MODULE -1 (FUNDAMENTAL)

**What is SDLC?**

* SDLC means Software Development Life Cycle. It is a structure imposed on the development of a software product that defines the process of planning, implementation, testing, documentation, deployment and maintenance and support.
* It is a series of steps, or phases that provides a model for the development and management of an application or piece of software.
* SDLC represent the process that establish life cycle for software, and provide mode for development, acquisition, and configuration of software systems.

**What is Software Testing?**

* Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software. It is executing a system in order to identify any gaps, defects, errors or any missing requirements.
* Testing is necessary at every stage when the software / application is made. It can also stated as the process of validating and verifying that a software or application is ready to go in market.
* It is to inspect whether the product meets the requirements that guided its design and development. Is it working as expected or not.
* The process consisting of all life cycle activities, both static and dynamic, concerned with planning, preparation, and evaluation of software products and related work products to determine that they satisfy specified requirements, to demonstrate that they are fit for purpose and to detect defects.
* Testing takes place throughout the Software Development Life Cycle.

**What is Agile Methodology?**

* Agile is a combination of iterative and incremental process which focus on process adaptability and customer satisfaction by rapid delivery of working model.
* Agile method heavily depends on customer side.
* In agile we are not following particular documentation or process but we work as the customer tells us.
* In agile we break the model into smaller incremental builts / cycle / iteration.
* The time span for each iteration is 1-3 weeks i.e. minimum 1 week – maximum 3 weeks.
* While working in agile we need to make sure about the deadlines that is we need to deliver each working model at a decided deadline strictly.
* In Agile, we are not creating documents or less documents.
* After the end of 1 iteration we display the working model to the customer and if there is any changes or any new requirements we will get them solve by the next iteration.

**What is SRS ?**

* A software requirements specification (SRS) is a complete description of the behaviour of the system to be developed.
* It include set of cases which helps easy to know all the interaction of users will have with software.
* Use Case are also know as functional requirements. In addition to use case, the SRS also contains Non-Functional requirements.

**What is OOPS?**

* OOP means Object Oriented Programming.
* In this process, the main thing is the Object oriented result.
* In simple words, this process is only result based process, It only focus on object rather than process.
* For an example, If a chief is cooking in kitchen and the customer ordered something, In this the object is to make a dish. So the customer is not concerned how he made that dish, But he is concerned only about to have that dish / having that object.

**Write Basic concepts of OOPS?**

* Basic Concepts of OOPS :

1. Object
2. Class
3. Encapsulation
4. Inheritance
5. Polymorphism
6. Abstraction

**What is Object ?**

* OBJECT : An Object refers to an individual thing. Object is a basic unit of OOP (Object Oriented Programming). Object is a part of a class.

**What is Class ?**

* CLASS : Class works as a blue print for an object.

For an Example, “Samsung is a Phone” In this the Object is consider as “Samsung” and the Class is consider as “Phone”

There can be types of object in a class but there cannot be types of class. In the bellow given example there can be different companies phone but The Class “Phone” is always the same.

**What is Encapsulation ?**

* Encapsulation : “Wrapping up Data into a single unit” . It enables data hiding, Hiding irrelevant information from users.

**What is Inheritance ?**

Inheritance : That means to have ability to adapt the behaviour of parent class into the child class.

