

# Software Testing Assignment

## Module - 1 (Fundamental)

### ➤ 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.

There are a number of different development models.

A Software Development Life Cycle is essentially a series of steps, or Phases, that provide a model for the development and lifecycle Management of an application or piece of software.

#### SDLC PHASES:

- Requirement
- Analysis
- Design
- Implementation
- Testing
- Maintenance

### ➤ What is software testing?

- Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.
- Testing is the process of evaluating a system or its component(s) with the intent to find that whether it satisfies the specified requirements or not.
- Software testing is a process of executing a program or application with the intent of finding the software bugs.

It can also be stated as the **process of validating and verifying** that a Software program or application or product:

- Meets the business and technical requirements that guided it's design and Development
- Works as expected
- Can be implemented with the same characteristic

## ➤ What is agile methodology?

- 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.
- These builds are provided in iterations.
- Each iteration typically lasts from about one to three weeks.
- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- At the end of the iteration a working product is displayed to the customer and important stakeholders.

## ➤ What is SRS

- A software requirements specification (SRS) is a complete description of the behavior of the system to be developed.
- It includes a set of use cases that describe all of the interactions that the users will have with the software.
- Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional (or supplementary) requirements.
- Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).
- Recommended approaches for the specification of software requirements are described by **IEEE 830-1998**.
- This standard describes possible structures, desirable contents, and qualities of a software requirements specification.

## ❖ Types of Requirements

- Customer Requirements
- Functional Requirements
- Non-Functional Requirements

## ➤ What is oops

- Identifying objects and assigning responsibilities to these objects.
- Objects communicate to other objects by sending messages.
- Messages are received by the method of an object
- 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.

## ➤ Write Basic Concepts of oops

- Object
- Class
- Encapsulation
- Inheritance
- Polymorphism
- Overriding
- Overloading
- Abstraction

## ➤ 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" is anything to which a concept applies.
- This is the basic unit of object oriented programming (OOP).
- That is both data and function that operate on data are bundled as a unit called as object.

The two parts of an object

Object = Data + Methods

or

to say the same differently

## ➤ What is class

- A blueprint for an object
- A class represents an abstraction of the object and abstracts the properties and behavior of that object.
- This doesn't actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object.

## ➤ 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.
- **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).
- Encapsulate in plain English means to enclose or be enclosed in or as if in a capsule. In Java, a class is the capsule (or unit). In Java, everything is enclosed within a class or interface, unlike languages such as C and C++, where we can have global variables outside classes.
- Encapsulation enables **data hiding**, hiding irrelevant information from the users of a class and exposing only the relevant details required by the user.

## ➤ What is inheritance?

- Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship
- This is a very important concept of object-oriented programming since this feature helps to reduce the code size.
- 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.
- In general, Java supports single-parent, multiple-children inheritance and multilevel inheritance (Grandparent-> Parent -> Child) for classes and interfaces. Java supports multiple inheritances (multiple parents, single child) only through interfaces.
- In a class context, inheritance is referred to as implementation inheritance, and in an interface context, it is also referred to as interface inheritance.

## ➤ 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.
- The most important aspect of an object is its *behavior* (the things it can do).
- A behavior is initiated by sending a *message* to the object (usually by calling a method).
- The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called polymorphism.

## ➤ Explain Phases of the waterfall model

The classical software lifecycle models the software development as a step- by - step “waterfall” between the various development phases.

- ❖ The waterfall is unrealistic for many reasons, especially:
  - ✓ Requirements must be “frozen” to early in the life cycle
  - ✓ Requirements are validated too late
- ❖ When to use:
  - ✓ Requirements are very well documented, clear and fixed.
  - ✓ Product definition is stable.
  - ✓ Technology is understood and is not dynamic.
  - ✓ There are no ambiguous requirements.
  - ✓ Ample resources with required expertise are available to support the product.
  - ✓ The project is short.

### Phases

- Requirements Collection
- Analysis
- Design
- Implementation
- Testing
- Maintenance

### ➤ Write phases of spiral model

- Planning
- Risk Analysis
- Engineering
- Customer Evaluation

### ➤ Write agile manifesto principles

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.

Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability.

### ➤ Explain working methodology of agile model and also write pros and cons.

- ✓ 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.
- ✓ Each iteration typically lasts from about one to three weeks.
- ✓ Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.
- ✓ At the end of the iteration a working product is displayed to the customer and important stakeholders.

