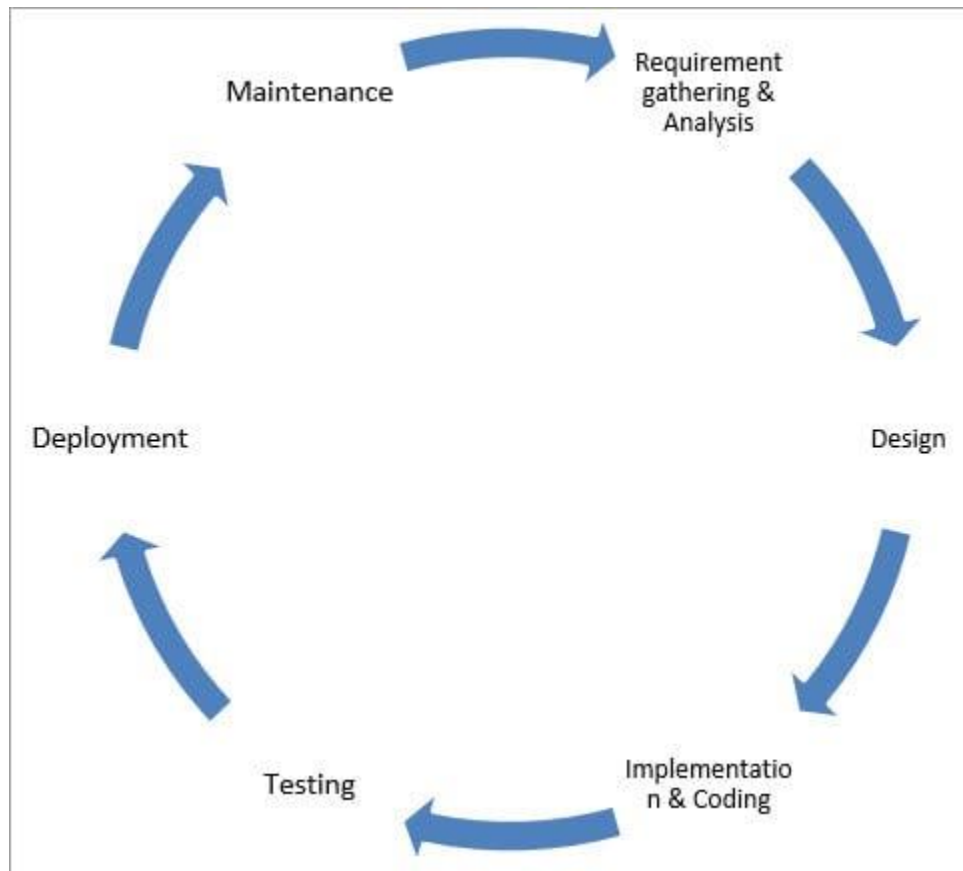


Module - 1

1. What is SDLC?

- SDLC is a structure imposed on the development of a software Product that defines the process for planning implementations, testing, documentation, deployment and on going maintainance and support.

2. Write SDLC phases with basic Introduction ?



1. Requirement gathering & Analysis :

- During this phase, all the relevant information is collected from the customer to develop a product as per their expectation.
- There for two types of Requirements.

- 1. Functional Requirements :** A functional requirement defines a function of a system or its component, where a function is described as a specification of behavior between inputs and outputs,
 - 2. Non-Functional Requirements :** A non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours.
- Once the requirement gathering is done, an analysis is done to check the feasibility of the development of a product. In case of any ambiguity, a call is set up for further discussion.
 - Once the requirement is clearly understood, the SRS (Software Requirement Specification) document is created. This document should be thoroughly understood by the developers and also should be reviewed by the customer for future reference.

2. Design :

- The Design Phase is an essential phase of the Software Development Life Cycle. The list of requirements that you develop in the definition phase is used to make design choices. In the design phase, one or more designs are created to achieve the project result. Depending on the project subject, the design phase products include dioramas, flow-charts, sketches, site trees, HTML screen designs, photo impressions, prototypes, and UML schemas.
- Two types of design :
 1. HLD (High Level Design) :
 2. LLD (Low Level Design) :

3. Implementation & Coding :

- It is longest phase of SDLC.
- In this phase developer start build the entire system by writing code using the chosen programming language.