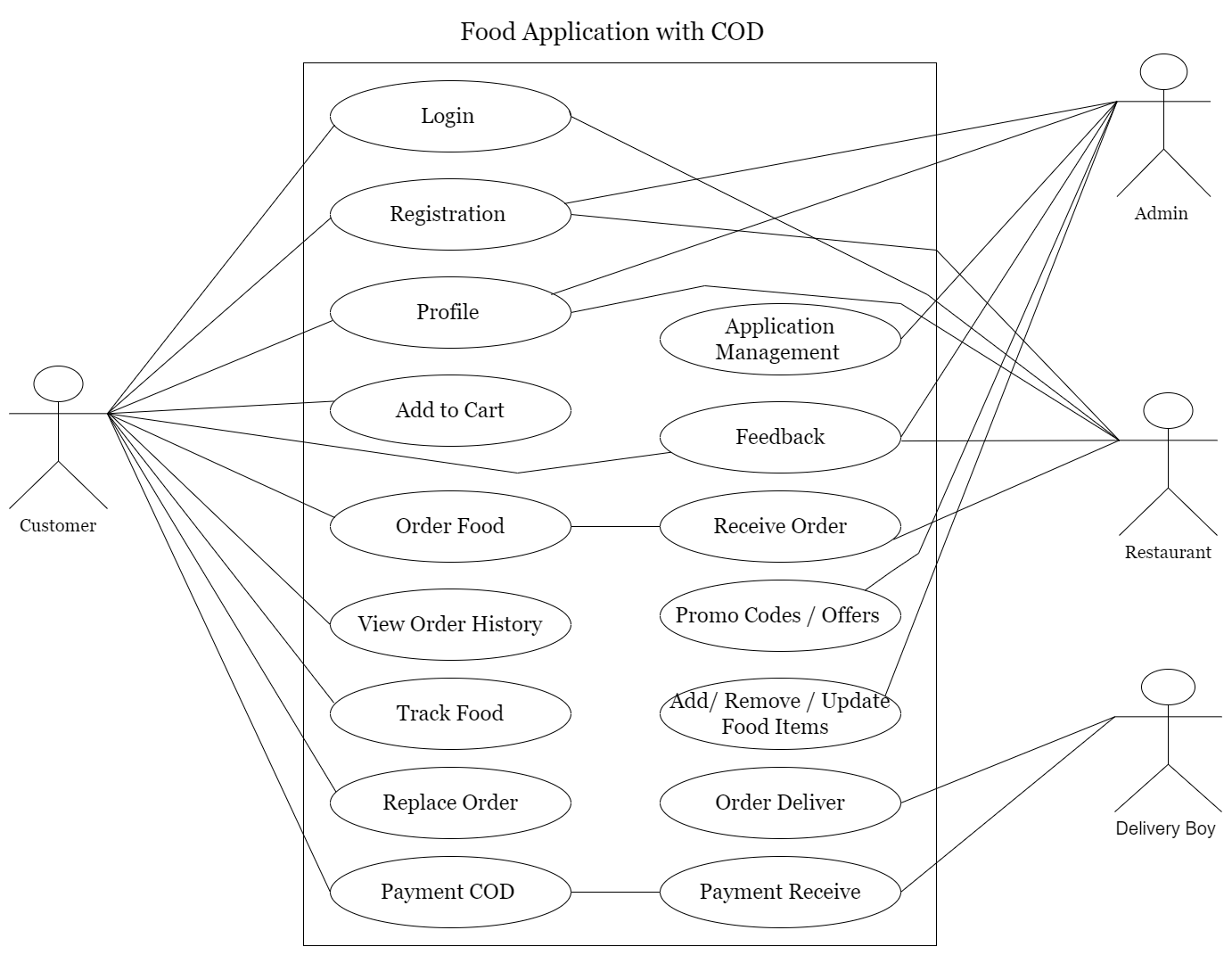
For an example, If the parent class is in business field the child class also have the ability to adapt that nature of having business in their own self.

**What is Polymorphism ?**

* Polymorphism : Ability to react in different ways. The word “Polymorphism” derived from the meaning :

