**1.** **What is SDLC**

Answer: SDLC is a software development life cycle of a step-by-step process

Requirements gathering, analysis, design, implementation, testing, maintenance and support.

## **2.What is software testing?**

Answer: Software testing is a process used to identify the correctness, completeness,

And quality of developed computer software.

## **3.What is agile methodology?**

Answer: agile SDLC model is a combination of iterative and incremental process models

With focus on process adaptability and customer satisfaction by rapid delivery of working software product.

## **4.What is SRS**

Answer: 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.

## **5.What is oops**

Answer: OOPS is an object-oriented programming system: Blackbox testing

## **6. write basic concepts of oops**

\* Object

\* Class

\* Encapsulation

\* Inheritance

\* Polymorphism

\* Abstraction

## **7.What is object**

Answer: Object is an instances of a class

## **8.What is class**

Answer: Class is a collection of data member (variables) and member function

(process, methods) with its behaviours.

## **9.What is encapsulation**

Answer: wrapping up of data into single unit

Private your data member and member function.

## **10.What is inheritance**

Answer: properties of parent class extend into child class main

purpose is reusability, extensibility.

## **11.What is polymorphism**

Answer: ability to take one name having many forms

There are mainly 2 types

1. Compile time (method overloading)

2. Run time (method overriding)

## **12.write SDLC phases with basic introduction**

Answer: SDLC PHASES:

\* Requirements gathering

\* Analysis

\* Design

\* Implementation

\* Testing

\* Maintenance

\* Requirements gathering = `customer needs user and business needs change during the project.

\* ANALYSIS = The analysis phase defines the requirement of the system, independent of

how these requirements will be accomplished.

\* DESIGN = The design team can now expand upon the information established in the

Requirements documents.

\* implementations = the end deliverable is the product its self-there are already many

Established techniques associated with implementation.

\* TESTING = Simply stated, quality is very important. it is much easier to explain to a

there is a missing feature than to explain to a customer why the product lacks quality.

\* MAINTENANCE = maintenance is the process of changing a system after it has been deployed.

## **13.Explain phases of the waterfall model**

Answer: phases of waterfall model:

\* Requirements gathering

\* Analysis

\* Design

\* Implementation

\* Testing

\* Maintenance

\* Requirements gathering = `customer needs user and business needs change during the project.

\* ANALYSIS = The analysis phase defines the requirement of the system, independent of

how these requirements will be accomplished.

\* DESIGN = The design team can now expand upon the information established in the

Requirements documents.

\* implementations = the end deliverable is the product its self-there are already many

Established techniques associated with implementation.

\* TESTING = Simply stated, quality is very important. it is much easier to explain to a

there is a missing feature than to explain to a customer why the product lacks quality.

\* MAINTENANCE = maintenance is the process of changing a system after it has been deployed.

## **14. write phases of spiral model**

Answer: phases of Spiral model

* Planning
* Risk analysis
* Engineering
* Customer evaluation

:**15. write agile manifesto principles**

Answer: Agile manifesto

\* INDIVIDUALS

\* WORKING SOFTWARE

\* CUSTOMER COLLABORATION

\* RESPONDING TO CHANGE

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

Answer: working methodology

Documentation is needed, but working software is much needed. agile is not saying that documentation is not needed, but working software is much needed.

PROS:

· Resource requirements are minimum

· Minimal rules, documentation easily employed.

· Suitable for fixed or changing requirements

· Delivers early partial working solutions

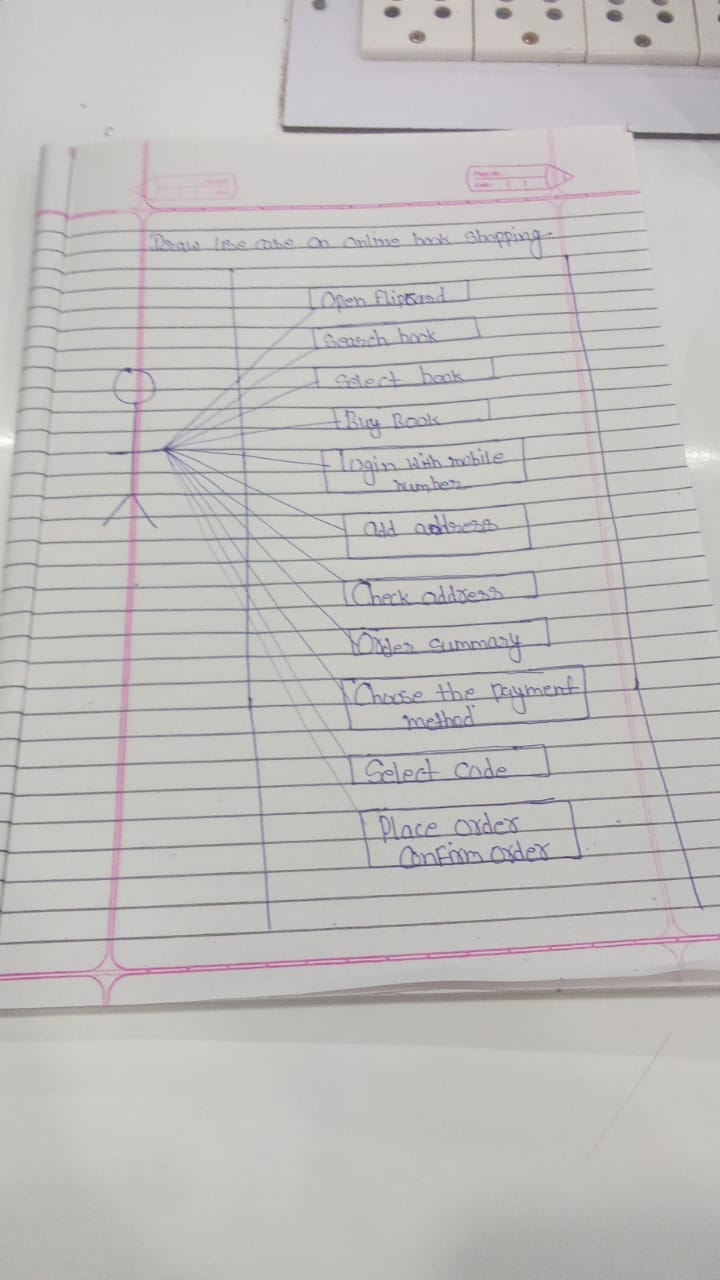
· Good model for environments that change steadily