4. Testing :

- It is also known as validation phase of SDLC.
- Testing starts once the coding is complete and the modules are released for testing. In this phase, the developed software is tested thoroughly and any defects found are assigned to developers to get them fixed.
- Retesting, regression testing is done until the point at which the software is as per the customer's expectation. Testers refer SRS document to make sure that the software is as per the customer's standard.

5. Deployment :

- Once the product is tested, it is deployed in the production environment or first UAT (User Acceptance Testing) is done depending on the customer expectation.
- In the case of UAT, a replica of the production environment is created and the customer along with the developers does the testing. If the customer finds the application as expected, then sign off is provided by the customer to go live.

6. Maintenance :

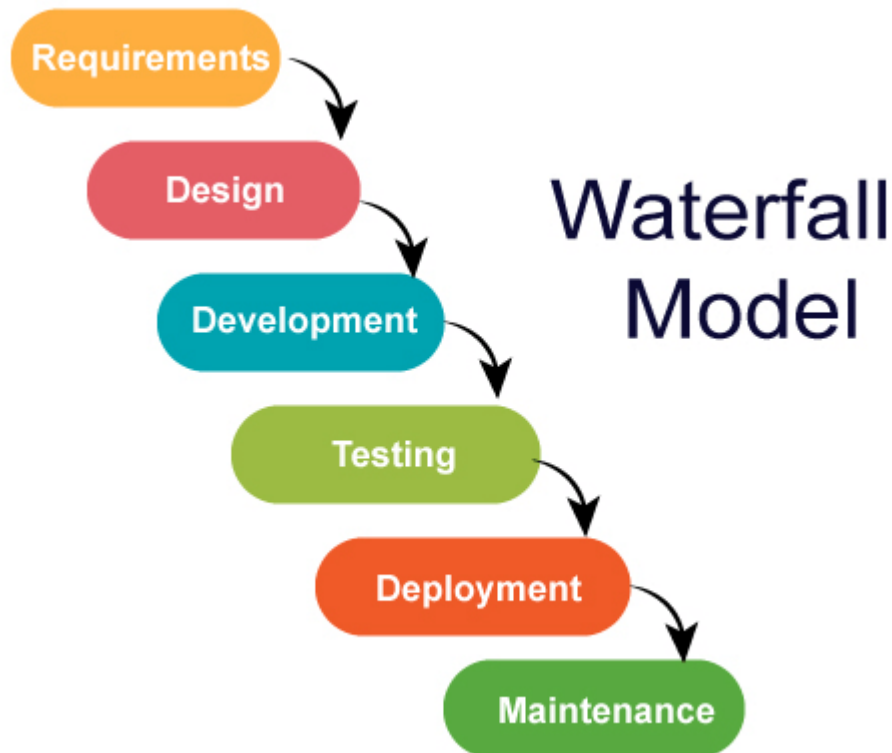
- In a software lifetime, type of maintenance may vary based on its nature. It may be just a routine maintenance task as some bug discovered by some user or it may be a large event in itself based on maintenance size or nature.
- Following are some types of maintenance based on their characteristics :
 1. **Corrective Maintenance** : This includes modifications and updates done in order to correct or fix problems, which are either discovered by user or concluded by user error reports.
 2. **Adaptive Maintenance** : This includes modifications and updates applied to keep the software product up-to-date and tuned to the ever-changing world of technology and business environment.
 3. **Perfective Maintenance** : This includes modifications and updates done in order to keep the software usable over long

period of time. It includes new features, new user requirements for refining the software and improve its reliability and performance.

3. What is SRS ?

- SRS Stands for software requirements specification.
- SRS is a document that captures complete description about how the system is expected to perform. It is usually signed off at the end of requirements engineering phase.

4. Explain phases of the waterfall Model ?



- Waterfall Model is the very first model that is used in SDLC. It is also known as the linear sequential model.
- In this model, the outcome of one phase is the input for the next phase. Development of the next phase starts only when the previous phase is complete.