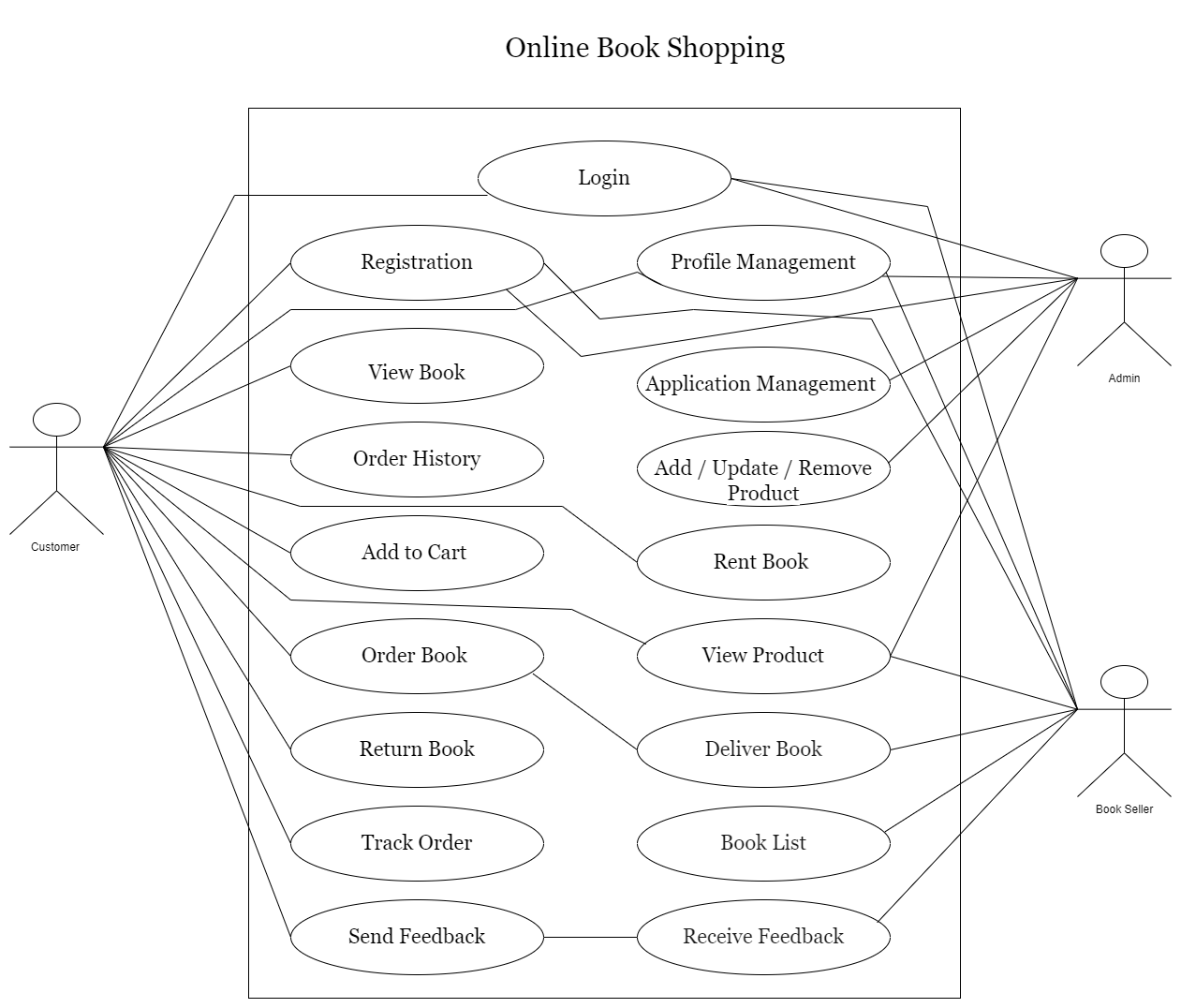
“POLY means MANY” and “MORPHS means FORMS” that means having many forms.

**Draw a Usecase on Food Application with COD.**

* USECASE:

