Write agile manifesto principles

- **Individuals and interactions**
  In agile development individual interactions and motivations are key factors to develop a good software. individual interactions are important for working in teams and around other individuals.
- **Working software**
  A working software to show the client is preferred over just showing documents of how the development. It is easy to understand client's requirements with a working software instead of depending on a document.
- **Customer collaboration**
  To gather Customer requirements interaction with customer are required to come to an understanding about the requirements.
- **Responding to change**
  Agile development is all about quick change in the development if and improvement is needed or to continue to develop. It's a changing process not a ironclad plan to follow.

What is agile methodology?

Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. Agile model represents the work flow of a software by making the development process simple and in multiple smaller parts that can be integrated after words or Added in the future. Its iterations and durations are decided before the implementation starts. Once the work begins, teams cycle through a process of planning, executing, and evaluating.

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

Fig. agile model

Following are the phases in the Agile model are as follows:

1. **Requirements gathering**
   The requirements to develop the are defined here from the business opportunities to the plan time, needed resources for the project. Feasibility of the software is checked in terms of business, technical, Economic etc.

2. **Design the requirements**
   After Gathering the requirements user interaction and flow diagrams are made to denote new features and how to interconnect them to the existing system.

3. **Construction/ iteration**
   After designing the software implementation on the work documents is done to generate a useable working software to start the project development. The functionality can be minimum at start.

4. **Testing/ Quality assurance**
   Testing on the software is done to check for bugs or errors and to check its performance.

5. **Deployment**
   The product is deployed for UAT and is made available to run on the user environment.

6. **Feedback**
   After UAT is done the feedback of the product is taken to check faults or future improvements are needed for the product.

Advantages:

1. Frequent Delivery
2. Face-to-Face Communication with clients.
3. Efficient design and fulfils the business requirement.
4. Anytime changes are acceptable.
5. It reduces total development time.

Dis-Advantages:

1. Misinterpretation can be caused by the shortage of formal documents, when checked by different team members.
2. maintenance of the finished project can become a difficulty if the human resources are allocated to multiple projects due to lack of proper documentation.
3. Human resources required needed to be highly skilled.

What is SDLC ?

SDLC is software development life cycle. It is used to define how and using what type of methods and models the software could be generated or developed. It has different phases to ensure the development can go smoothly and Efficiently. It aims to produce a high-quality system that meets or exceeds customer expectations, works effectively and efficiently.

What is software testing ?

Software testing is set of processes to check the software for bugs and errors , to ensure its correctness or efficiency as per the requirements of the client or the SRS.

What is SRS ?

SRS (Software Requirement Specifications) comprises user requirements for a system as well as detailed specifications of the system requirements. The SRS is a specification for a specific software product, program, or set of applications that perform particular functions in a specific environment.

Write SDLC phases with basic introduction

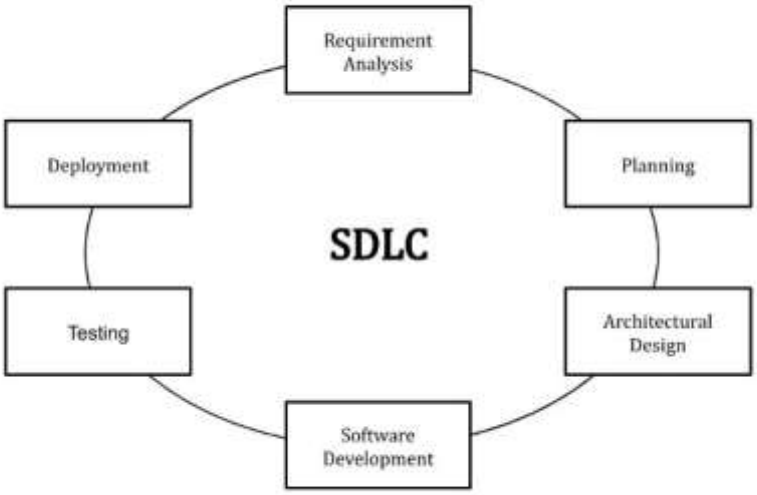


Fig. SDLC

SDLC - Software Development Life Cycle has 6 Phases.

1. **Requirement Phase**
   Requirement gathering and analysis is the most important phase in the software development lifecycle. Details about What client wants and how the product should function are gathered and documented.