- First, Requirement gathering and analysis is done. Once the requirement is freeze then only the System Design can start. Herein, the SRS document created is the output for the Requirement phase and it acts as an input for the System Design.
- In System Design Software architecture and Design, documents which act as an input for the next phase are created i.e. Implementation and coding.
- In the Implementation phase, coding is done and the software developed is the input for the next phase i.e. testing.
- In the testing phase, the developed code is tested thoroughly to detect the defects in the software. Defects are logged into the defect tracking tool and are retested once fixed. Bug logging, Retest, Regression testing goes on until the time the software is in go-live state.
- In the Deployment phase, the developed code is moved into production after the sign off is given by the customer.
- Any issues in the production environment are resolved by the developers which come under maintenance.

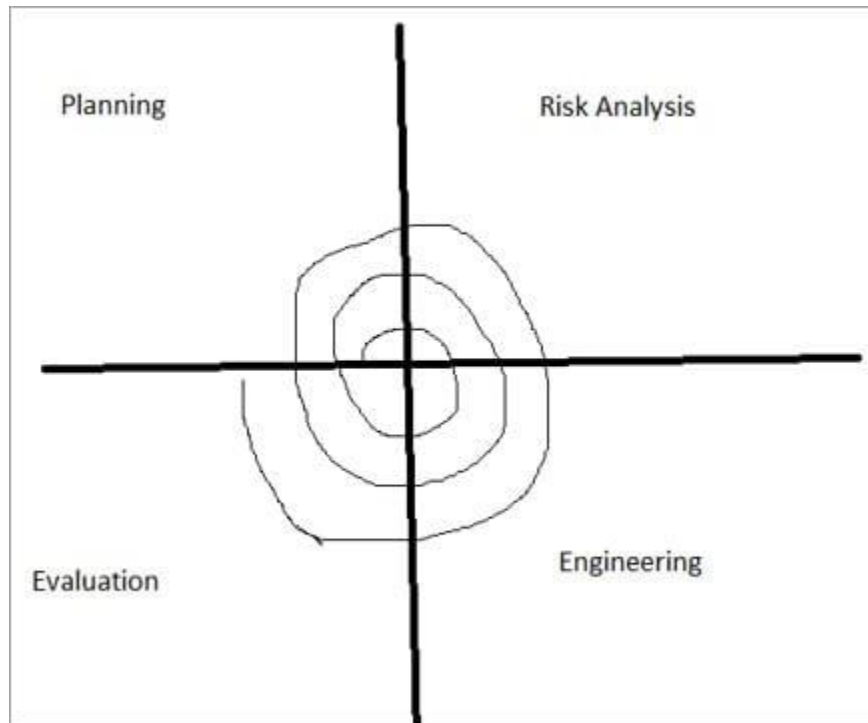
Advantages of the Waterfall Model:

- Waterfall model is the simple model which can be easily understood and is the one in which all the phases are done step by step.
- Deliverables of each phase are well defined, and this leads to no complexity and makes the project easily manageable.

Disadvantages of Waterfall model:

- Waterfall model is time-consuming & cannot be used in the short duration projects as in this model a new phase cannot be started until the ongoing phase is completed.
- Waterfall model cannot be used for the projects which have uncertain requirement or wherein the requirement keeps on changing as this model expects the requirement to be clear in the requirement gathering and analysis phase itself and any change in the later stages would lead to cost higher as the changes would be required in all the phases.

5. Write Stages of spiral model ?



- The Spiral Model includes iterative and prototype approach.
- Spiral model phases are followed in the iterations. The loops in the model represent the phase of the SDLC process i.e. the innermost loop is of requirement gathering & analysis which follows the Planning, Risk analysis, development, and evaluation. Next loop is Designing followed by Implementation & then testing.
- **Spiral Model has four phases:**
 - Planning
 - Risk Analysis
 - Engineering
 - Evaluation

(i) Planning:

The planning phase includes requirement gathering wherein all the required information is gathered from the customer and is

documented. Software requirement specification document is created for the next phase.

(ii) Risk Analysis:

In this phase, the best solution is selected for the risks involved and analysis is done by building the prototype.

For Example, the risk involved in accessing the data from a remote database can be that the data access rate might be too slow. The risk can be resolved by building a prototype of the data access subsystem.

(iii) Engineering:

Once the risk analysis is done, coding and testing are done.

(iv) Evaluation:

Customer evaluates the developed system and plans for the next iteration.

