Software Testing Assignment

Module–1(Fundamental)

1. What is SDLC

   Software Development Life Cycle (SDLC) is a methodology or step by step approach to produce software with high quality ,lowest cost in the shortest time period by defining the phases like Planning, Analysis & Design, Coding & Implementation and Testing and Maintenance.

2. What is software testing?

   Software testing is a process used to identify correctness, completeness and quality of a developed working software.

3. What is agile methodology?

   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

   A software requirements specification (SRS) is a complete description
   of the behaviour of the system to be developed.

5. What is oops

   Object Oriented Programming (OOP) is viewed as collection of objects. It is used to structure the software program into simple reusable code , referred as functional or blackbox testing.

6. Write Basic Concepts of oops

   1. Object
   2. Class
   3. Encapsulation
   4. Inheritance
   5. Polymorphism
   6. Abstraction

7. What is object

   Object is basic unit of OOP which is accessed by its properties called data member and member functionality.

8. What is class

   Class is a collection of a data member and member function with its behaviour.

9. What is encapsulation

   Wrapping up of data and functions into a single unit is called Encapsulation. It hides /include private access of data member and member function.

10. What is inheritance

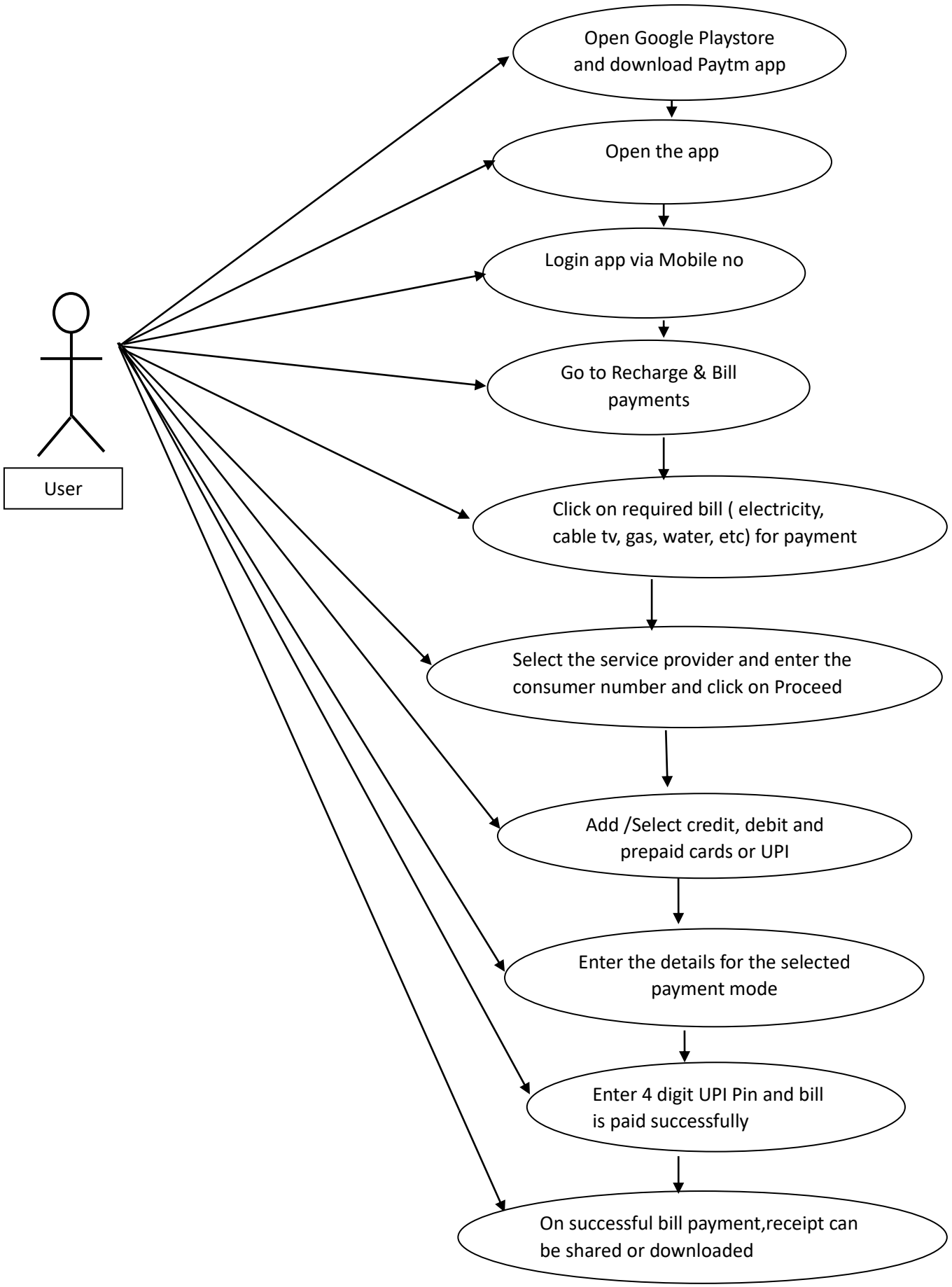
    Inheritance means that one class (super, base) inherits the properties of another class (sub, derived).