2. **Analysis Phase**
   Once the requirement gathering and analysis is done the next step is to define and document the product requirements and get them approved by the customer. A Document is made using the all the product requirements to be designed and developed during the project life cycle. The Document is called the SRS (Software Requirement Specification) document.

3. **Design Phase**
   It describes how each and every feature in the product should work and how every component should work. It gives the architecture of the software product to be developed only the design will be there and not the code. Based on the SRS product is Designed in this phase.

4. **Development Phase**
   By using Documents from previous phases a product is made or generated using the SRS. coding and implementation of the software is done. At the end of this phase product is made.

5. **Testing Phase**
   After the software is made testing of the software manually or using automated testing tools depends on the process. testing is to check its correctness, useability, function, efficiency. The software is tested in this phase and checked if it meets the Requirements of SRS.

6. **Maintenance Phase**
   AFTER the testing is done the software is deployed for use. once deployed the developed system can come up with problems and it is needed to fix it. These issues found are fixed or patched using updates. The Maintainace of the software is done as per SLA (Service Level Agreement).

Explain Phases of the waterfall model.

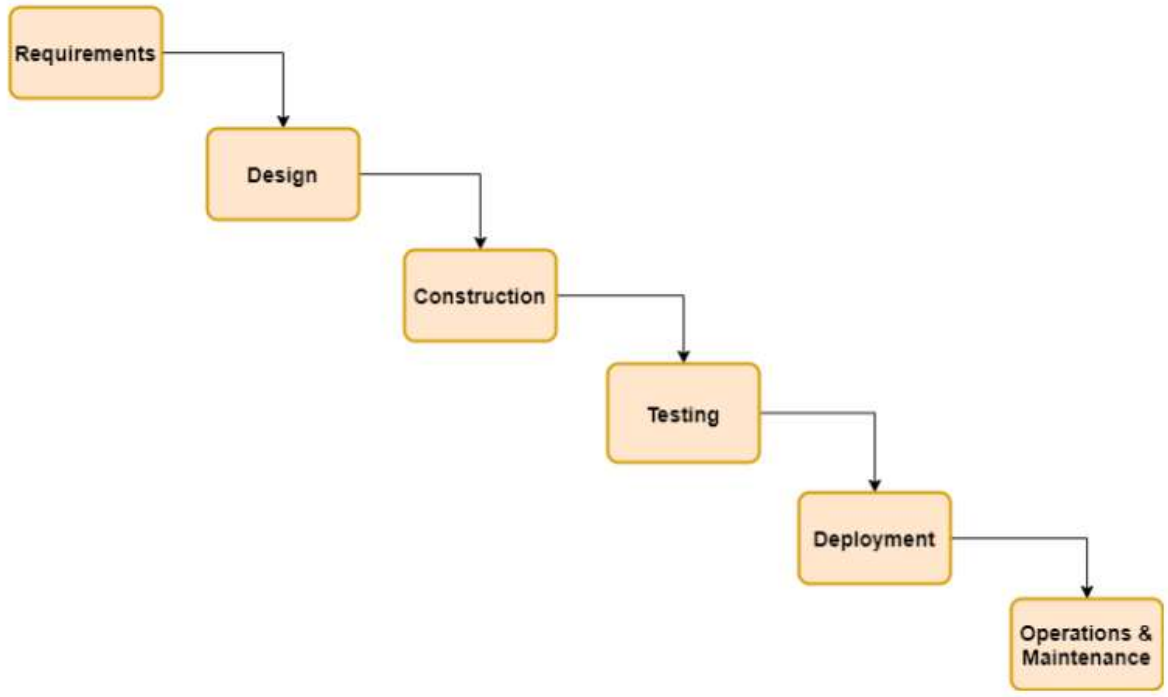


Fig. Waterfall model

The waterfall model is divided into various stages, which are as follows:

1. **Requirement collection**
   Requirement collection is done by noting clients needs and requirement in a SRS Document. The document should be easy to understand and all requirements should be clearly listed. Requirements are not changed after this stage.

2. **Feasibility study**
   A Feasibility study is done after the SRS is made to check it by the multiple point of views that the project can be made or not. the Study is done on multiple parameters like legal, technical, operational feasibility, Economic, Schedule.

3. **Design**
   After feasibility study the Design part of the project is done. From what architecture to use to what type of different hardware or software, programming language, structure to use is decided here.

4. **Coding**
   implementation of the designed software is done in this stage. The software is coded using programing languages and made to work as per the SRS.