Advantages of Spiral Model:

- Risk Analysis is done extensively using the prototype models.
- Any enhancement or change in the functionality can be done in the next iteration.

Disadvantages of Spiral Model:

- The spiral model is best suited for large projects only.
- The cost can be high as it might take a large number of iterations which can lead to high time to reach the final product.

6. What is oops?

- OOPS is Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data and code: data in the form of fields (often known as attributes or *properties*), and code, in the form of procedures (often known as methods).
- A common feature of objects is that procedures (or methods) are attached to them and can access and modify the object's data fields.

7. Write basic concept of oops?

- There are some following concept of oops.
 1. Object
 2. Class
 3. Encapsulation
 4. Inheritance
 5. Polymorphism
 6. Abstraction

8. What is Object?

- Object is an encapsulation of data along with functions that act upon that data.
- An object consists of:
 1. **Name** = the variable name we give it
 2. **Member data** = the data that describes the object
 3. **Member functions** = behavior aspects of the object (functions related to the object itself).

9. What is Class?

- Class is a blueprint for objects. A class is a user-defined type that describes what a certain type of object will look like. A class description consists of a declaration and a definition.

10. What is Encapsulation?

- Encapsulation prevents external code from being concerned with the internal workings of an object.
- Encapsulation in Java is a *process of wrapping code and data together into a single unit*, for example, a capsule which is mixed of several medicines.

11. What is Polymorphism?

- We can define polymorphism as the ability of a message to be displayed in more than one form.
- A real life example of polymorphism, a person at the same time can have different characteristics. Like a man at the same time is a father, a husband, an employee.

12. What is Inheritance?

- inheritance is the mechanism of basing an object or class upon another object (prototype-based inheritance) or class (class-based inheritance), retaining similar implementation. Also defined as deriving new classes (sub classes) from existing ones such as super class or base class and then forming them into a hierarchy of classes.
- There are various types of inheritance.

1. Single inheritance

- where subclasses inherit the features of one superclass. A class acquires the properties of another class.

2. Multiple inheritance

- where one class can have more than one superclass and inherit features from all parent classes.

3. Multilevel inheritance

- Where a subclass is inherited from another subclass. It is not uncommon that a class is derived from another derived class as shown in the figure "Multilevel inheritance".
- The class *A* serves as a *base class* for the *derived class B*, which in turn serves as a *base class* for the *derived class C*. The class *B* is known as *intermediate* base class because it provides a link for the inheritance between *A* and *C*.

