**What is SDLC?**

**SDLC** is mainly called software development life cycle. This specifies the task to perform at different stages by software developer or engineer. Which include requirement planning (gathering), Analysis, Design, Coding, Testing and Maintence.

**Requirement plannin**g includes- what are the requirements needed to develop the product like software tools/hardware, requirement of peoples, product to complete (life span), cost, which methodology required etc.

**Analysis** includes- here we analyzing the by documentation called SRS.

**Design**- what we have gathered and make a documented will perform in designing phase like blue print.

**Coding**- here the actual work is started, we do coding here on the basis of information we gather and make a blueprint. Here developer’s work is started.

**Testing**- In this phase the actual work of tester is started. The work done so far is tested here and if defect is formed then further send to the developing term.

**Maintenance**- This is the last phase of any project here the product or application is maintained and get the feedback of the product/application and re-leased the product/application.

**What is agile methodology**?

**Agile** Methodology means “**Move Quickly”.** It is a combination of iterative and incremental process model. Where the large project are deal. Rapidly deploy of any application in much more organized way without having to wait for a longer time at the same time you want to make sure that you have small chunks of code that can be shipped to the client or whichever application you’re working with that’s the reason agile exist today.

**Pro**- Frequently Delivering, Face to Face communication with client, Changes are done frequently.

**Con-** Less documentation, strict delivery Management, Depends heavily on Customers interaction, Updated Technology.

**What is SRS?**

SRS is basically software Requirement Specification. It is a document which describe what will be the feature would be added to the application/product. Generally it includes Functional/Non-Functional with Use-cases.

**What is oops?**

OOPS- Object Oriented Programming-is a paradigm to design a program using classes and objects. It simplifies software development and maintenance by providing some concepts: Object, Class, Inheritance, Polymorphism, Abstraction, and Encapsulation.

**Write Basic Concepts of oops.**



**Object:** Any entity that has state and behavior is known as an object.

**Class:** Collection of objects is known as class.

**Inheritance:** When one object acquires all the properties and behaviors of a parent object, it is known as inheritance.

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

**Abstraction:** Hiding internal details and showing functionality is known as abstraction.

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

**What is Object?**

Any entity that has state and behavior is known as an object.

For example, a chair, pen, table, keyboard, bike, etc.

**What is Class?**

Collection of objects is known as class.

**What is Encapsulation?**

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

For example, a capsule, it is wrapped with different medicines.

**What is Inheritance?**

When one object acquires all the properties and behaviors of a parent object, it is known as inheritance.

**What is Polymorphism?**

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

**What is RDBMS?**

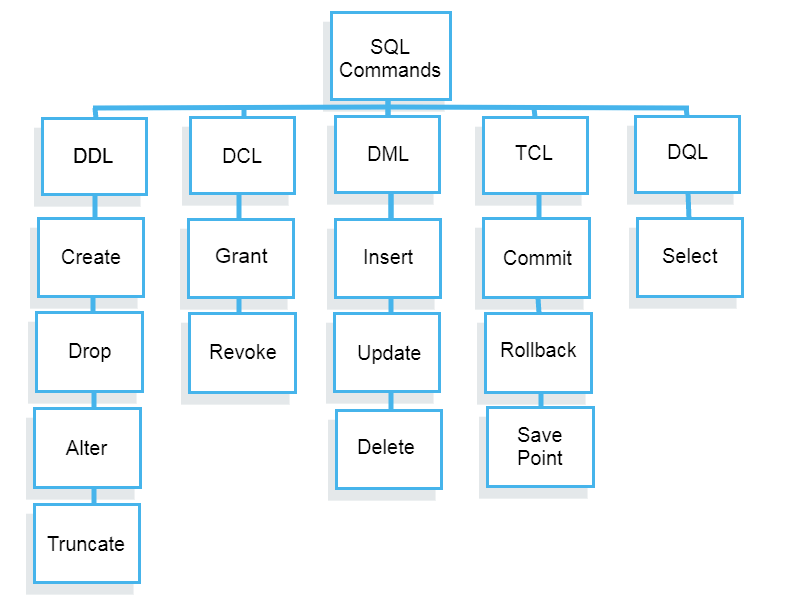
The software used to store, manage, query, and retrieve data stored in a relational database is known as relational database management system (RDBMS).