5. **Testing**
   After coding stage the developed software is checked for function, correctness and useability. The software is checked for bugs. if there are bugs the developer will verify that the given bug is valid or not. If it is correct, it will be fixed by the developer and change with the new one. After that tester will re-test it and verify that the bug is fixed or not.

6. **Installation**
   After testing the software, it is deployed for use as per the client. if the SRS requirements are satisfied When the application is stable, it will install into the client's environment for their use.

7. **Maintenance**
   After Deployment there may be some issues that need to be tested and fixed. Taking care of the product, time to time is called the maintenance, which includes the variations that happen in the hardware and software to maintain the operational effectiveness and also increase the performance. Patches or updates are provided to solve that problem.

Write phases of spiral model.

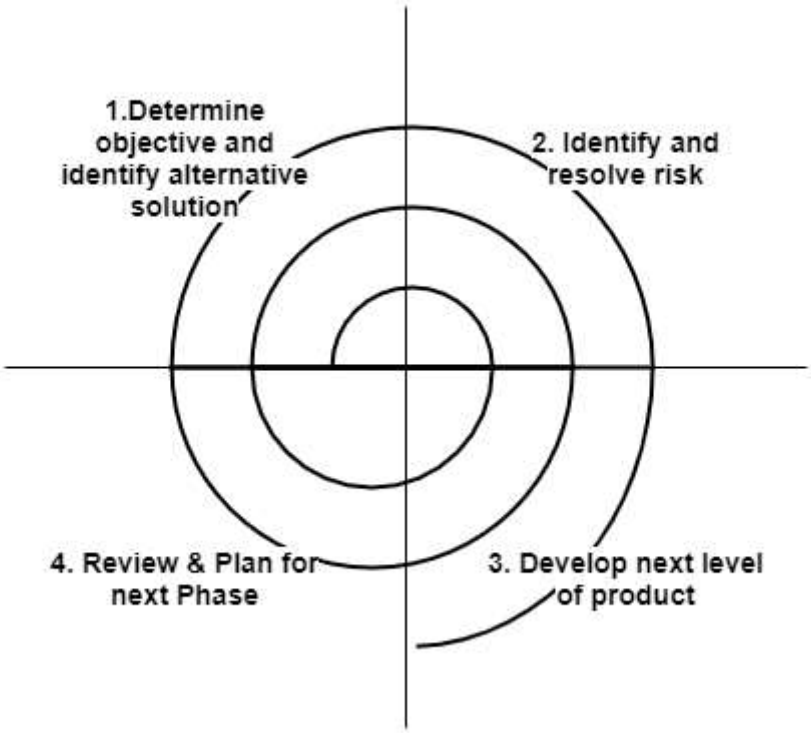


Fig. Spiral model

The different phases of the spiral model are as follows:

1. **Requirement analysis**
   First phase starts with the requirement of the software the requirement for the whole software subsystems and unit requirements are documented to understand client requirement.

2. **Design**
   In this phase the product is logically designed. The logical design, architectural design, flow charts and other representations are made to design the product.

3. **Coding**
   In this phase the product is made as per the requirements of the client. it refers to the construction of the real application in every cycle.

4. **Testing and risk analysis**
   In this phase, test the build at the end of the first cycle and also analyze the risk of the software on the different aspects such as managing risks, detecting, and observing the technical feasibility. Build is given to the client for the test and feedback on the product.

What is oops ?

OOP (object oriented programing) is a programing structure that organizes its data using objects, class, methods and attributes. oops is a programing methodology that generates class and objects to define relations and communication between data.

Write Basic Concepts of oops

- **Objects**
  Object is a class variable or an instance of class.an object can be anything as long as it has attributes and behaviours.

- **Classes**
  Class is a user defined data type. It is a blueprint of data and member functions. it is a logical entity. It is a collection of objects.

- **Data abstraction**
  Abstraction refers to the act of representing essential features without including the background details.
  Hiding internal details and showing functionality is known as abstraction. For example phone call, we don't know the internal processing.

- **Data encapsulation**
  Wrapping up of data and functions into a single unit is called as data encapsulation. It is the way of combining both data and the functions that operate on that data under a single unit.

- **Inheritance**
  It is a process by which object of one class inherit the properties of objects of another class. It is the capability to define a new class in terms of an existing class. An existing class is known as a base class and the new class is known as derived class.

- **Polymorphism**
  Polymorphism is the concept that supports the capability of data to be processed in more than one form.

What is object?

