Software Testing Assignment

Module 1

Q1. What is SDLC?

- SDLC is the abbreviation for Software Development Life Cycle. SDLC is the structure that shows the process of planning, design, implementation, testing, and maintenance of a software's development.
- There are a number of development models from which the development team can choose to implement based on the type of software being developed.

Q2. What is software testing?

- Software Testing the process of evaluating a software or the components of a software to check whether it/they fulfil the needs for which it/they have been developed while maintaining a modicum of quality and standard.

Q3. What is agile methodology?

- Agile methodology is the most popular model used in modern software industry.
- Agile SDLC model is a combination of iterative and incremental models with focus on process adaptability and customer satisfaction by rapid delivery of the product.
- Using iterative model, an iteration of the software is created first and is displayed to the customer and other stakeholders. The later iterations contain the other features that keep adding to the first iteration and thus, a final product is created in the end.

Q4. What is SRS?

- SRS or Software Requirement Specification is the document produced by a group of senior engineers regarding the development of a product/project that contains all the details for the software's scope, standards, quality, and any functional or non-functional requirements.
- It is the complete description of the behaviour of the software that is to be developed.
- SRS will have all the instances of the use cases that the user will have with the software.

Q5. What is OOPS?

- OOPS refers to the Object Oriented Programming. OOPS is a way or method of writing code using the concept of **objects and classes**.
- It is based on four main principles: **Encapsulation, Inheritance, Polymorphism, and** Abstraction.

- This method of coding helps the code to stay structured, reuseable, and easier to maintain.

Q6. Write Basic Concepts of oops.

- The basic concepts of OOPS are as follows:
  1. **Class** - Blueprint/template for creating objects
  2. **Object** - Instance of a class with its data and behaviour.
  3. **Encapsulation** - Wrapping data together.
  4. **Inheritance** - One class reusing another class's features/methods.
  5. Polymorphism – One face, many forms
  6. **Abstraction** - only showing necessary details while keeping other data/details hidden

Q7. What is object?

- An object is an instance of a class representing a real-world entity.
- It contains actual values/data and can use the methods defined in its class.

Q8. What is class?

- A class is a blueprint used to create objects.
- It defines the properties (variables) and behaviors (methods) that the objects will have.

Q9. What is encapsulation?

- Encapsulation means bundling data and methods together in one unit and restricting direct access to the data.
- This protects data and improves security and maintainability.

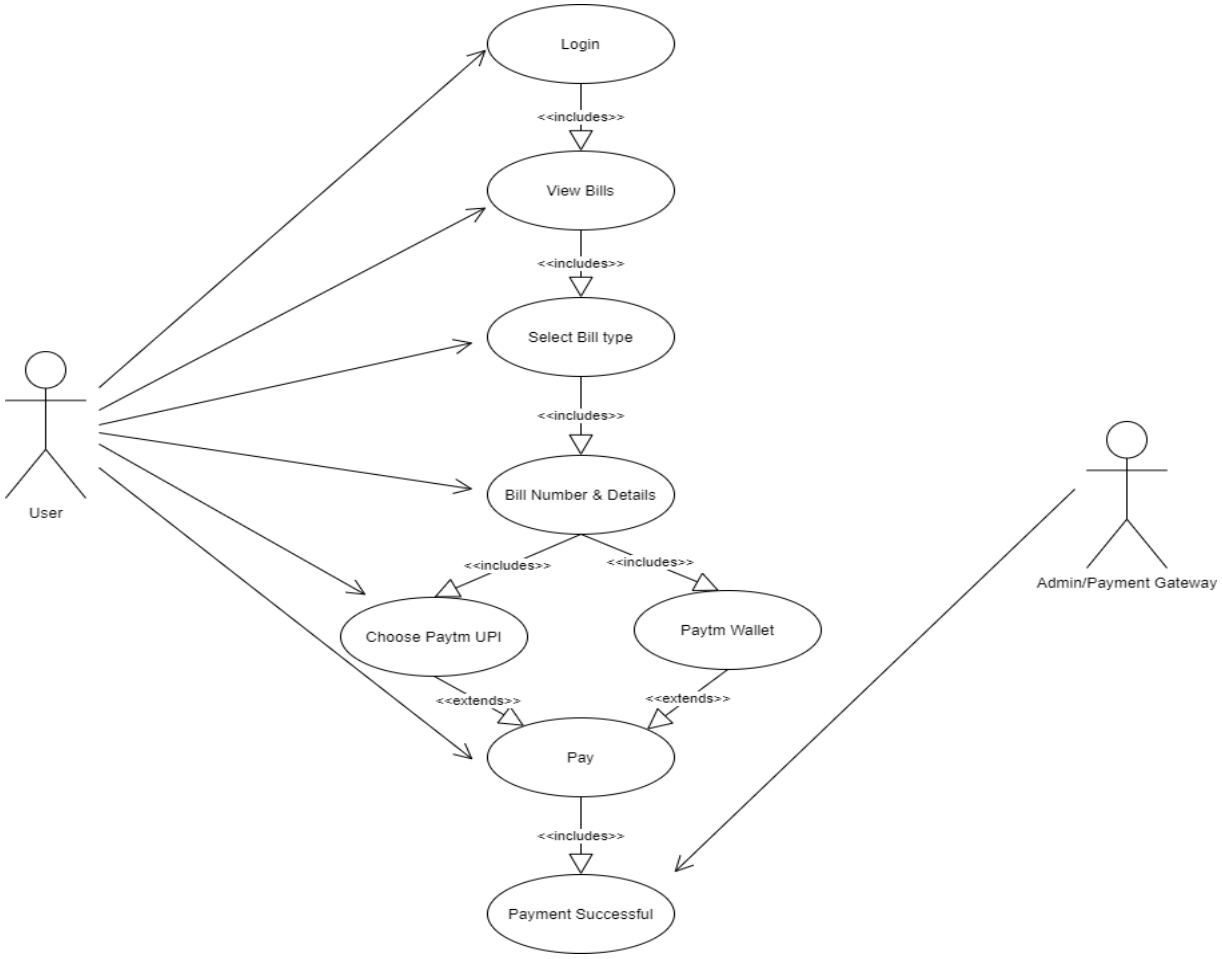
Q10. What is inheritance?