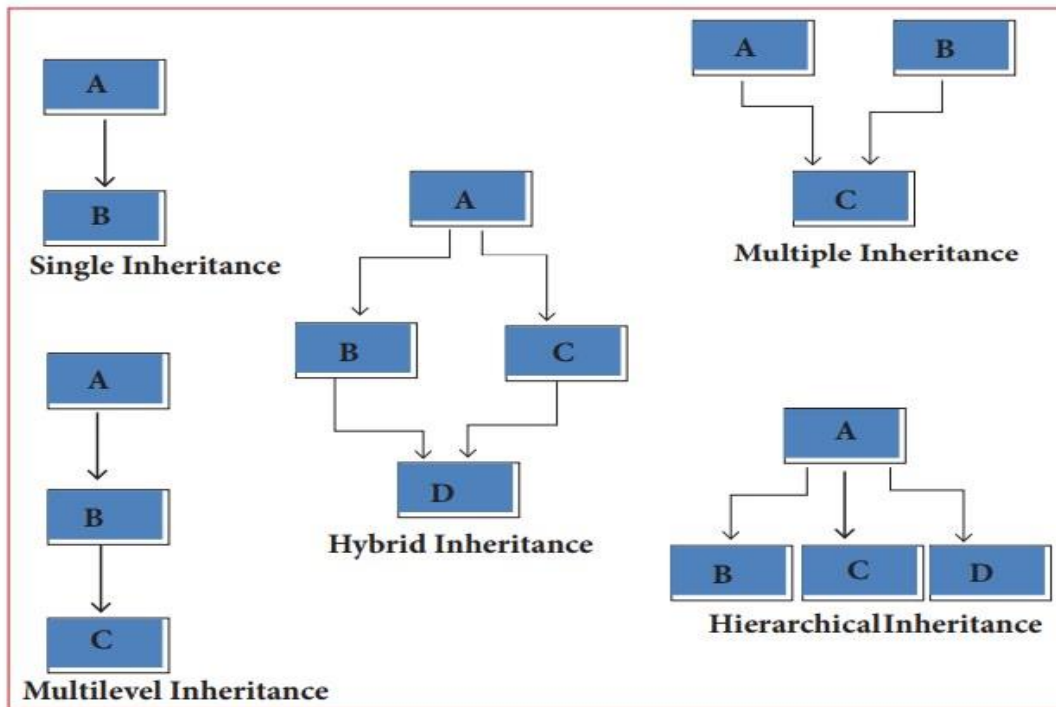
4. Hierarchical inheritance

- This is where one class serves as a superclass (base class) for more than one sub class. For example, a parent class, *A*, can have two subclasses *B* and *C*. Both *B* and *C*'s parent class is *A*, but *B* and *C* are two separate subclasses.

5 . Hybrid inheritance

- Hybrid inheritance is when a mix of two or more of the above types of inheritance occurs. An example of this is when class *A* has a

subclass B which has two subclasses, C and D. This is a mixture of both multilevel inheritance and hierarchal inheritance.



13. What is RDBMS?

- A relational database management system (RDBMS) is a database management system that is based on a relational model. The relational model uses the basic concept of a relation or table. RDBMS is the basis for SQL, and for database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

14. What is SQL?

- SQL is Structured Query Language (SQL) is a domain-specific language used in programming and designed for managing data held in a relational database management system (RDBMS), or for stream processing in a relational data stream management system (RDSMS).for example, Microsoft SQL Server, MySQL, Oracle.

15. Write SQL command.

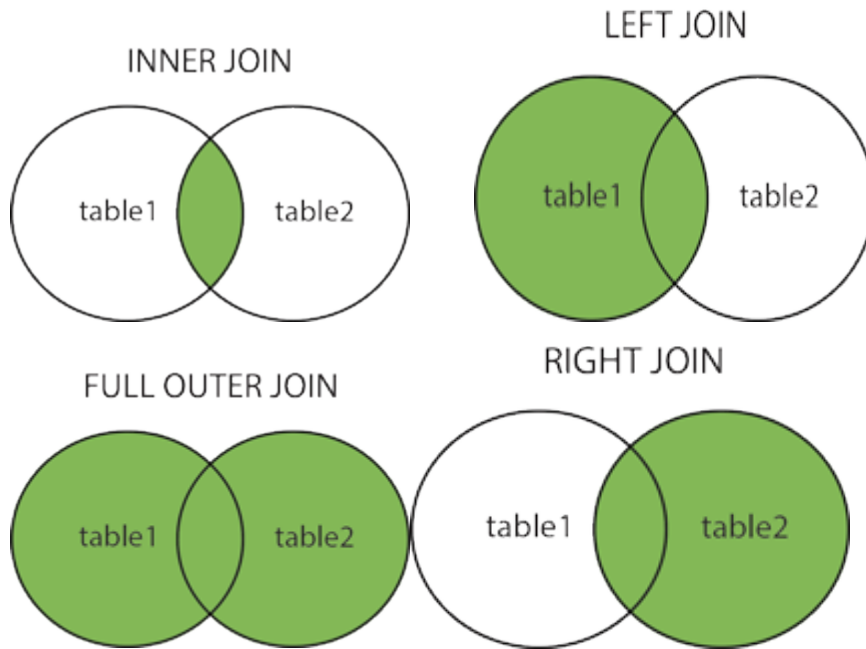
- SQL statements are categorized into four different types of command, which are
 1. **DML (DATA MANIPULATION LANGUAGE):**
 - The Data Manipulation Language allows to manipulate the information. You can insert data, Update data, Delete data.
 2. **DDL (DATA DEFINITION LANGUAGE):**
 - The Data Definition Language allows to create, alter, drop and database objects.
 3. **DCL (DATA CONTROL LANGUAGE):**
 - The main commands are the GRANT, REVOKE and DENY commands.
 4. **DQL (DATA QUERY LANGUAGE):**
 - The main command is SELECT
 - **Select** : It is used to retrieve data from the database.