11. What is polymorphism

    Polymorphism is an ability to take one name having many different forms.

12. Draw Usecase on Online book shopping

13. Draw Usecase on online bill payment system (paytm)



14. Write SDLC phases with basic introduction

    There are mainly six phases in SDLC.

    1. **Requirement Gathering**: This phase involves gathering information about the software requirements from stakeholders, such as customers, end-users, and business analysts.
    2. **Analysis** : This phase starts with the requirement document delivered by the requirement phase and maps the requirements into architecture.The architecture defines the components, their interfaces and behaviors.
    3. **Design** : In this phase, the software design is created, which includes the overall architecture of the software, data structures, and interfaces. 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.
    4. **Coding & Implementation** : Given the architecture document from the design phase and the requirement document from the analysis phase, the team should build exactly what has been requested.The implementation phase deals with issues of quality, performance, baselines, libraries, and debugging.
    5. **Testing** : The software is thoroughly tested to ensure that it meets the requirements and works correctly.
    6. **Maintenance** : This phase includes ongoing support, bug fixes, and updates to the software.

15. Explain Phases of the waterfall model

    Waterfall model is a linear and sequential approach to software development that consists of several phases that must be completed in a specific order. he phases include:

1. **Requirements Gathering :** The first phase involves gathering requirements from customers or stakeholders for the project.
2. **Analysis :** On the basis of requirement document, analyze them to understand the scope and objectives of the project.
3. **Design :** This involves creating a detailed design document that outlines the software architecture, user interface, and system components.
4. **Implementation:** The implementation phase involves coding the software based on the design specifications. This phase also includes unit testing to ensure that each component of the software is working as expected.
5. **Testing:** In the testing phase, the software is tested as a whole to ensure that it meets the requirements and is free from defects.
6. **Maintenance:** The final phase of the Waterfall Model is maintenance, which involves fixing any issues that arise after the software has been deployed and ensuring that it continues to meet the requirements over time.

16. Write phases of spiral model

    1. Planning
    2. Risk Analysis
    3. Engineering
    4. Evaluation

17. Write agile manifesto principles

    1. Individuals and interactions over processes and tools.
    2. Working software over comprehensive documentation
    3. Customer collaboration over contract negotiation
    4. Responding to change over following a plan

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

    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.

    - Agile Methods break the product into small incremental builds.
    - These builds are provided in iterations.
    - Each iteration typically lasts from about one to three weeks.
    - Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis,design, coding, unit testing, and acceptance testing.
    - At the end of the iteration a working product is displayed to the customer and important stakeholders.

    Pros :