- Inheritance allows one class to acquire properties and methods of another class.
- This promotes code reuse and establishes relationships between classes.

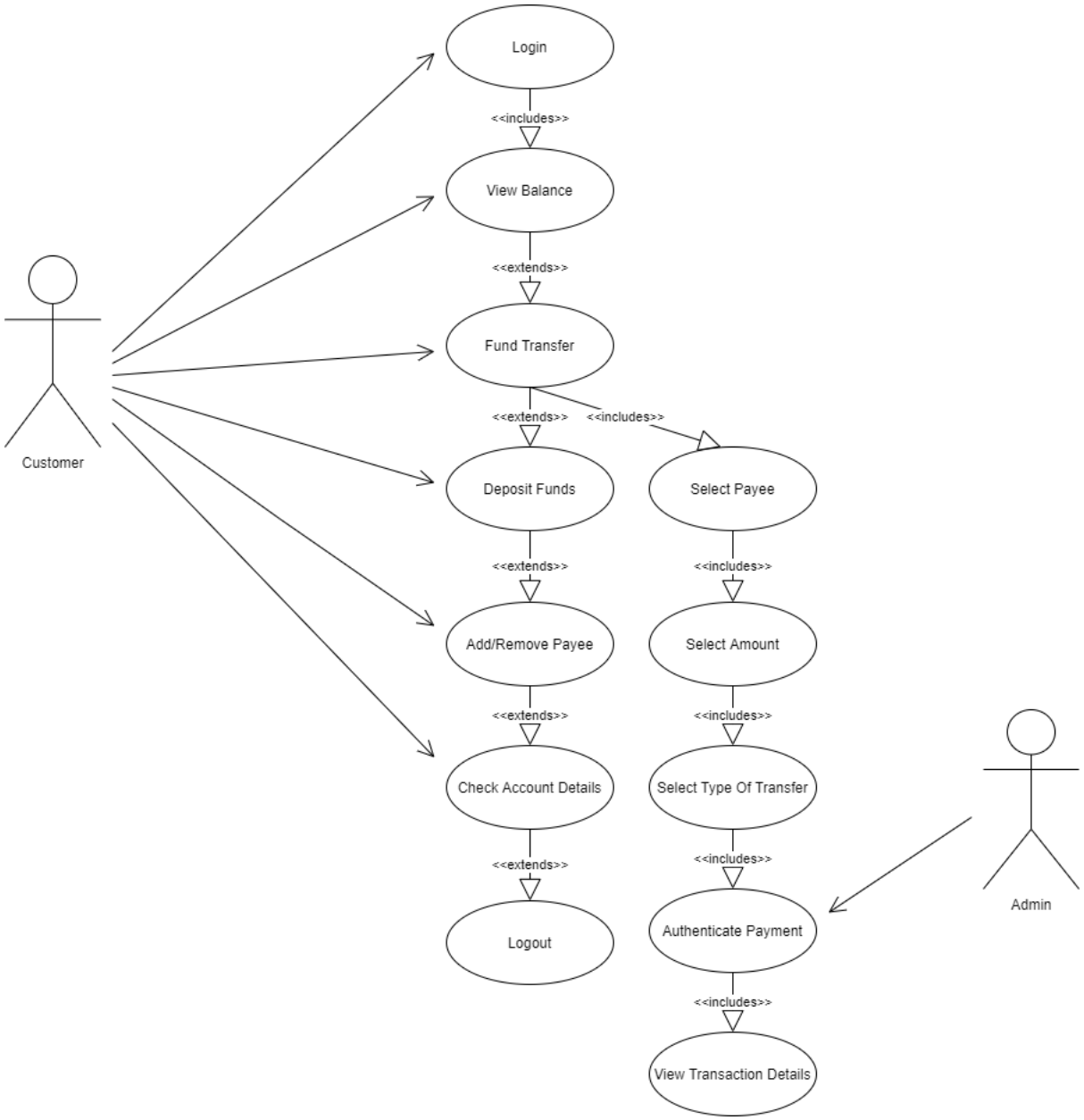
Q11. What is polymorphism?