Object is a instance of a class. Objects can correspond to real-world objects or an abstract entity. Anything can be an object. Object consists of two parts data and method. Any entity that has state and behavior is known as an object. It can be physical or logical.

Ex: pen, ball, chair etc.

What is class?

Classes are user-defined data types that act as the blueprint for individual objects, attributes and methods. Collection of objects is called class. It is a logical entity.

Ex: pen is a class where there are different types of pen in it.

What is encapsulation?

This principle states that all important information is contained inside an object and only select information is exposed. The implementation and state of each object are privately held inside a defined class. Other objects do not have access to this class or the authority to make changes. They are only able to call a list of public functions or methods. This characteristic of data hiding provides greater program security and avoids unintended data corruption.

Binding (or wrapping) code and data together into a single unit are known as encapsulation.

What is inheritance?

Classes can reuse code from other classes. Relationships and subclasses between objects can be assigned, enabling developers to reuse common logic while still maintaining a unique hierarchy. This property of OOP forces a more thorough data analysis, reduces development time and ensures a higher level of accuracy.

When one object acquires all the properties and behaviors of a parent object, it is known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

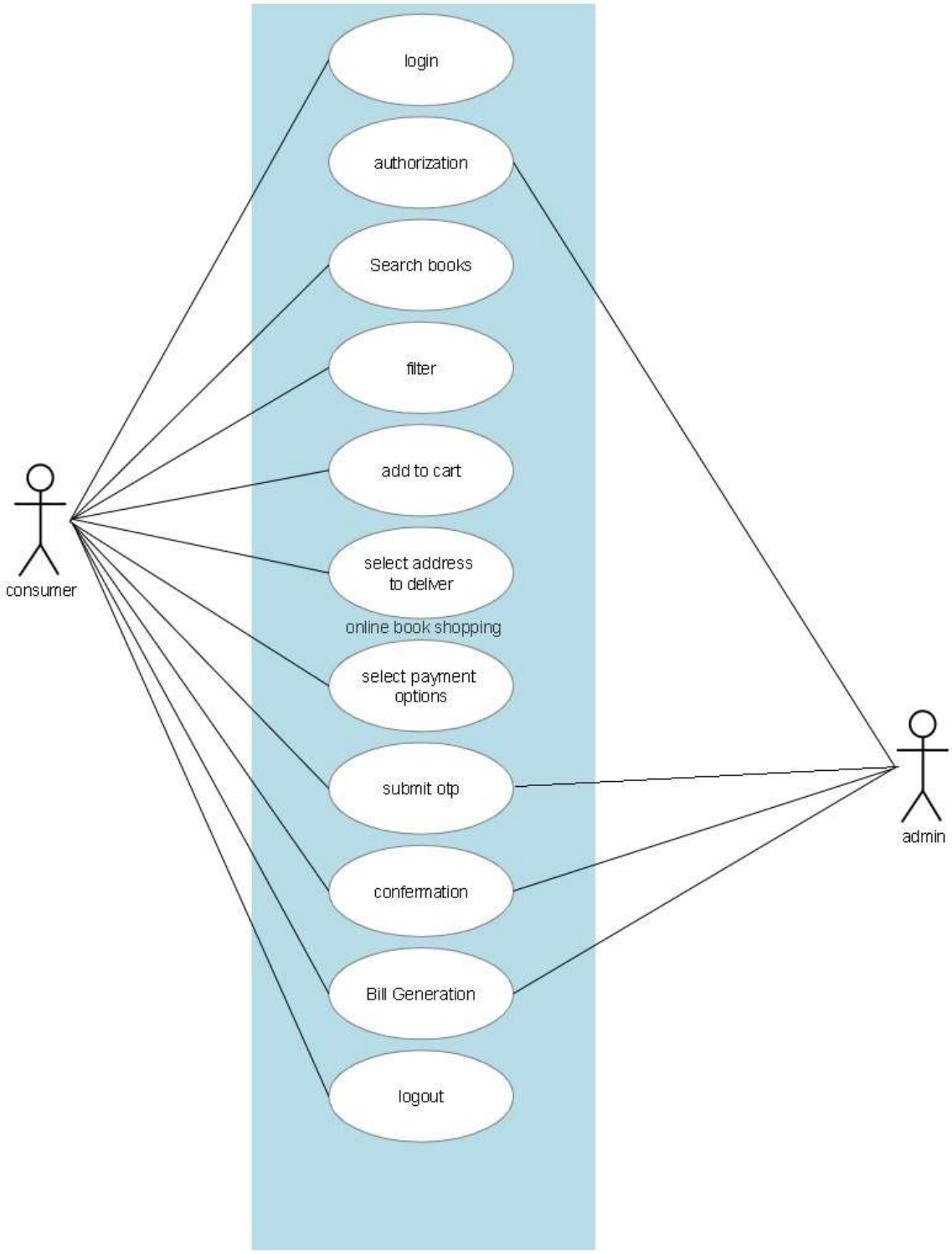
EX: From parent table to child table some properties are shares similarities or shares some data.

