Assignment

Module-1

Q1.What is SDLC?

Ans. SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support.

- A Software Development Life Cycle is essentially a series of steps, or phases.

Q2. What is software testing?
Ans. Software Testing is a process used to identify the correctness, completeness, and quality of developed computer software.

- ➢ Mainly there are two types of Testing are there:-
  - Static Testing:- It can test and find defects without executing code. StaticTesting is done during verification process. For example: reviewing, walkthrough, inspection, etc.
  - Dynamic Testing: In dynamic testing the software code is executed to demonstrate the result of running tests. It's done during validation process.For example: unit testing, integration testing, system testing, etc.

Q3. Write SDLC phases with basic introduction?
Ans. SDLC is a structure imposed on the development of a software product that defines the process for planning, implementation, testing, documentation, deployment, and ongoing maintenance and support.

- ➢ SDLC Phases are given below:-

  1.Requirment Gathering:- Establish Customer Needs
  2.Analysis:- Model And Specify the requirements-"What"
  3.Design:- Model And Specify a Solution – "Why"
  4.Implementation:- Construct a Solution In Software
  5.Testing:- Validate the solution against the requirements
  6.Maintenance:- Repair defects and adapt the solution to the new requirements.

Q4. Explain Phases of the waterfall model?
Ans. The classical software lifecycle models the software development as a step-by-step "Waterfall" between the various development phases.

- ➢ The main phases of waterfall are given below:-
  - Requirment collection
  - Analysis
  - Design
  - Implementation
  - Testing
  - Maintenance
- ➢ The waterfall is unrealistic for many reasons,especially:-
- Requirements must be "**frozen**" to early in the life cycle
- Requirements are **validated too late**

Q5. What is SRS?
Ans. A software requirements specification (SRS) is a complete description of the behavior of the system to be developed.

- ➢ Requirements are categorized in several ways:-
  - Customer Requirements
  - Functional Requirements
  - Non-Functional Requirements

Assignment

Module-1

Q6. What is oops ?
Ans. OOPs Stands for object oriented programming languages. OOPS Stands for Object Oriented Programming system in which programs are considered as a collection of objects. Each object is nothing but an instance of a class.

Q7. Write Basic Concepts of oops?
Ans. The Basic concepts of OOPS are given below:-

- Object
- Class
- Encapsulation
- Inheritance
- Polymorphism
    I. Overloading
    II. Overriding
- Abstraction

Q8.What is object?
Ans. An object represents an individual,identifiable item, unit, or entity,either real or abstract, with a well-defined role in the problem domain.That is both data and function that operate on data are bundled as a unit called as object.For ex.

Q9.What is Class?
Ans. A class represents an abstraction of the object and abstracts the properties and behavior of that object. When you define a class, you define a blueprint for an object. An object is a particular instance of a class which has actual existence and there can be many objects (or instances) for a class.

Q10.What is Encapsulation?
Ans. Encapsulation is the practice of including in an object everything it needs hidden from other objects. The internal state is usually not accessible by other objects.

- Encapsulation enables data hiding**,** hiding irrelevant information from the users of a class and exposing only the relevant details required by the user.

Q11. What is inheritance?
Ans**.** Inheritance means that one class inherits the characteristics of another class. This is also called a “is a” relationship.

- Inheritance is the process of forming a new class from an existing class that is from the existing class called as base class, new class is formed called as derived class.
- This feature helps to reduce the code size.

Q12. What is Polymorphism?
Ans. Polymorphism means “having many forms”.It allows different objects to respond to the same message in different ways, the response specific to the type of the object.

- There is two types of polymorphism:-
  1.Compile Time(overloading)
  2.Run Time(overriding)

Assignment

Module-1

Q13. Write phases of spiral model?
Ans.The Phases of spiral model is given below:-

- Planning:-Determinations of objectives,alternatives and constraints.
- Risk Analysis:-Analysis of alternatives and identification/resolution of risks.
- Engineering:-Development of the “next level” product.
- Customer Evalutaion:-Assesment of the result of engineering

Q14.What is agile methodology?
Ans. 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.

- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis,design, coding, unit testing, and acceptance testing.