CONS:

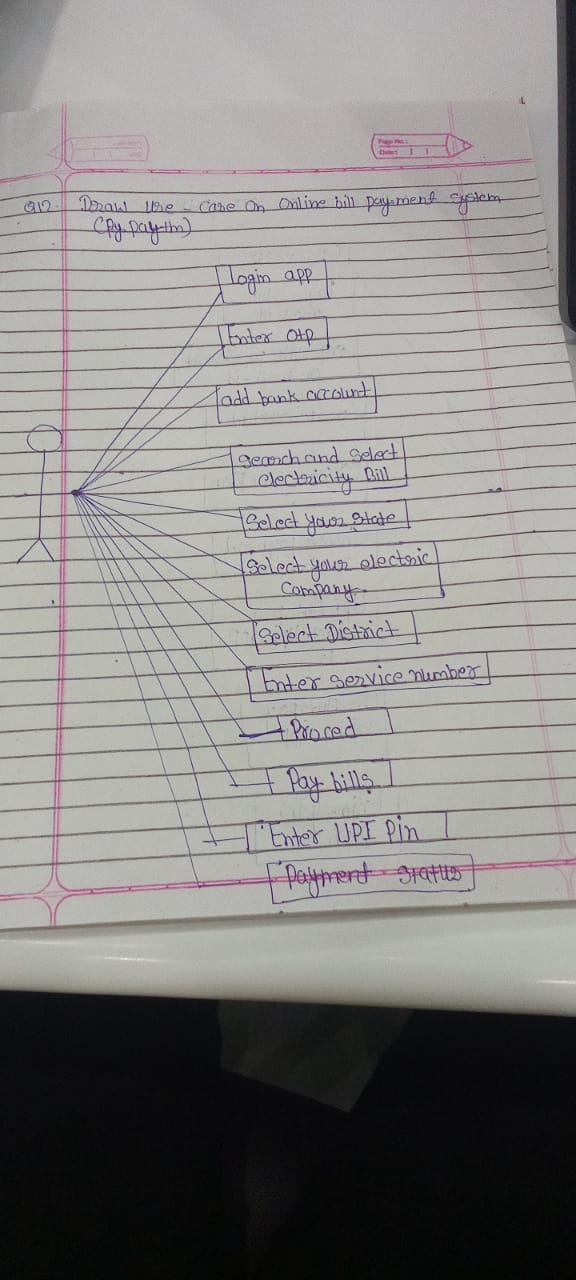
· Not suitable for handling complex dependencies