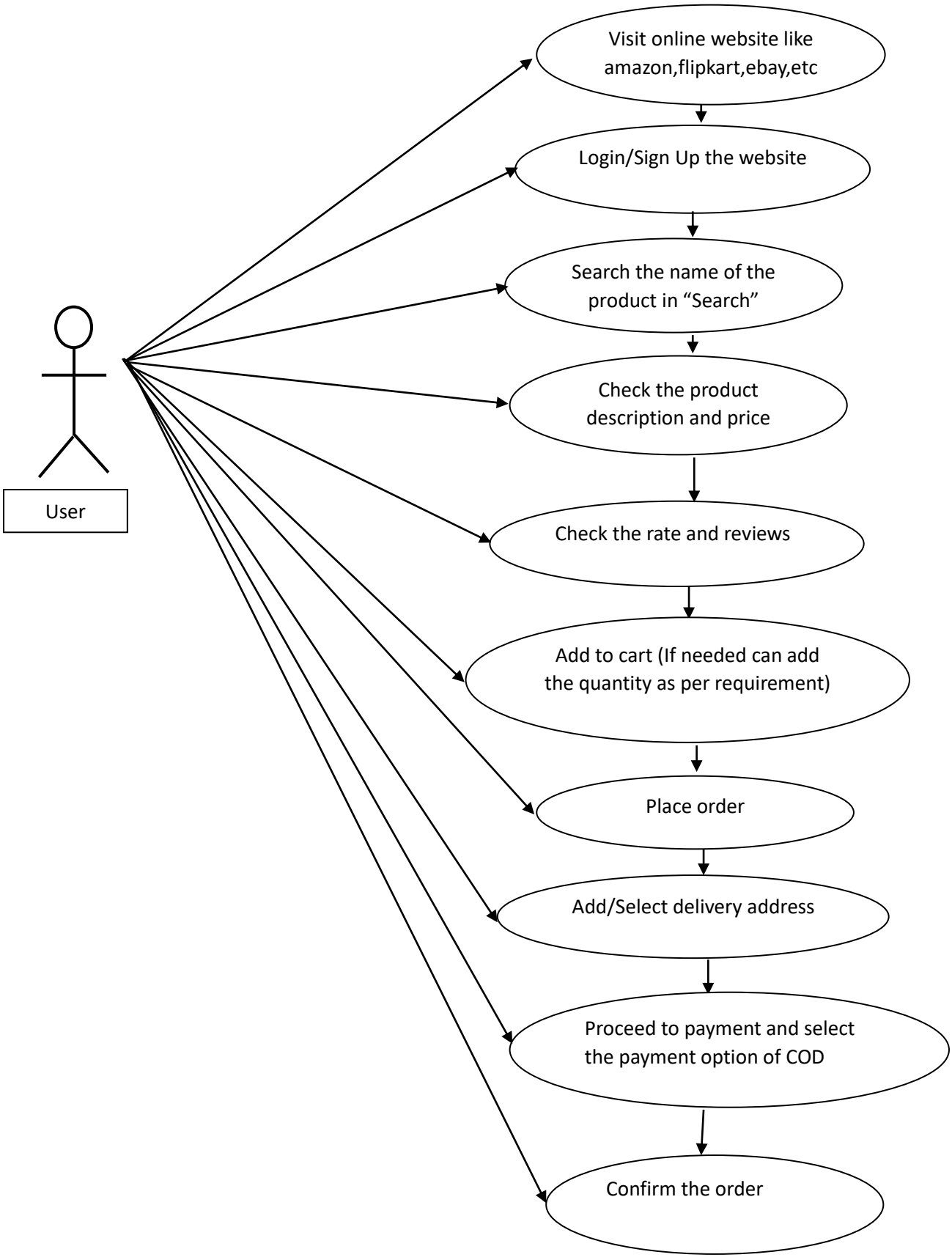
    1. Is a very realistic approach to software development.
    2. Functionality can be developed rapidly and demonstrated.
    3. Resource requirements are minimum.
    4. Suitable for fixed or changing requirements
    5. Delivers early partial working solutions.
    6. Good model for environments that change steadily.
    7. Minimal rules, documentation easily employed.
    8. Enables concurrent development and delivery within an overall planned context.
    9. Little or no planning required
    10. Easy to manage
    11. Gives flexibility to developers

    Cons:

    1. Not suitable for handling complex dependencies.

2. More risk of sustainability, maintainability and extensibility.
3. An overall plan, an agile leader and agile PM practice is a must without which it will not work.
4. Strict delivery management dictates the scope, functionality to be delivered, and adjustments to meet the deadlines.
5. Depends heavily on customer interaction, so if customer is not clear, team can be driven in the wrong direction.
6. There is very high individual dependency, since there is minimum documentation generated.
7. Transfer of technology to new team members may be quite challenging due to lack of documentation.

19. Draw usecase on Online shopping product using COD.



20. Draw usecase on Online shopping product using payment gateway