What is polymorphism?

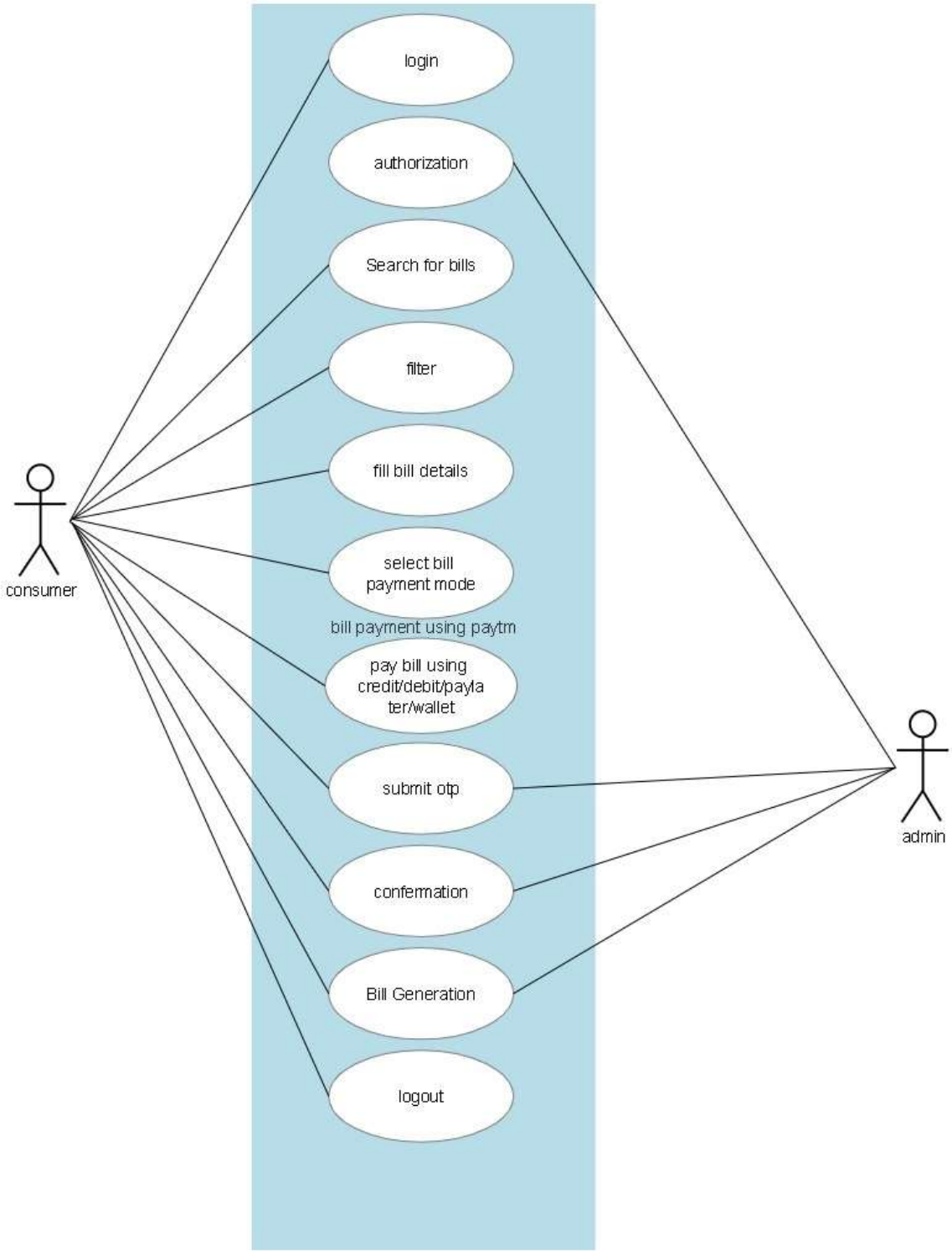
Objects are designed to share behaviors and they can take on more than one form. The program will determine which meaning or usage is necessary for each execution of that object from a parent class, reducing the need to duplicate code. A child class is then created, which extends the functionality of the parent class. Polymorphism allows different types of objects to pass through the same interface. The ability to change form is known as polymorphism.

