# Software Testing Assignment
## Module–1(Fundamental)

## What is SDLC?

SDLC stands for Software development life cycle.
It develop of a software product that defines the process for Planning, Implementation, Testing, Documentation, Deployment, and On going maintenance and Support**.**

## What is software testing?

Software Testing is process used to identify the correctness, completeness, and quality of developed computer software.
Testing is the process of evaluation a system or its components with intent to find that whether it satisfies the specified requirements or not.
Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual desire or requirements.

## What is agile methodology?

Agile SDLC model is a combination of iteration and incremental process model with focus on process adaptability and customers 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 last from about one to three weeks.
- Every iteration involves cross functional team working simultaneously on various areas like planning, requirement 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 behaviour 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 non functional 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.
- This standard describes possible structures, desirable contents, and qualities of a software requirements specification.
- Types of Requirements

There are three types of Software requirements specification.

1. Customer Requirements
2. Functional Requirements
3. Non-Functional Requirements

## What is oops?

Oops contains Object Oriented Programming Language.

- 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.

## Write Basic Concepts of oops

1. Object
2. Class
3. Encapsulation
4. Inheritance
5. Polymorphism
   Overloading(Compile time)
   Overriding(Run time)
6. Abstraction

## What is object?

Object is instance of class.
An object represents an individual, identifiable item, unit , or entity, either real or abstract, with a well- defined role in the problem domain.
This is the basic unit of object oriented programming.
That is both data and function that operate on data are bundled as a unit called as object.

## What is class?

Class define a blueprint for an object.
Class is structure which contains member of function, variables, objects etc
Class represents an abstraction of the object and abstracts the properties and behaviour of that object.

## What is encapsulation?

Encapsulation is to wrapping on data into single unit.
Encapsulation is placing the data and the functions that work on that data in the same place. While working with procedural languages it is not always clear which functions work on which variables but object oriented programming provides you framework to place the data and the relevant functions together in the same object.

## What is inheritance?

Inheritance is who access property of one class to another class.
Inheritance means that one class inheritance the parameters of another class these is also called a relationship.
This is a very important concept of object oriented programming since these feature helps to reduce the code size.

## What is polymorphism?

Polymorphism means "having many forms".
Polymorphism is same method name but having different functionalities.
It allows different objects to respond to the same message in different ways, the response specific to the type od the object.

## Write SDLC phases with basic introduction

There are six phases in SDLC .

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

**1. Requirement Gathering**

- Although requirements may be documented in written form, they may be incomplete, unambiguous, or even incorrect.
- User and business needs change during the project
- Validation is needed throughout the software lifecycle, not only when the "final system" is delivered.

Three types of problems can arise:

- Lack of clarity: It is hard to write documents that are both precise and easy-to-read.
- Requirements confusion: Functional and Non-functional requirements tend to be intertwined.
- Requirements Amalgamation: Several different requirements may be expressed together.

**2. Analysis**

- This analysis represents the "what" phase.
- The analysis phase defines the requirements of the system, independent of how these requirements will be accomplished.
- This phase defines the problem that the customer is trying to solve.
- The deliverable result at the end of this phase is a requirement document.
- The requirement documentaries to capture the requirements from the customer's perspective by defining goals.
- This phase represents the "how" phase.
- This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture.

**3. Design**

- The Design team can now expand upon the information established in the requirement document.
- The requirement document must guide this decision process.
- Analyzing the trade-offs of necessary complexity allows for many things to remain simple which, in turn, will eventually lead to a higher quality product.
- The architecture team also converts the typical scenarios into a test plan**.**

**4. Implementation**

- In the implementation phase, the team builds the components either from scratch or by composition.
- 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.
- The implementation phase deals with issues of quality, performance, baselines, libraries,

and debugging.

- The end deliverable is the product itself. There are already many established techniques associated with implementation.

5. **Testing**

- The testing phase is a separate phase which is performed by a different team after the implementation is completed.
- Unfortunately, delegating (alternate) testing to another team leads to as lack (dull) attitude regarding quality by the implementation team.
- If the teams are to be known as craftsmen, then the teams should be responsible for establishing high quality across all phases.
- an attitude change must take place to guarantee quality. Regardless if testing is done after the-fact or continuously, testing is usually based on a regression technique split into several major focuses, namely internal, unit, application, and stress.