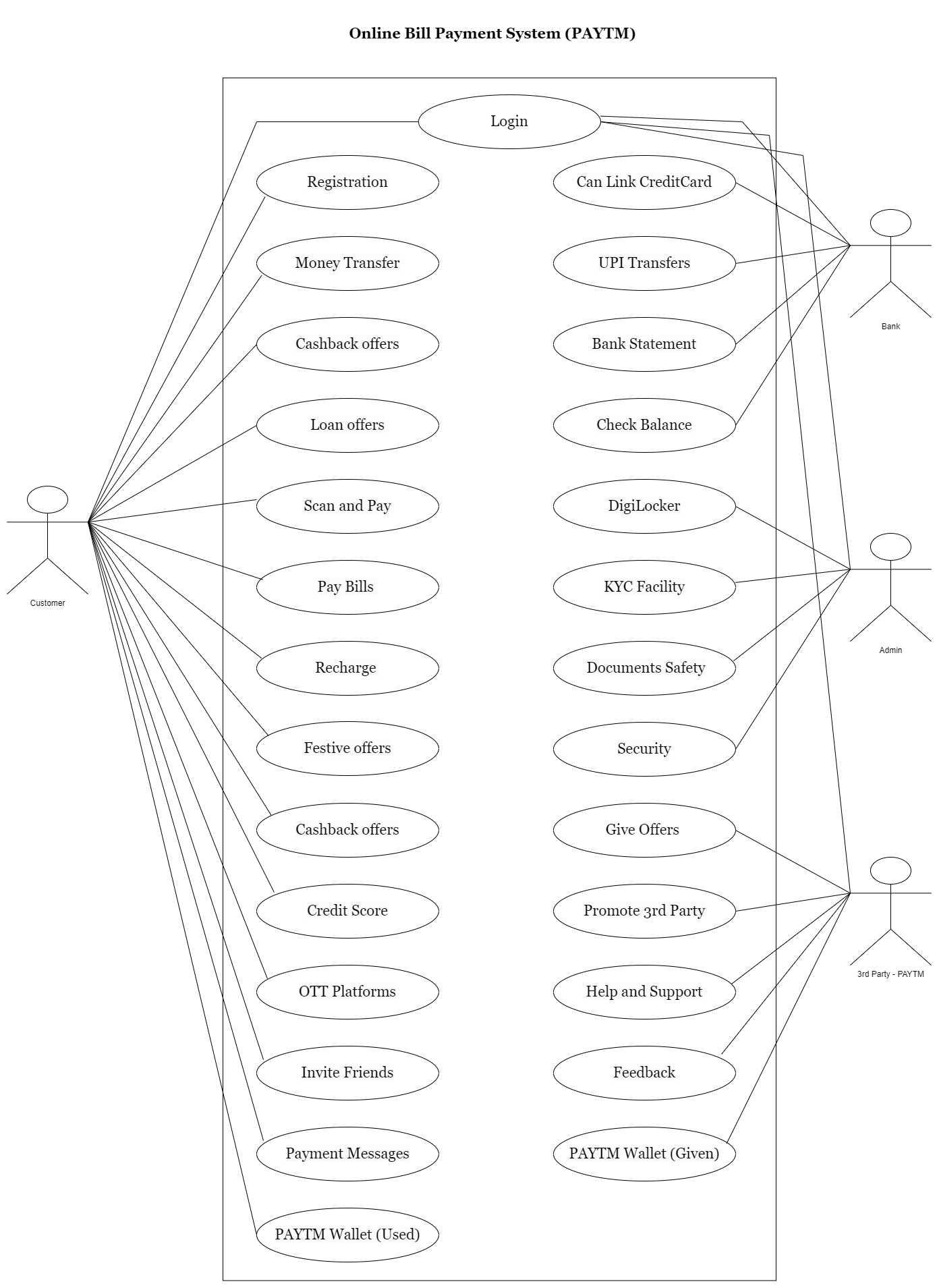
**Draw Usecase on Online Book Shopping**

* USECASE :



**Draw Usecase on Online Bill Payment System (Paytm)**

* USECASE :



**Write SDLC phases with basic information.**

* Phases of SDLC :

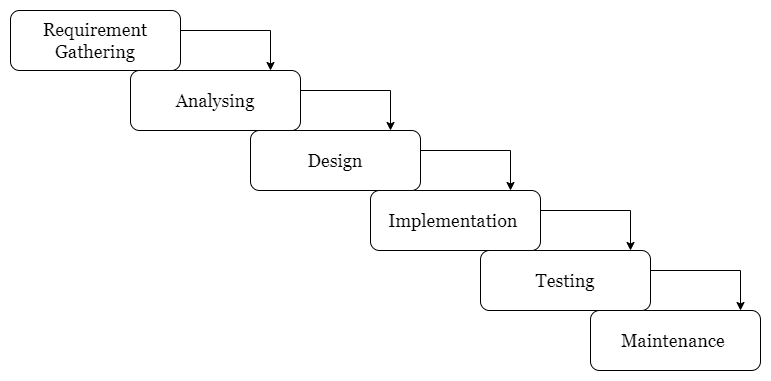
1. Requirement Gathering.
2. Analysing.
3. Design.
4. Implementation.
5. Testing.
6. Maintenance.