· Depends heavily

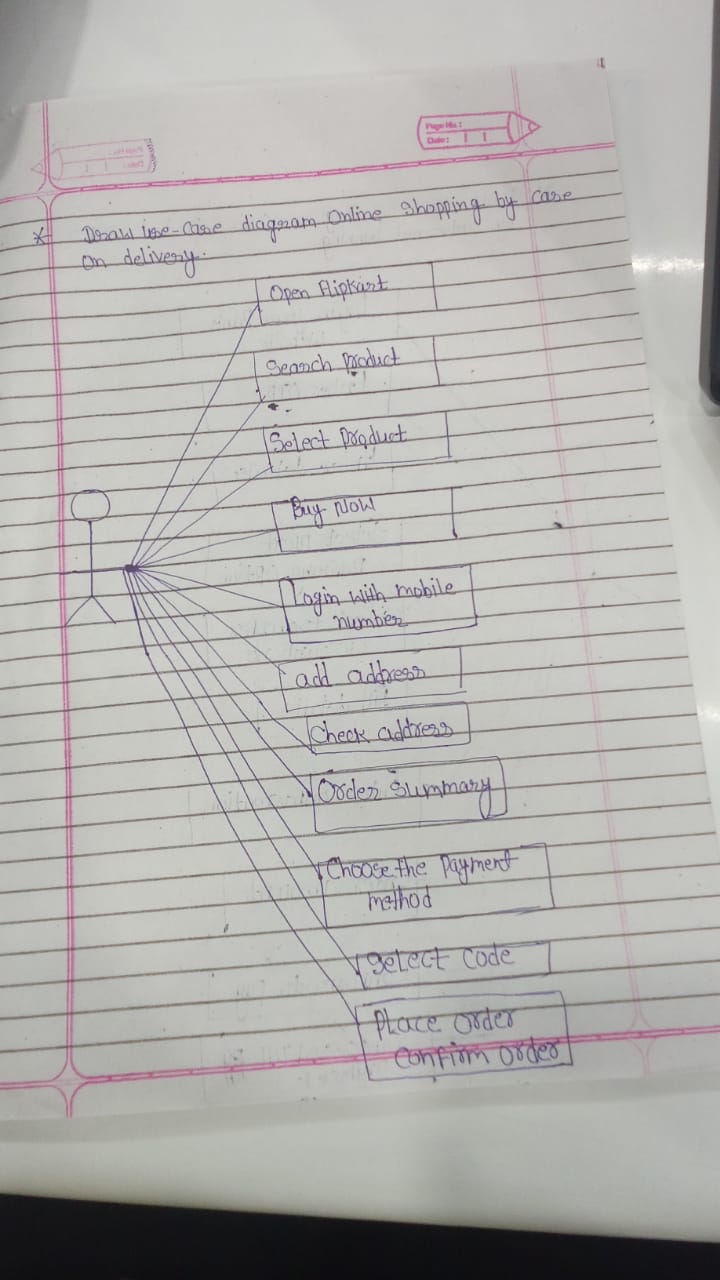
**17 . Draw Usecase on Online book shopping**

****

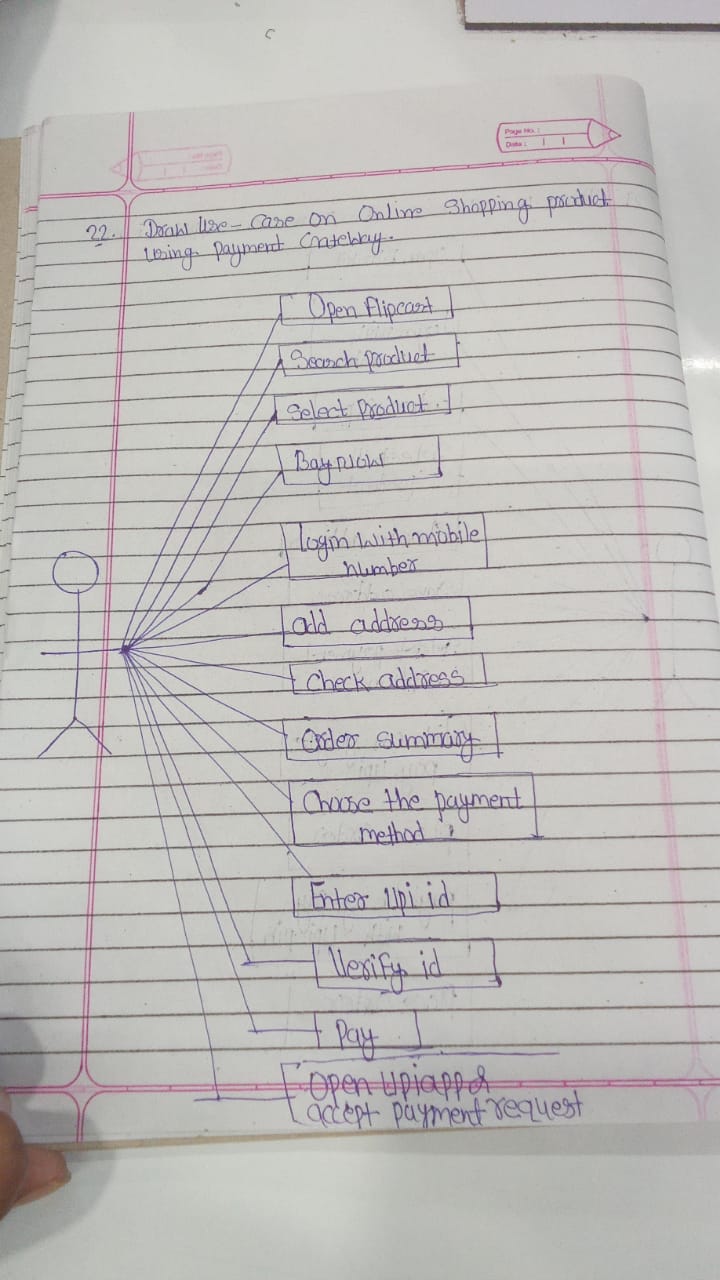
**18. Draw Usecase on online bill payment system (paytm)**



**18. Draw use case on Online shopping products using COD.**

****

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



## 

## 