- Polymorphism allows the same method or interface to behave differently in different situations.
- It increases flexibility and extensibility in programs.

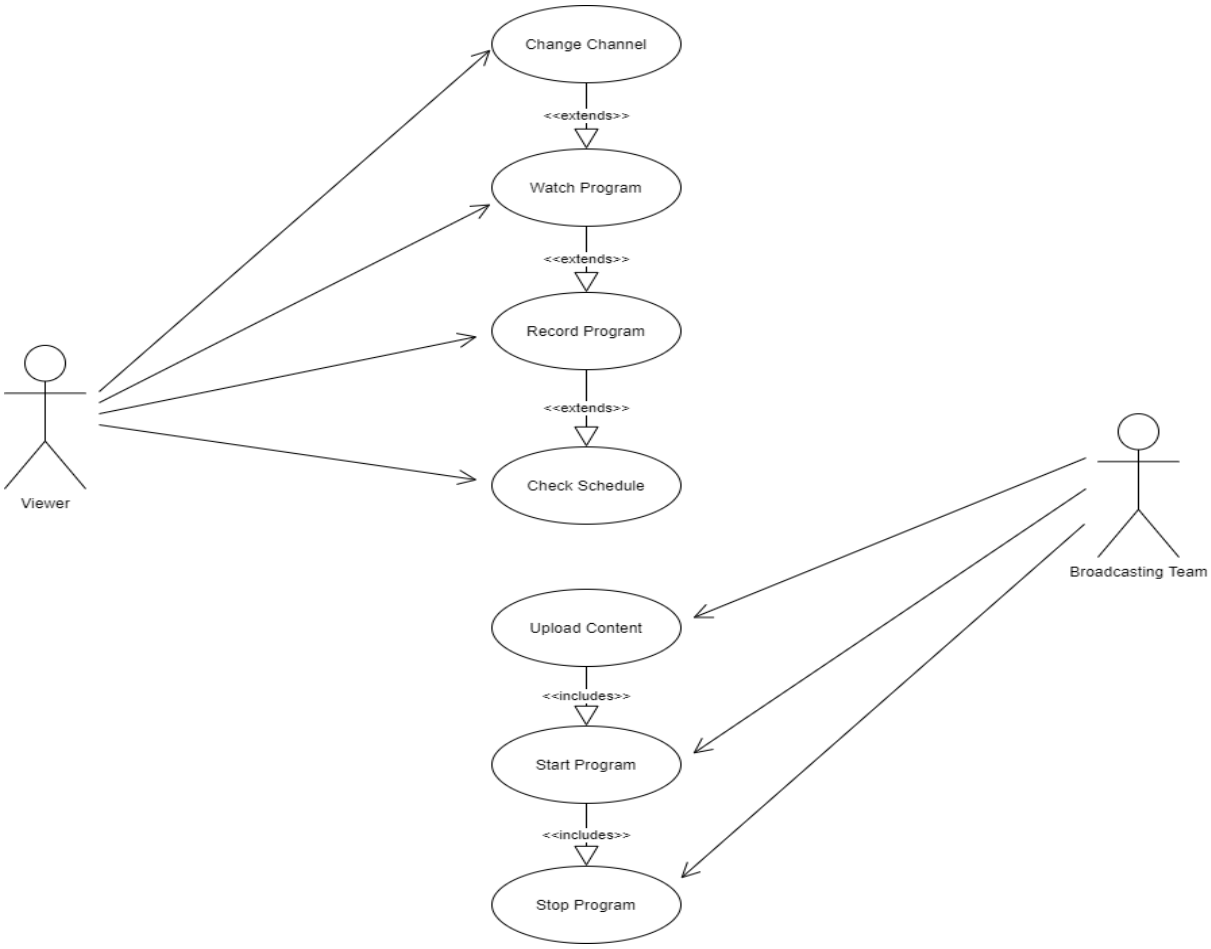
Q12. Draw Usecase on online bill payment system (paytm)