#### Pros

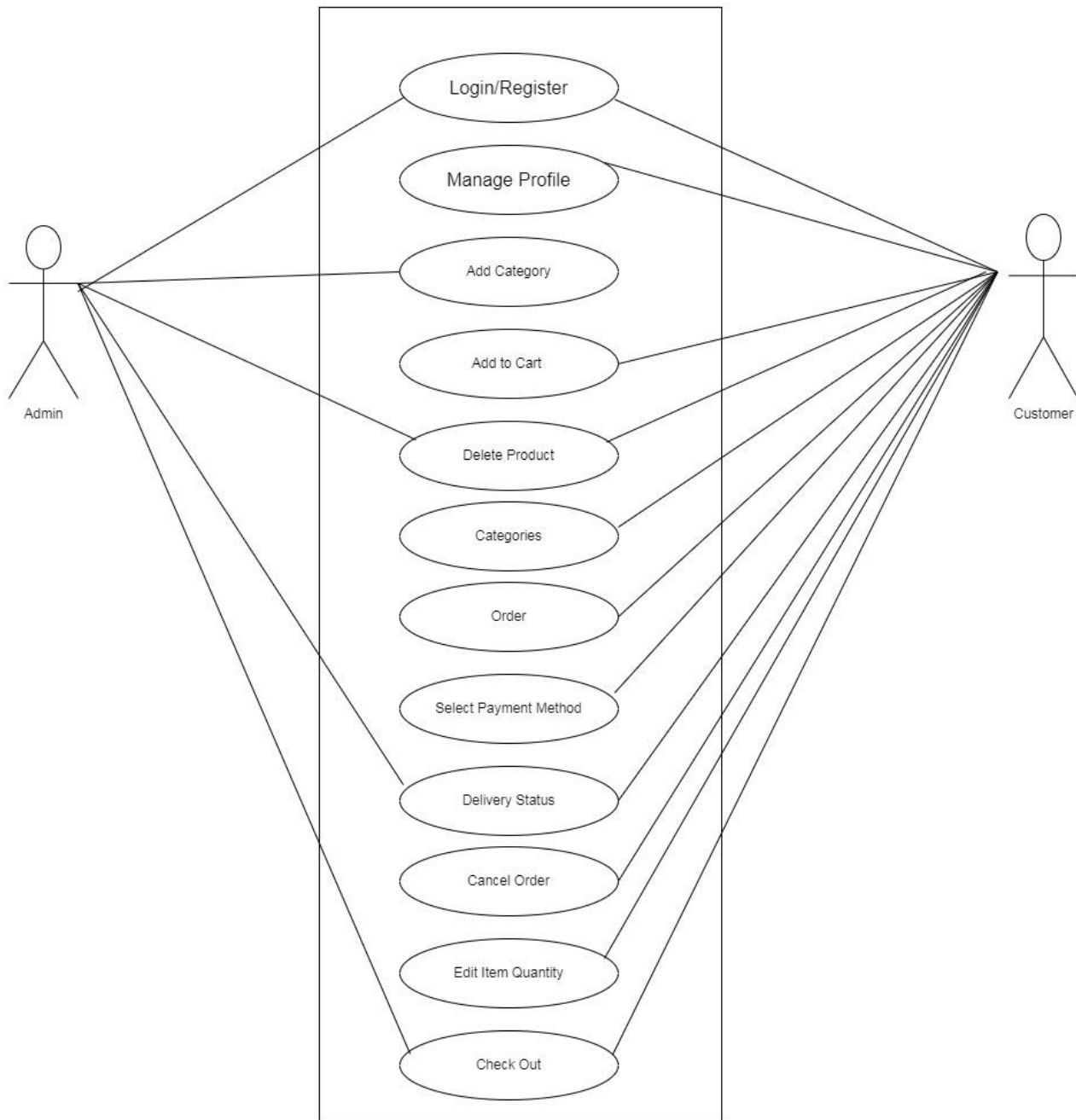
- ✓ Is a very realistic approach to software development
- ✓ Promotes teamwork and cross training.
- ✓ Functionality can be developed rapidly and demonstrated.
- ✓ Resource requirements are minimum.
- ✓ Suitable for fixed or changing requirements
- ✓ Delivers early partial working solutions.
- ✓ Good model for environments that change steadily.