16. What is join?

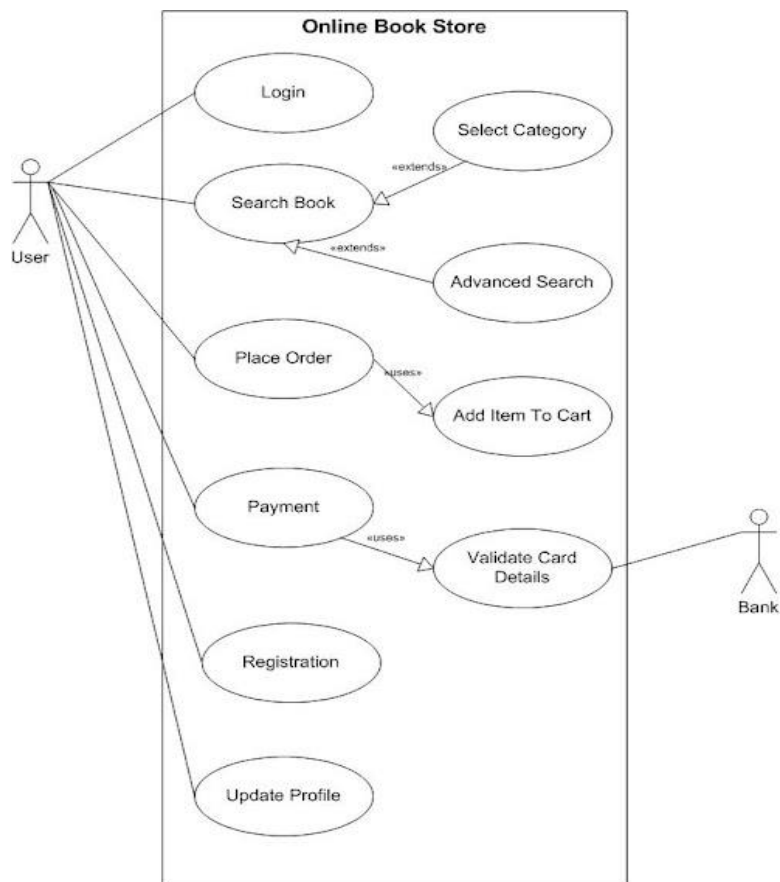
- A **JOIN** clause is used to combine rows from two or more tables, based on a related column between them.