* Requirement Gathering: The first phase of SDLC is to gather the proper requirements of customer. It is very important to understand the need of what customer actually wants. Requirements includes which features they want. They are mainly 2 types of requirements. That is, Functional Requirement and Non-Functional Requirement. In this it will define both “WHAT” the system works and “HOW GOOD” the system works.
* Analysing: This phase starts with the requirement documents delivered by Requirement Phase. In analysing phase it defines for the requirement of the system, of how these requirements will be accomplished. This phase defines the problem that the customer wants to resolve. It is a phase which will work ahead according to the requirements of the customer. It is a roadmap of how the making of software / application will be done.
* Design: In this phase, it gives a specific designing to what to implement, a layout/design of how to analysis the performance and also about the Test Plan. This phase will make the analysis part more in a crystal clear manner. If the design is not performed perfectly it may lead to lower quality product.
* Implementation: This phase is basically to construct upon the planning and analysing in the software that means from the scratch . It is the part where CODING will take place. The implementation will start once the required documentation from designing phase and analysis phase is give. According to the Requirements and How the software will work, now it is the time to implement. This phase deals with issues of quality, performance, debugging, baselines, and libraries.
* Testing : The testing phase is a different phase which is performed by the a different team after implementation is done. In Testing phase team has to check all the high level requirements which will help to improve the quality of software/product. All the testing areas such as Unit Testing, Internal Testing, Application Testing, Stress Testing is has to be done by the tester.
* Maintenance Phase : It is one of the activity in SDLC which takes place after deployment of a software. In this the team has documents to track the defects and deficiencies. There are mainly 3 types of maintenance that takes place.

1. Corrective Maintenance
2. Adaptive Maintenance
3. Perfective Maintenance

**Explain Phases of Waterfall model.**

* Phases of Waterfall Model.



* In this Model, The requirements of customer has to taken at the firstly of the start of working.
* In waterfall model, we can’t go backward for any changes or mistake has done.
* This model is used in small projects with Experience Members in team.
* Once the working is started No other requirements or changes can be done.
* In this requirements should be well documented, clear and fixed so that the chances of failure gets low.
* Phases are processed and completed 1 at a time.
* There is high amount of risk because of we can’t make any changes after moving ahead.

**Write phases of Spiral Model**

There are 4 phases of Bohem’s Spiral Model. They are

1. Planning
2. Risk Analysis
3. Engineering
4. Customer Evaluation

* Planning : Determination of objectives, alternatives and Constrains
* Risk Analysis : Analysis of alternatives and identification / resolution of risks.
* Customer Evaluation : Assesment of the results of engineering.

This process is the process between go, no- go decision to the completion of the project. The process has 4 Phases. In this Model, first they will plan and ask about the requirements of the customer and will start working on the other phase that is known as Risk Analysis. In this phase they will analysis about the risk factor of something that will delay the project or Increase the cost in budget. After the risk analysis phase the phase of development occurs. Development of the product goes to next level. It is to making the basic model of the project. As the first model is ready, Now they will show this to customer. After showing to the customer if there is any changes in requirements it can be done. In this phase it is allowed to modify the requirements. If there is any add in requirements, the work will start from the starting from the requirement phase. This Spiral will go on until the customer is agreed with the final output of the project.

**Write Agile Manifesto Principles**

1. Individuals and Interactions
2. Working Software
3. Customer Collaborations
4. Responding to the Change

**Explain working methodologies of agile model and also write the pros and cons**

Agile Model is a combination of iterative and incremental process models with the focus on process of adaptability and customer satisfaction. In agile the tasks are divided to small time frames to deliver specific features for a release. In this the flow of working model is to first Plan, Requirement Analysis, Designing, Building and Testing. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. At the end of the iteration a working product is displayed to the customer and important stakeholders.