- ✓ Minimal rules, documentation easily employed.
- ✓ Enables concurrent development and delivery within an overall planned context.
- ✓ Little or no planning required
- ✓ Easy to manage
- ✓ Gives flexibility to developers

### **Cons**

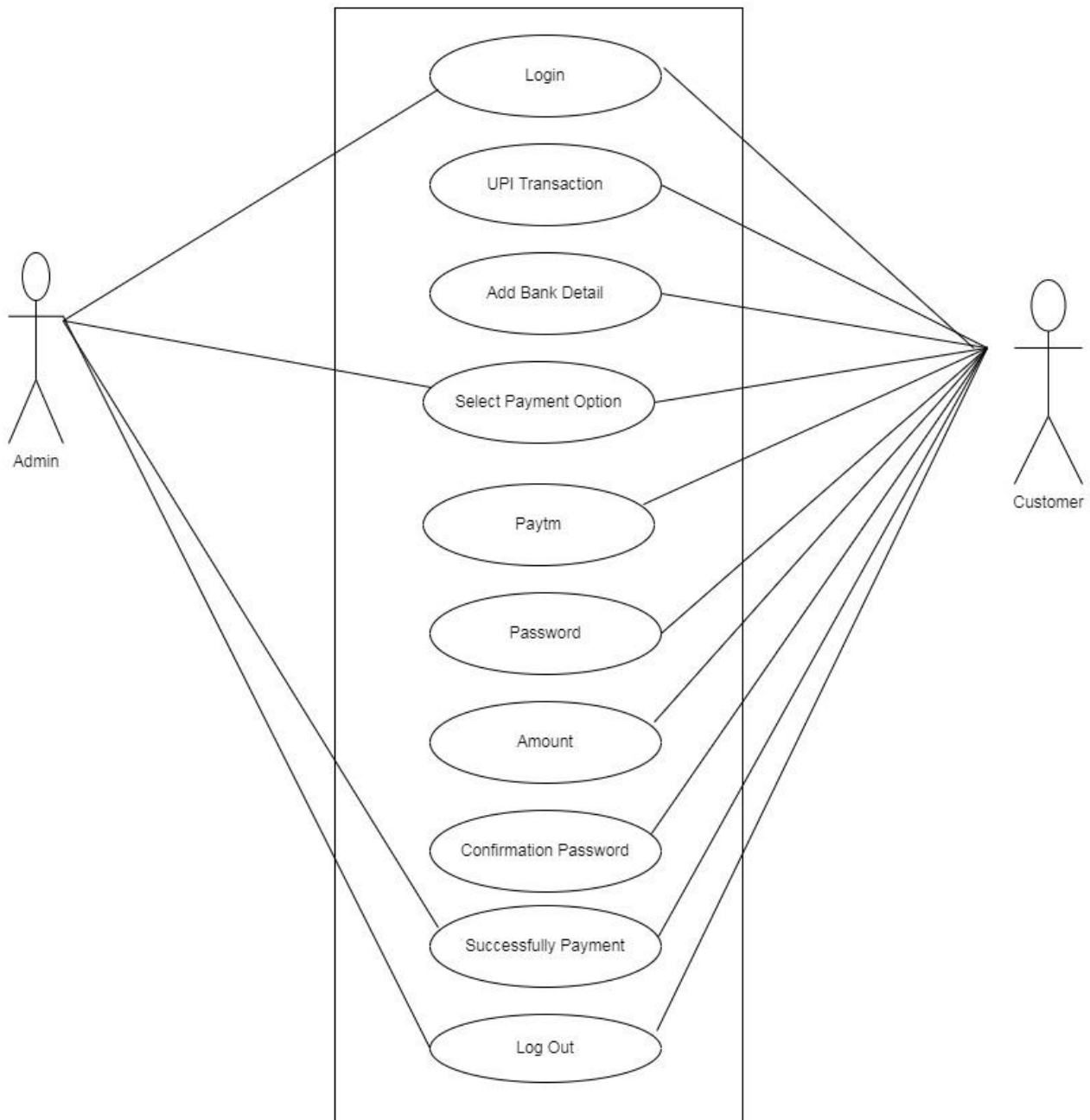
- ✓ Not suitable for handling complex dependencies.
- ✓ More risk of sustainability, maintainability and extensibility.
- ✓ An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- ✓ Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
- ✓ Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
- ✓ There is very high individual dependency, since there is minimum documentation generated.
- ✓ Transfer of technology to new team members may be quite challenging due to lack of documentation.