Q15.Write agile manifesto principles?
Ans.Mainly the principles are given Below:-

1. Individuals and interactions
2. Working software
3. Customer collaboration
4. Responding to change

Q16. Explain working methodology of agile model and also write pros and cons?
Ans. 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.

- Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis,design, coding, unit testing, and acceptance testing.

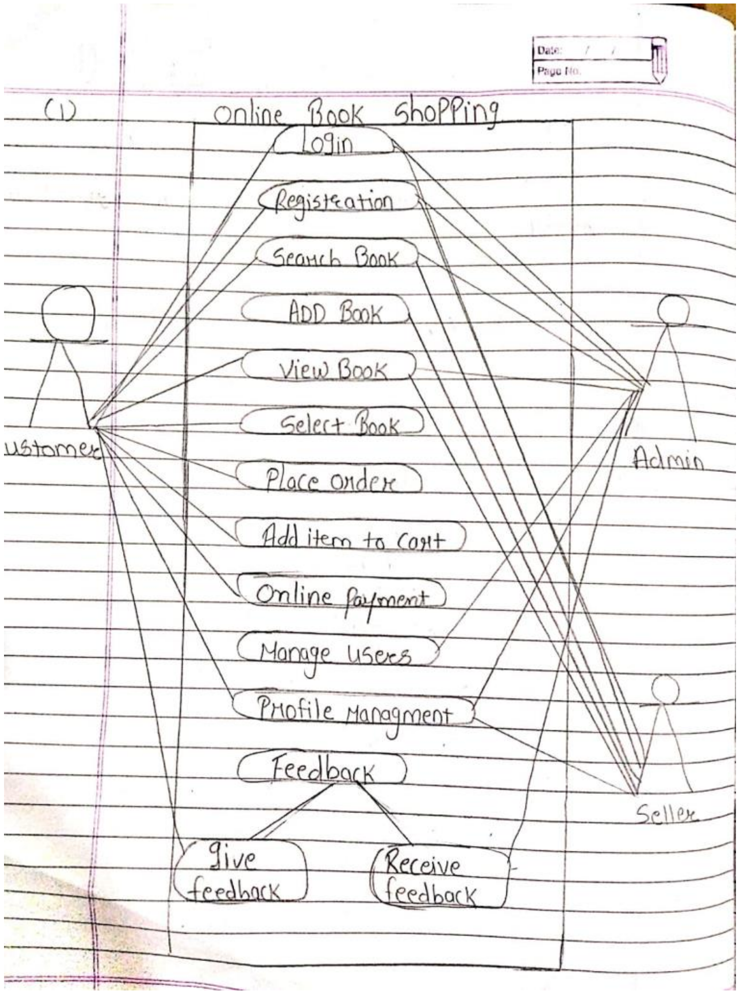
The Pros and Cons are given below:-

- **Pros**
  - is a very realistic approach to software development
  - Promotes teamwork and cross training.
  - Functionality can be developed rapidly and demonstrated.
  - Resource requirements are minimum.
  - Suitable for fixed or changing requirements
  - Delivers early partial working solutions.
  - Good model for environments that change steadily.
  - Minimal rules, documentation easily employed.
  - Enables concurrent development and delivery within an overall planned context.
  - Little or no planning required
  - Easy to manage
  - Gives flexibility to developers