**PROS:**

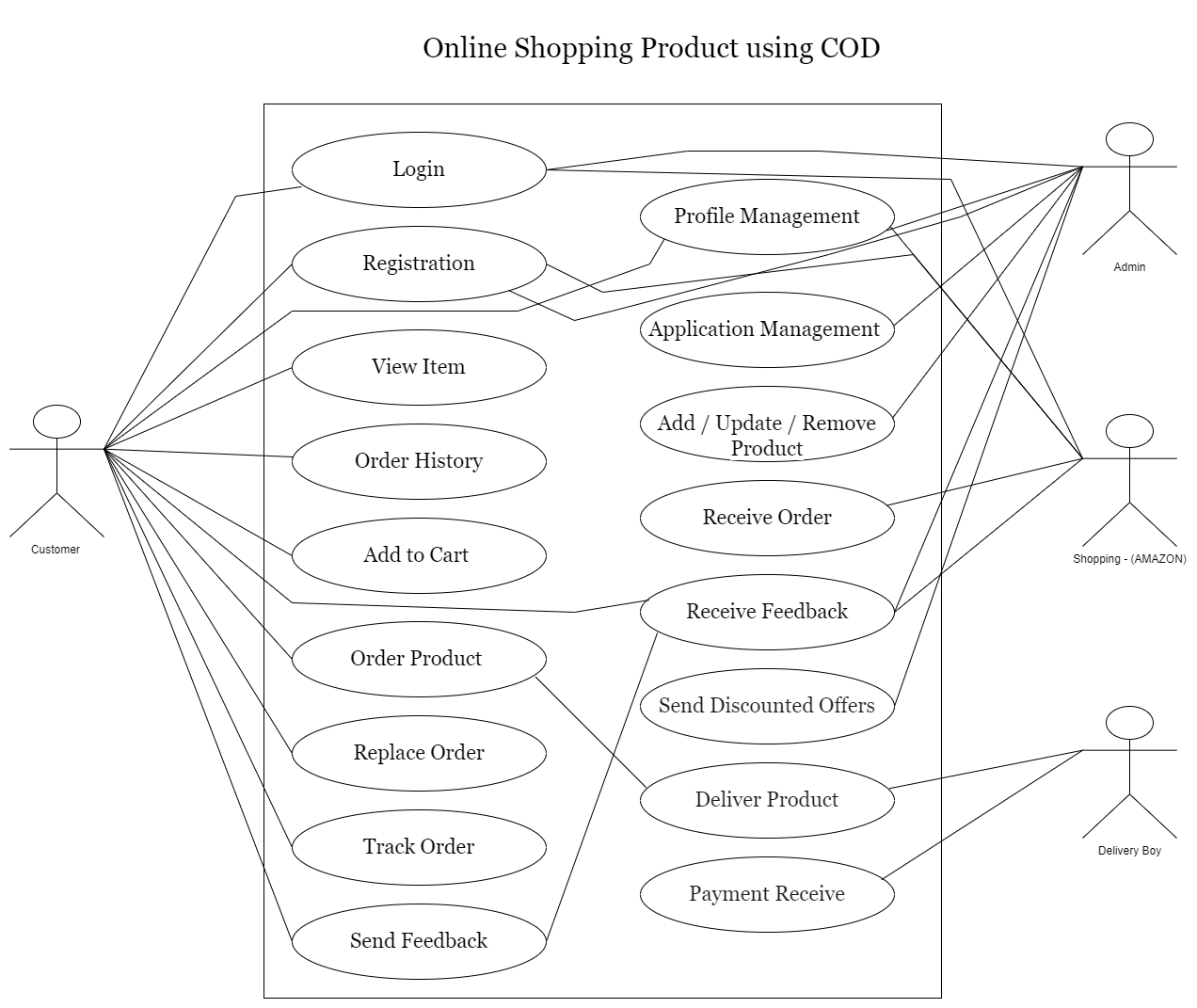
* Little or no little planning required.
* Easy to manage
* Give flexibility to developers.
* Functionality can be developed rapidly and demonstrated
* Delivers early partial working solutions
* Suitable for fixed or changing requirements
* Resource requirements are minimum.