**What is SQL?**

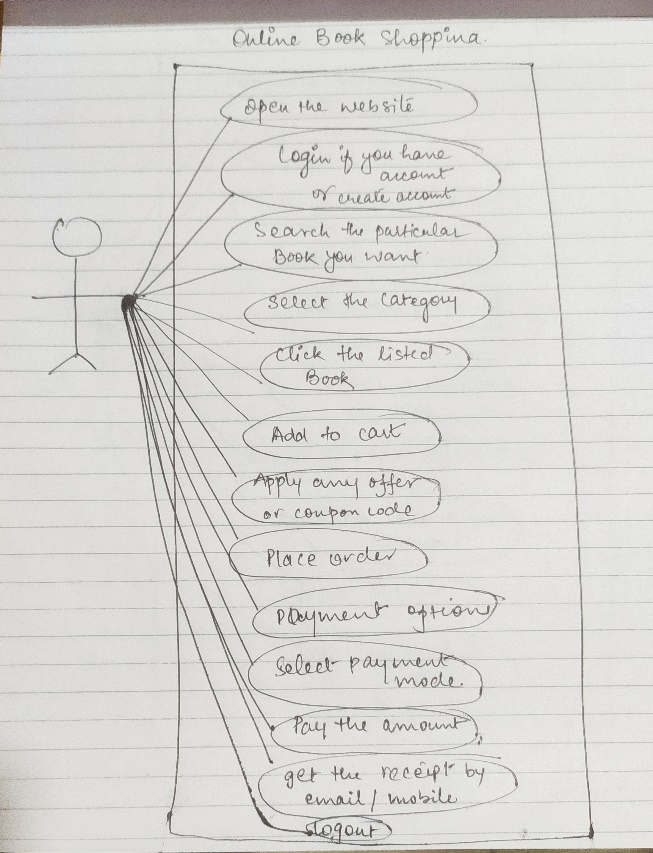
Structured Query Language (SQL) is a standardized programming language that is used to manage relational databases and perform various operations on the data in them.

**Write SQL Commands**

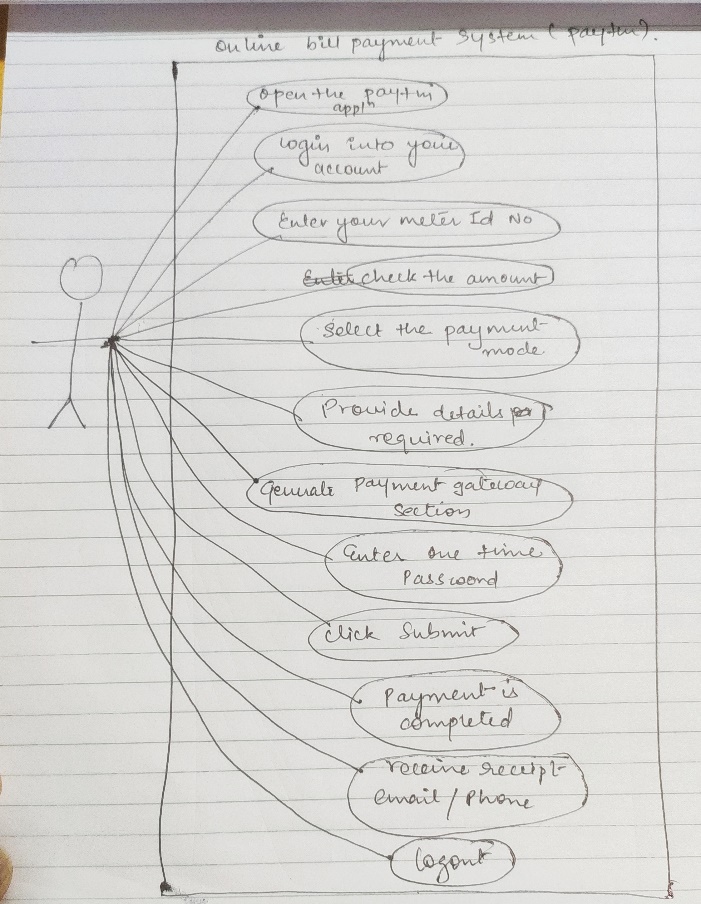
There are basically five SQL commands:



Draw Use case on online book shopping



Draw Use case on online bill payment system (paytm)



**Write SDLC phases with basic introduction**

SDLC (Software Development Life Cycle) is a framework that defines activities that are performed during the software development process. There are 6 phases in SDLC model as given below.

**Requirement:** In this phase, all the requirements are collected from the customer/client. They are provided in a document called Businessmen requirement specification (BRS) and System requirement specification (SRS). All the details are discussed with the customer/client in detail.

**Design:** It has two steps:

* **High-level design (HLD):** It gives the architecture of software products.
* **Low-level design (LLD):** It describes how each and every feature in the product should work and every component.

**Implementation:**

* This is the longest phase.
* This phase consists of Front end + Middleware + Back-end.
* **In front-end:**Development of coding is done even SEO settings are done.
* **In Middleware:** They connect both the front end and back end.
* **In the back-end:** A database is created.