If one task is performed in different ways, it is known as polymorphism.

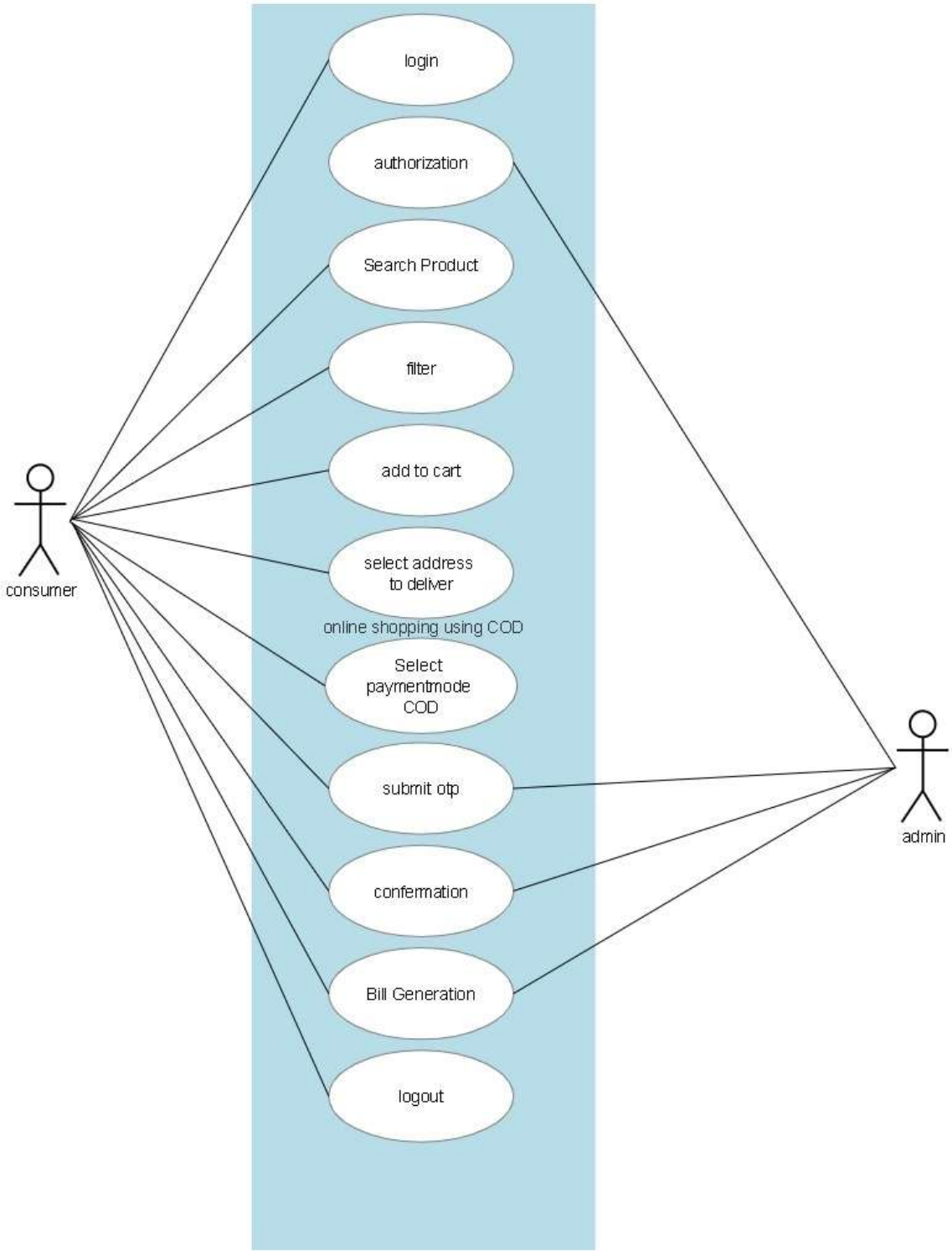
Draw Usecase on Online book shopping



Draw Usecase on online bill payment system (paytm)



Draw usecase on Online shopping product using COD.



Draw usecase on Online shopping product using payment gateway.