**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.
* 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.

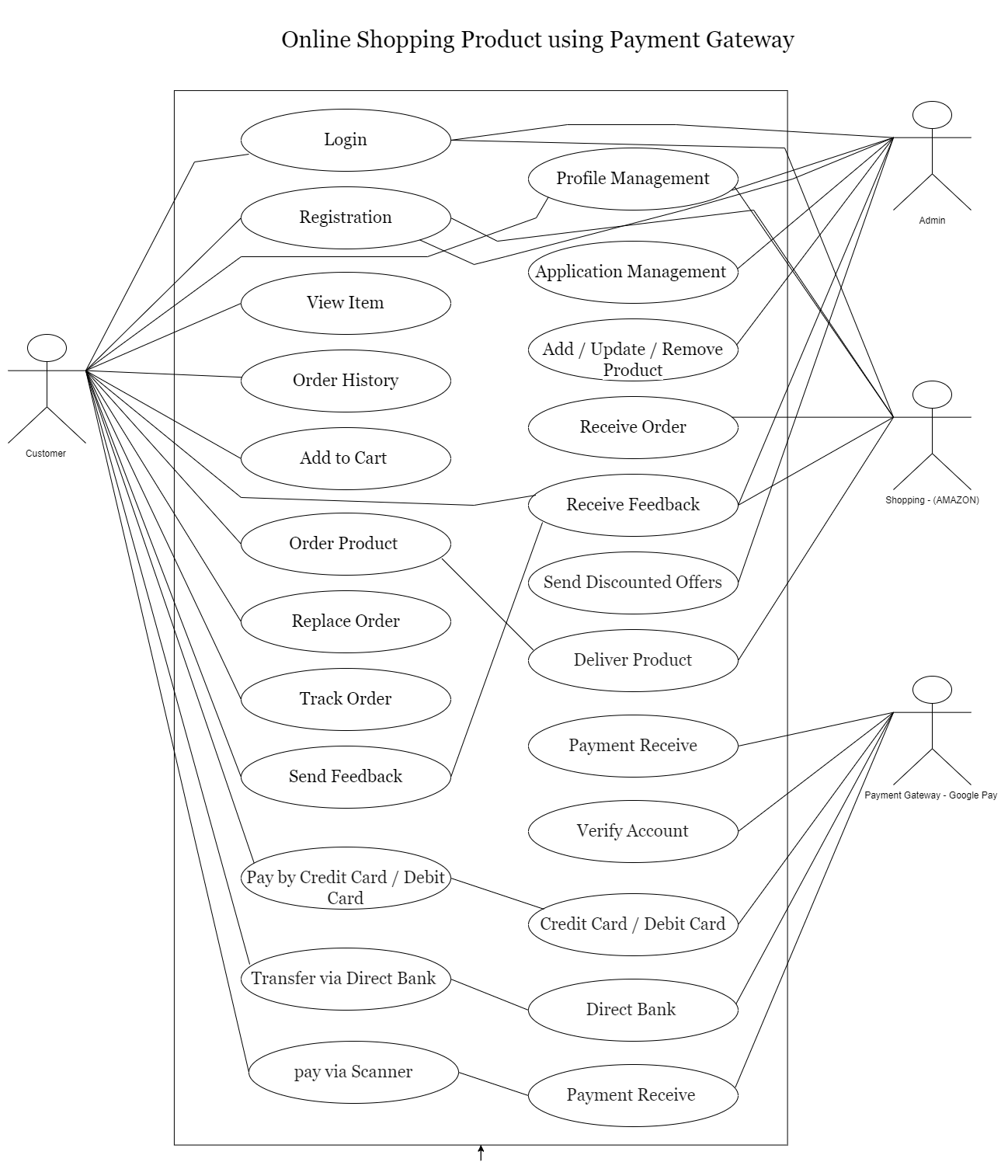
**Draw usecase on Online Shopping product using COD**

USECASE :



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

USECASE :

****