**Testing:** Testing is carried out to verify the entire system. The aim of the tester is to find out the gaps and defects within the system and also to check whether the system is running according to the requirement of the customer/client.

**Maintenance:** Once the product has been delivered to the client a task of maintenance starts as when the client will come up with an error the issue should be fixed from time to time.

**Explain Phases of the waterfall model**

**Winston Royce introduced the Waterfall Model in 1970**.This model has **five phases**: Requirements analysis and specification, design, implementation, and unit testing, integration and system testing, and operation and maintenance. The steps always follow in this order and do not overlap. The developer must complete every phase before the next phase begins. This model is named "**Waterfall Model**", because its diagrammatic representation resembles a cascade of waterfalls.

**1. Requirements analysis and specification phase:** The aim of this phase is to understand the exact requirements of the customer and to document them properly. Both the customer and the software developer work together so as to document all the functions, performance, and interfacing requirement of the software. It describes the **"what**" of the system to be produced and not "how. “In this phase, a large document called **Software Requirement Specification (SRS)** document is created which contained a detailed description of what the system will do in the common language.

**2. Design Phase:** This phase aims to transform the requirements gathered in the **SRS** into a suitable form which permits further coding in a programming language. It defines the overall software architecture together with high level and detailed design. All this work is documented as a **Software Design Document (SDD**).

**3. Implementation and unit testing:** During this phase, design is implemented. If the SDD is complete, the implementation or coding phase proceeds smoothly, because all the information needed by software developers is contained in the SDD. During testing, the code is thoroughly examined and modified. Small modules are tested in isolation initially. After that these modules are tested by writing some overhead code to check the interaction between these modules and the flow of intermediate output.

**4. Integration and System Testing:** This phase is highly crucial as the quality of the end product is determined by the effectiveness of the testing carried out. The better output will lead to satisfied customers, lower maintenance costs, and accurate results. Unit testing determines the efficiency of individual modules. However, in this phase, the modules are tested for their interactions with each other and with the system.

**5. Operation and maintenance phase:** Maintenance is the task performed by every user once the software has been delivered to the customer, installed, and operational.

**Write phases of spiral model**

It has four stages or phases: The planning of objectives, risk analysis, engineering or development, and finally review. A project passes through all these stages repeatedly and the phases are known as a Spiral in the model.

1. **Determine objectives and find alternate solutions –** This phase includes requirement gathering and analysis. Based on the requirements, objectives are defined and different alternate solutions are proposed.
2. **Risk Analysis and resolving –**In this quadrant, all the proposed solutions are analyzed and any potential risk is identified, analyzed, and resolved.
3. **Develop and test:** This phase includes the actual implementation of the different features. All the implemented features are then verified with thorough testing.
4. **Review and planning of the next phase –**In this phase,the software is evaluated by the customer. It also includes risk identification and monitoring like cost overrun or schedule slippage and after that planning of the next phase is started.

**Write agile manifesto principles**

An individuals and interactions over processes and tools.

Working software over comprehensive documentation.

Customer collaboration over contract negotiation.

Responding to change over following a plan.

**What is join?**

**Join** statement is used to combine data or rows from two or more tables based on a common field between them.

**Write type of joins.**

Four types of joins are there:

Right join

Left join

Inner join

Full join

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

The meaning of Agile is swift or versatile. **“Agile process model**" refers to a software development approach based on iterative development. Agile methods break tasks into smaller iterations, or parts do not directly involve long term planning. The project scope and requirements are laid down at the beginning of the development process. Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

Each iteration is considered as a short time "frame" in the agile process model, which typically lasts from one to four weeks. The division of the entire project into smaller parts helps to minimize the project risk and to reduce the overall project delivery time requirements. Each iteration involves a team working through a full software development life cycle including planning, requirements analysis, design, coding, and testing before a working product is demonstrated to the client.



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

1. **Requirements gathering**
2. **Design the requirements**
3. **Construction/ iteration**
4. **Testing/ Quality assurance**
5. **Deployment**
6. **Feedback**

**Requirements gathering:** In this phase, you must define the requirements. You should explain business opportunities and plan the time and effort needed to build the project. Based on this inform.

**Design the requirements:** When you have identified the project, work with stakeholders to define requirements. You can use the user flow diagram or the high-level UML diagram to show the work of new features and show how it will apply to your existing system action, you can evaluate technical and economic feasibility.

**Construction/ iteration:** When the team defines the requirements, the work begins. Designers and developers start working on their project, which aims to deploy a working product. The product will undergo various stages of improvement, so it includes simple, minimal functionality.