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.
- Software maintenance is also one of the phases in the System
- Development Life Cycle (SDLC), as it applies to software development.
- The maintenance phase is the phase which comes after deployment of the software into the field.
- The developing organization or team will have some mechanism to
- document and track defects and deficiencies.

## Write agile manifesto principles

There are four Principles of Agile manifesto.

1. **Individuals and interactions.**
   In agile development self organisation and motivation are important as are interaction like co-location and pair programming.
2. **Working Software.**
   Demo working software is consider the best means of communication with the customer to understand their requirement instead of just depending on documentation.
3. **Customer Collaboration**
   As per requirement can not be gathering completely in the beginning of the project due to various factors, continues customers interaction is very important to proper product requirements.
4. **Responding to change**
   Agile development is focused on quick responses to change and continuous development.

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

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 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.

**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.

## Explain Phases of the waterfall model

The classical software life cycle model the software development as a step by step waterfall between the various development phases.

- 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.

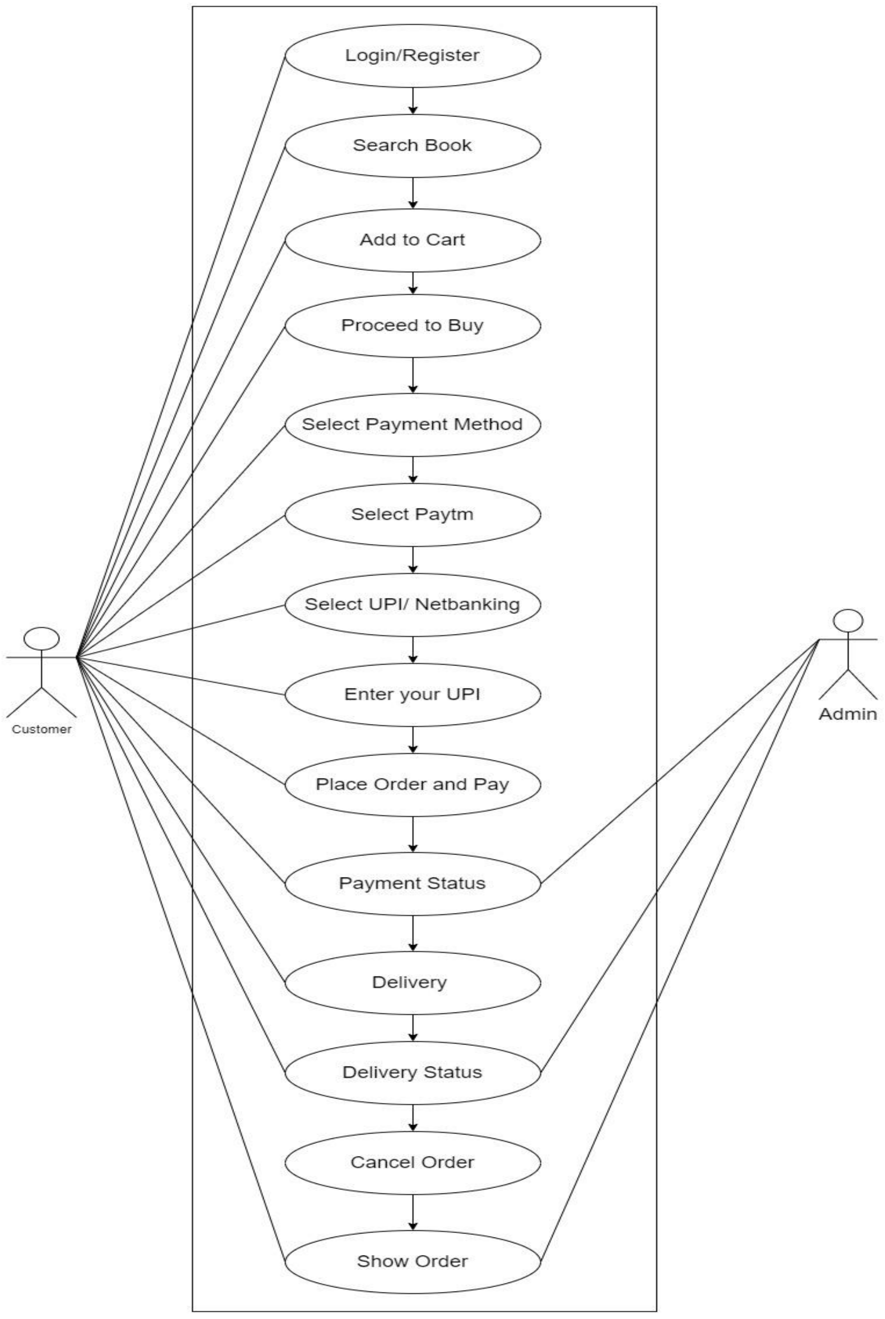
## Write phases of spiral model

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.