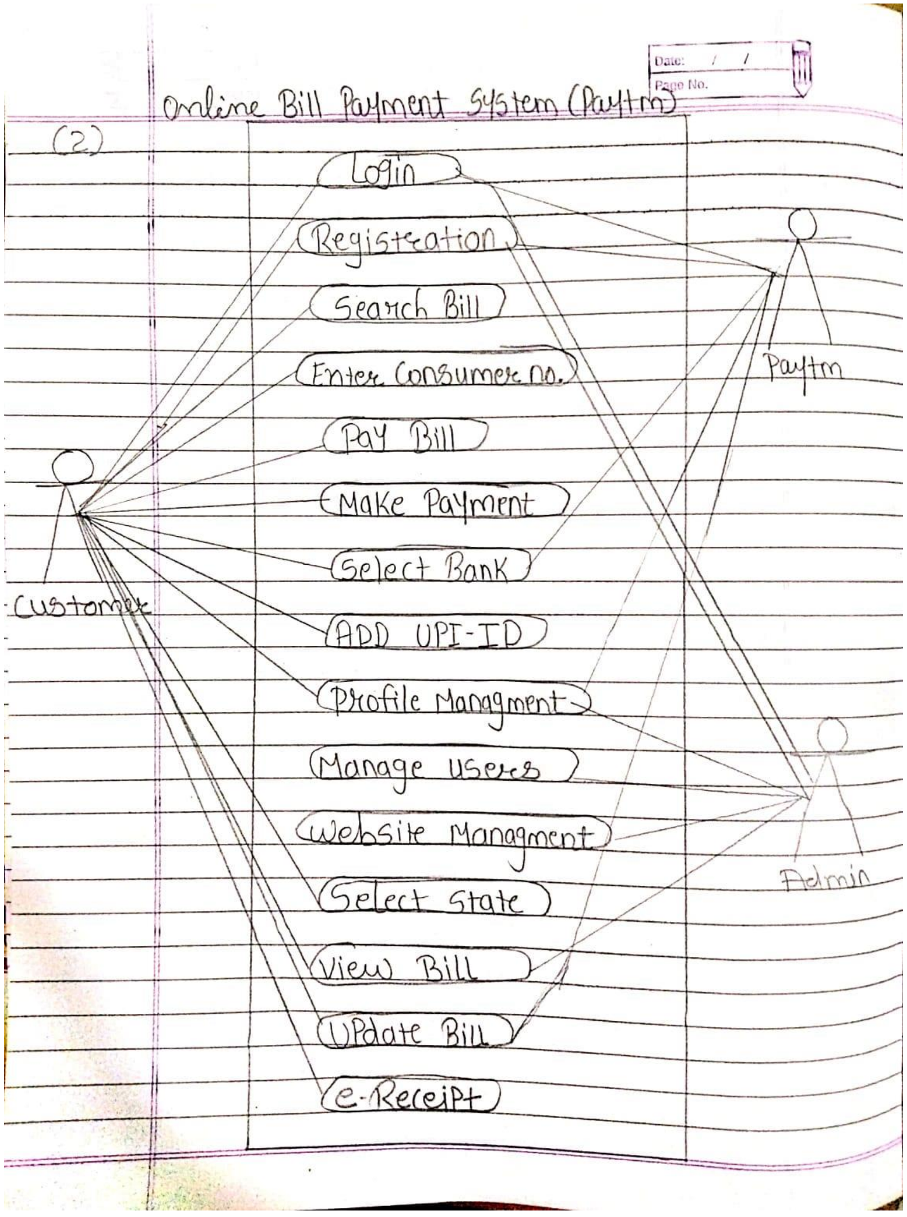
- **Cons**
  - Not suitable for handling complex dependencies.
  - More risk of sustainability, maintainability and extensibility.
  - An overall plan, an agile leader and agile PM practice is a must without which it will not work.
  - Strict delivery management dictates the scope, functionality to be 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.

- Transfer of technology to new team members may be quite challenging due to lack of documentation.

Q17.Draw Usecase on Online book shopping?



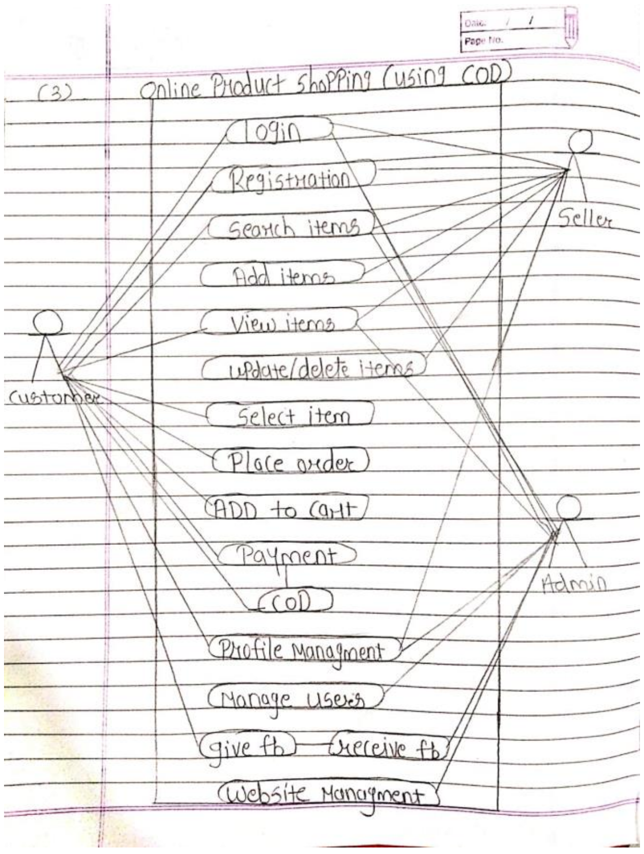
Q18.Draw Usecase on online bill payment system (paytm)?



