**Answer**                :- 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.

                              :- Iterative approach is taken and working software build is delivered after each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

**Pros**                    :- Resource requirements are minimum.

                              :- Is a very realistic approach to software development.

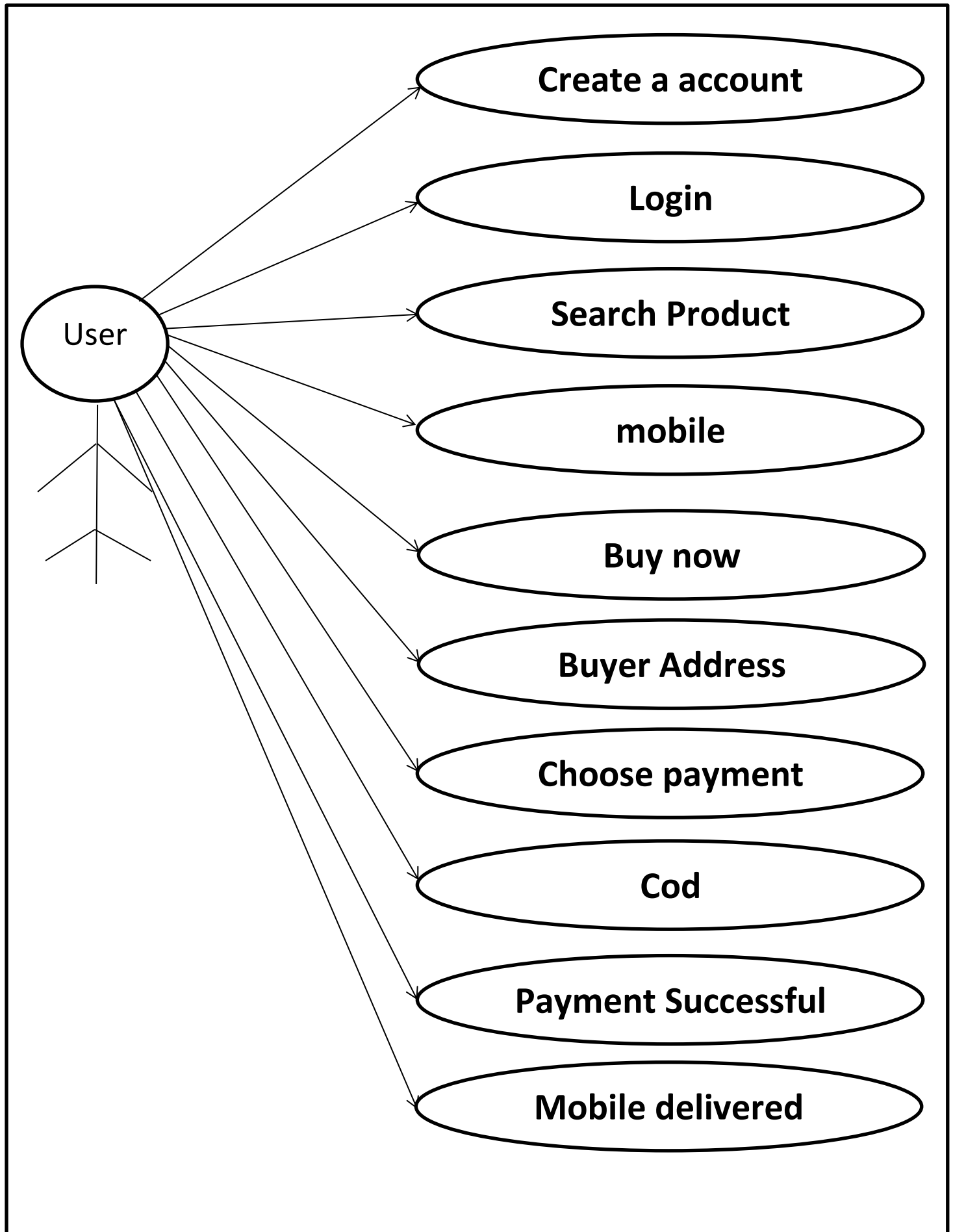
                              :- Promotes teamwork and cross training.

**Cons**                    :- Not suitable for handling complex dependencies.

                              :- More risk of sustainability, maintainability and extensibility.

                              :- 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.**