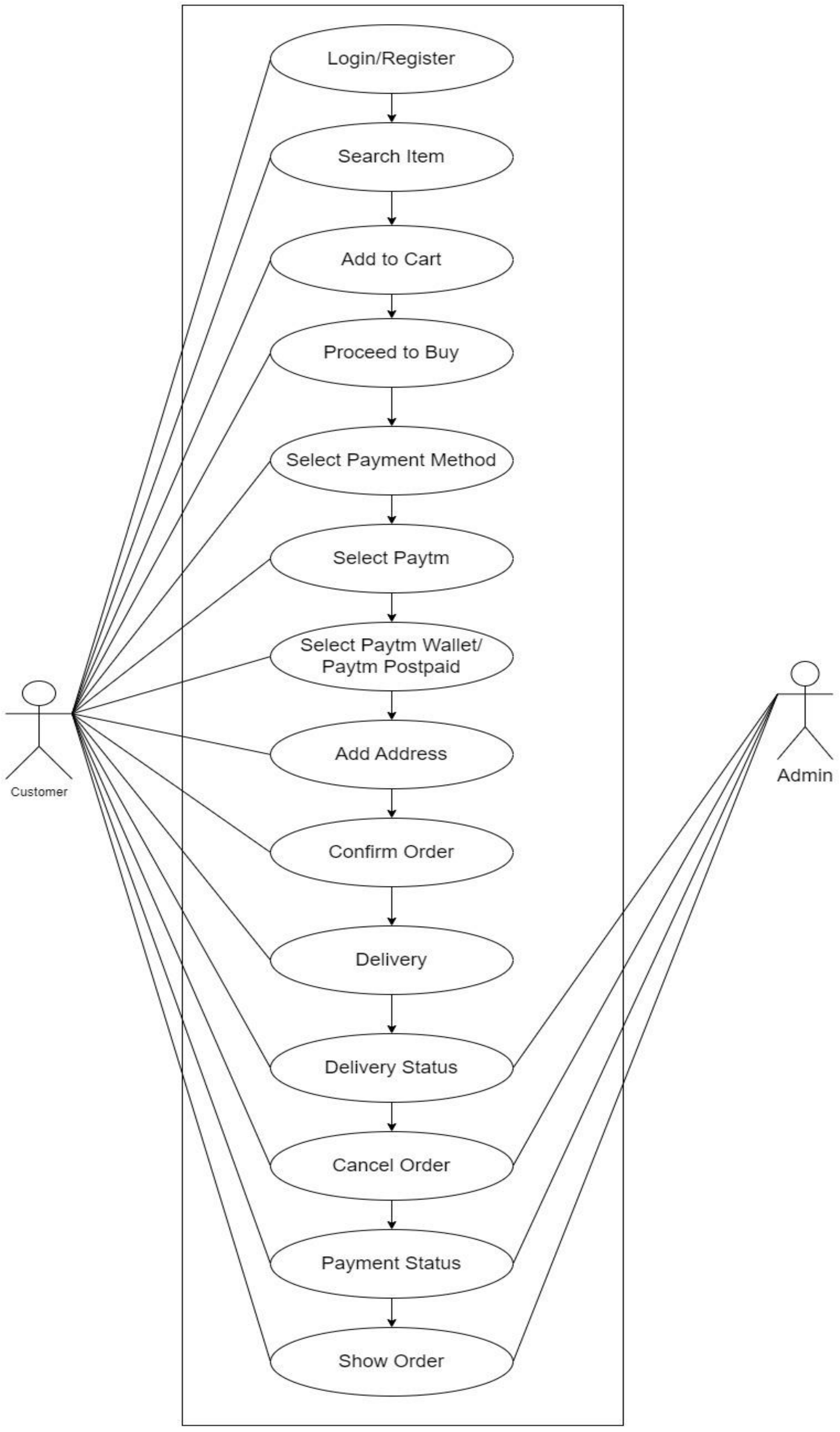
- When costs there are a budget constraint and risk evaluation is important.