➤ Draw Use case on online book shopping :

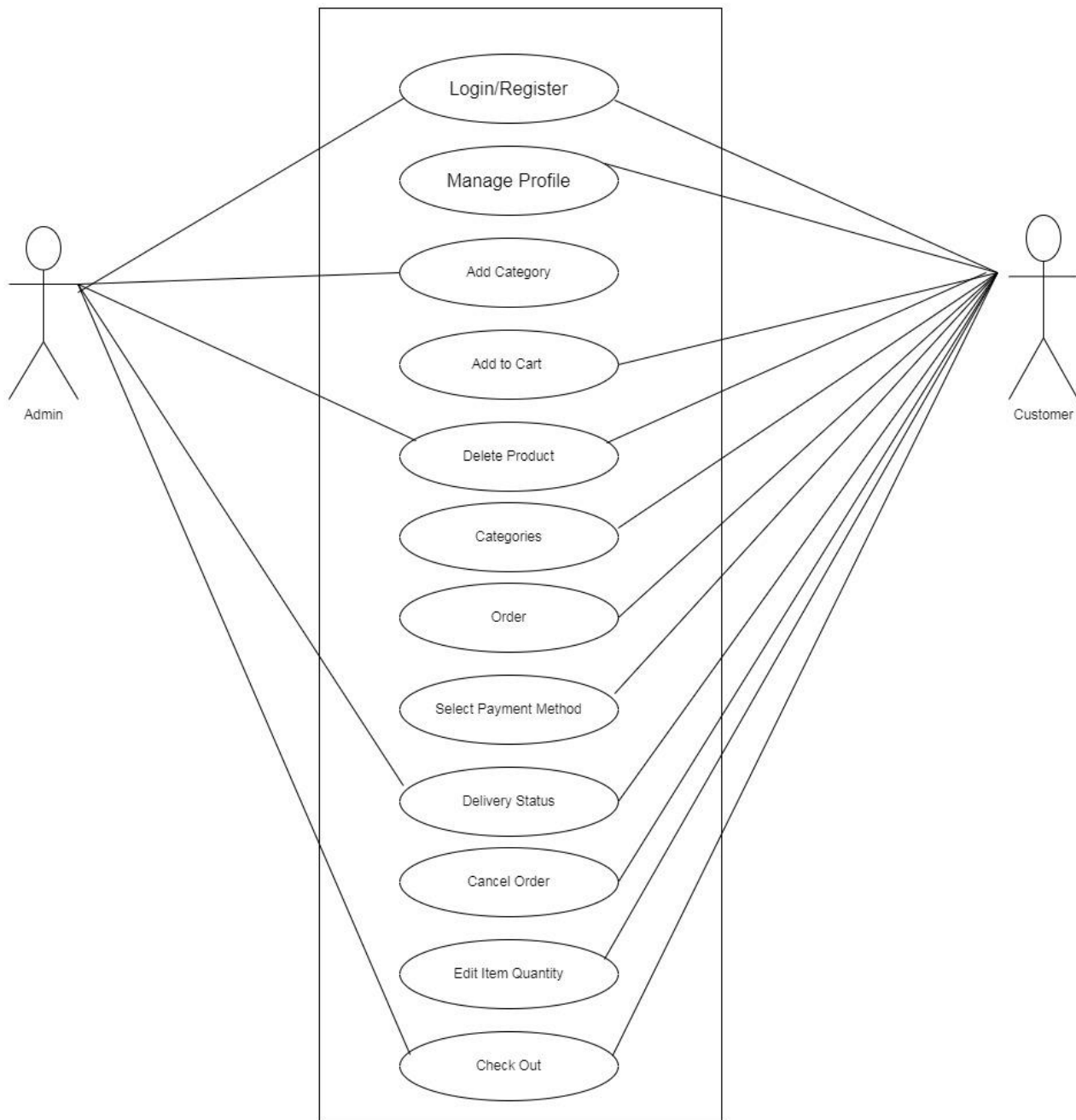




➤ Draw Use case on online bill payment system (paytm) :



➤ Draw use case on Online shopping product using COD.



➤ Draw use case on Online shopping product using payment gateway.