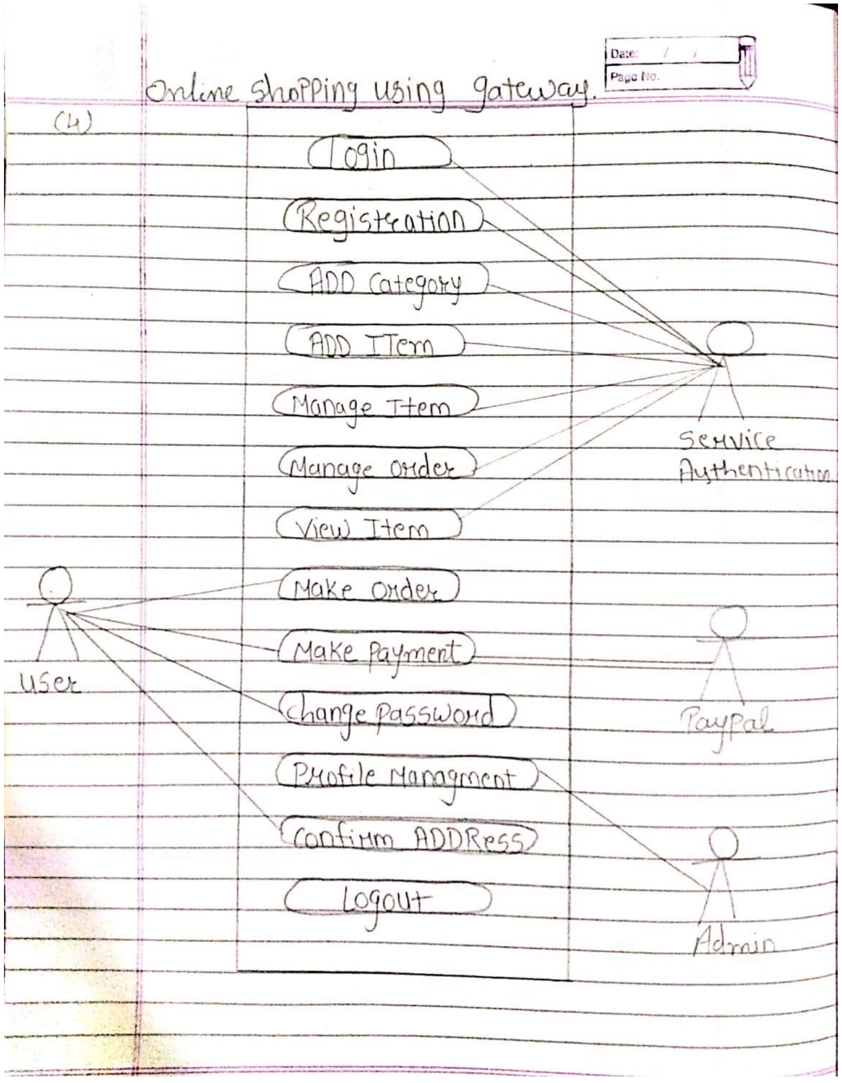
Assignment

Module-1

Q19.Draw usecase on Online shopping product using COD?



Q20.Draw usecase on Online shopping product using payment gateway?


Date: / /
Page No.
Online shopping using gateway.
(4)
Login
Registration
ADD Category
ADD ITem
Manage ITem
Manage Order
View Item
Make Order
Make Payment
Change Password
Profile Managment
Confirm ADDRess
Logout
Service Authentication
User
PayPal
Admin