- For medium to high risk project.
- Long term project commitment because of potential changes to economic priorities as the requirements change with time.
- Customer is not sure of their requirements which are usually the case.
- Requirements are complex and need evaluation to get clarity.
- New product line which should be released in phases to get enough customer feedback.
- Significant changes are expected in the product during the development cycle.

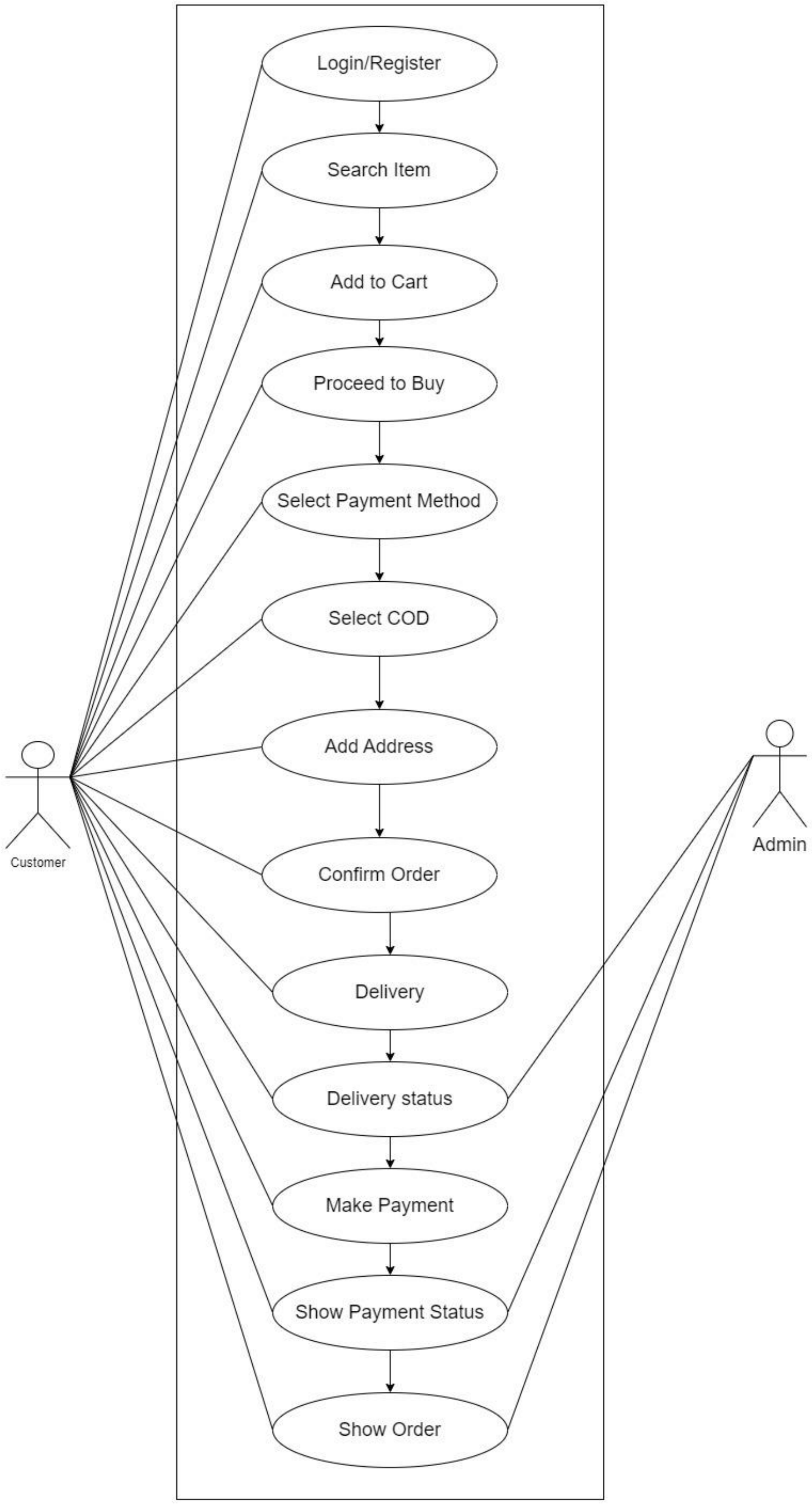
## Draw Use case on Online book shopping