**Testing:** In this phase, the Quality Assurance team examines the product's performance and looks for the bug.

**Deployment:** In this phase, the team issues a product for the user's work environment.

**Feedback:** After releasing the product, the last step is feedback. In this, the team receives feedback about the product and works through the feedback.

When to use Agile Model?

* When frequent changes are required.
* When a highly qualified and experienced team is available.
* When a customer is ready to have a meeting with a software team all the time.
* When project size is small.

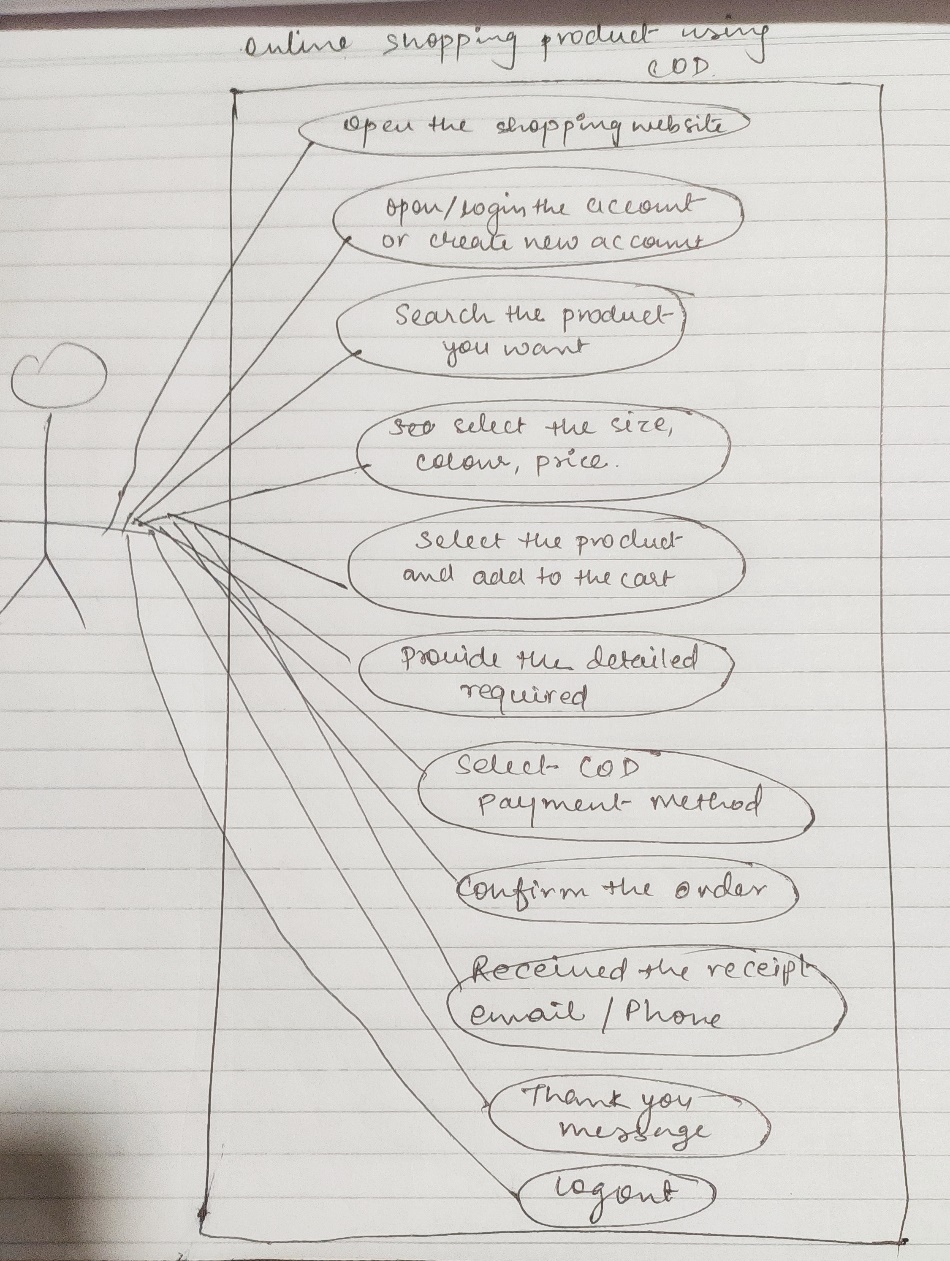
**Pros:**

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

**Cons:**

1. Due to the shortage of formal documents, it creates confusion and crucial decisions taken throughout various phases can be misinterpreted at any time by different team members.
2. Due to the lack of proper documentation, once the project completes and the developers allotted to another project, maintenance of the finished project can become a difficulty.

**Draw use case on Online shopping product using COD.**



**Draw use case on Online shopping product using payment gateway**