17. Write types of join.

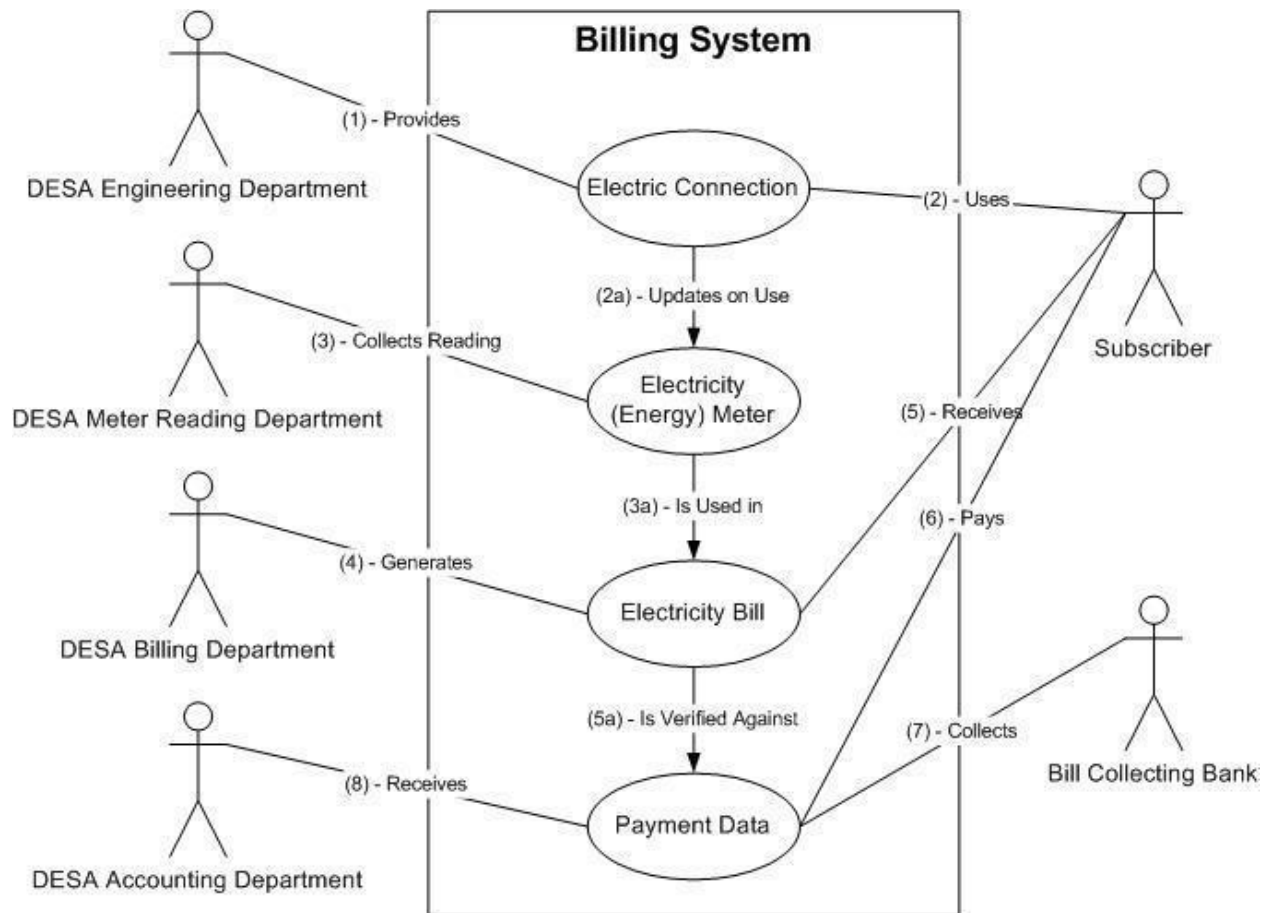
- Here are the different types of the JOINS in SQL:
 - **(INNER) JOIN**: Returns records that have matching values in both tables
 - **LEFT (OUTER) JOIN**: Returns all records from the left table, and the matched records from the right table
 - **RIGHT (OUTER) JOIN**: Returns all records from the right table, and the matched records from the left table
 - **FULL (OUTER) JOIN**: Returns all records when there is a match in either left or right table



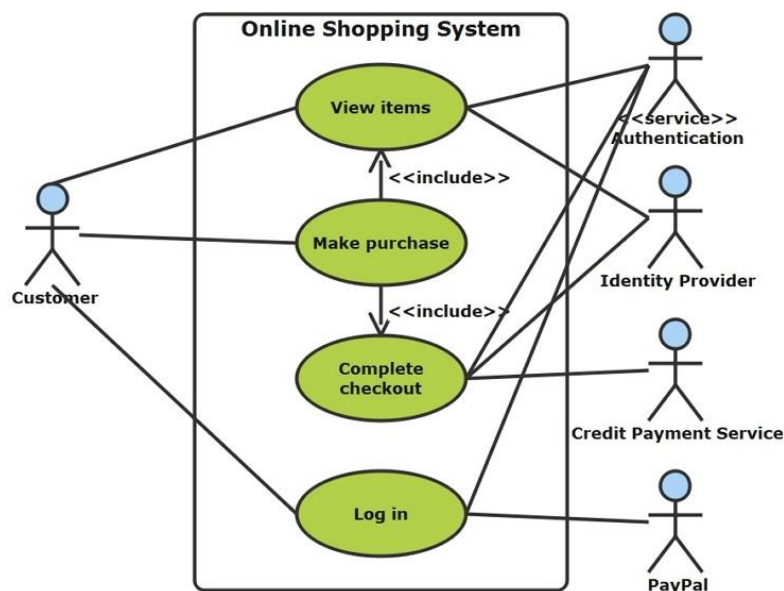
18. Draw usecase on Online book shopping.



19. Draw usecase on Online bill payment system.



20. Draw usecase on Online shopping product-using COD.



21. Draw usecase on online shopping product using payment gateway.