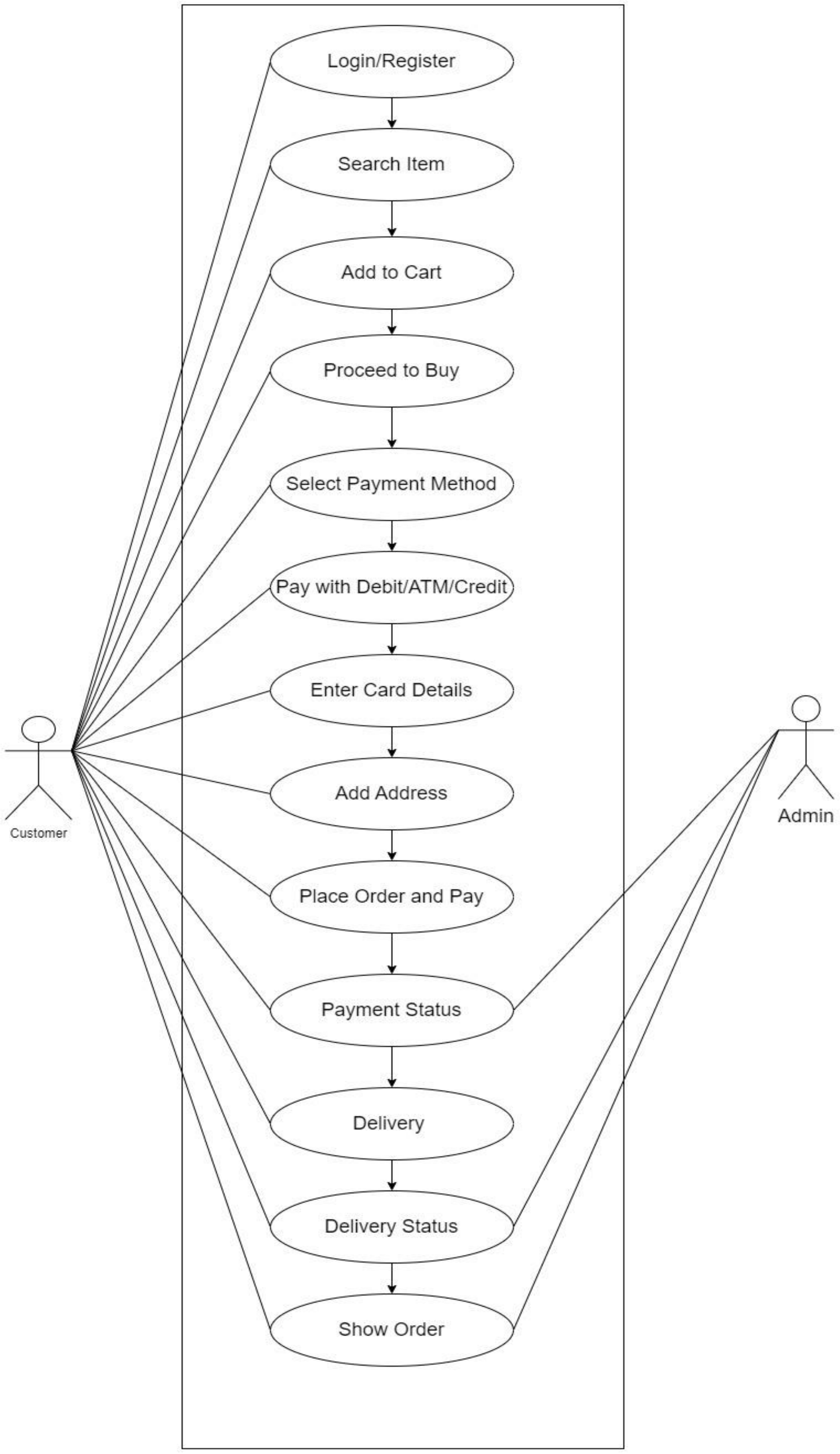
**□ Draw Use case on online bill payment system (pay tm)**



## □ Draw usecase on Online shopping product using COD.



 **Draw usecase on Online shopping product using payment gateway.**


Login/Register
Search Item
Add to Cart
Proceed to Buy
Select Payment Method
Pay with Debit/ATM/Credit
Enter Card Details
Add Address
Place Order and Pay
Payment Status
Delivery
Delivery Status
Show Order
Customer
Admin