Q13. Draw Usecase on banking system for customers.



Q14. Draw Usecase on Broadcasting System.



Q15. Write SDLC phases with basic introduction.

- SDLC has seven phases: Requirement Collection, Analysis, Design, Implementation, Testing, Deployment, and Maintenance.
- **1. Requirement Collection Phase:** This phase establishes the customer's needs and requirements. The CRS document is created in this phase.
- **2. Analysis Phase:** The created CRS is then thoroughly analysed and the requirements are understood. An SRS is created at the end of this phase for the next phases.
- **3. Design Phase:** In the design phase, the system architechture, UI, database structures, and other technicalities are established so that developers can follow that development model.
- 4. **Implementation Phase:** This phase is where the developers develop the logic and coding for the software to be built.
- **5. Testing phase:** In testing phase, the testing of the developed software is done so as to check whether there are any defects, and whether the requirements have been met or not.

- **6. Deployment:** In this phase, the software is deployed into the market and monitored live by the devs and testers to check its working in the external environment.
- **7. Maintenance:** In this phase, the product is checked for any defects and then corrected, or it is adapted into new versions for new environments, and perfected by adding new features

Q16. Explain phases of the Waterfall Model?

- Waterfall model is one of the simplest models in its structure. It has the same phases of the SDLC: Requirement Gathering and Analysis, Design, Implementation, Testing, and Maintenance.
- In this model, the requirements are frozen because of which the SDLC can only move forward to the next phases.
- Only in the later phases will any requirements be validated and which is why, only small projects are preferred for this model.

Q17. Write phases of Spiral Model.

- Boehm's Spiral Model is widely used when there are budget constraints and the risk evaluation is important.
- The phases in this model are:
  1. **Planning:** In this phase, the objectives, constraints and any alternative solutions are determined.
  2. **Risk Analysis:** Risks like delays, technical issues, and budget issues are identified and evaluated. Such risks are mitigated by conducting simulations, and building prototypes.
  3. **Engineering:** In this phase, software is developed and the product is tested. This phase has includes the design, coding, and testing.
  4. **Customer Evaluation:** The customer/client evaluates the resulted product created in the previous phase. Based on the product's evaluation, the next iterations are planned.
- The process's phases are iterated multiple times until the customer's needs and requirements are fulfilled. This methodology ensures the least risks involved.

Q18. Write agile manifesto principles.

- **Customer satisfaction** - Deliver working software early and periodically.
- **Accept changes** - Adapt requirements anytime in the development process.
- **Frequent delivery** - Release working software regularly in each iteration.
- **Collaboration** - Business team and developers work together and fill in the data with the stakeholders as well.
- **Trust & motivation** - Support the team to do their best. Motivated team always delivers quality work on time.

- **Face-to-face communication** - Direct talk works best to have no misinterpretations of requirements and any other data.
- **Working software implies progress** - Functionality matters, not paperwork. A functioning software is shows the primary progress.
- **Consistent pace** - Work at a steady, sustainable speed to gain development on attainable deadlines.
- **Good design & quality** - Keeps development flexible for evolving/changing requirements.
- **Simplicity** - Build only what's needed primarily. New features and non-functional requirements can be built on later iterations.
- **Self-organizing teams** - Teams decide and take ownership.
- **Regular reflection for teams to become more effective** – On top of reviewing and improving continuously, individuals should upskill and improve their processes.

Q19. Explain working methodology of agile model and also write pros and cons.

- Agile SDLC model is a combination of iterative and incremental models with focus on process adaptability and customer satisfaction by rapid delivery of the product.
- In Agile method, the product is broken into small components and is provided to the client/customer in parts.
- After end of the each iteration, a working product is released and displayed to customer.

Pros:

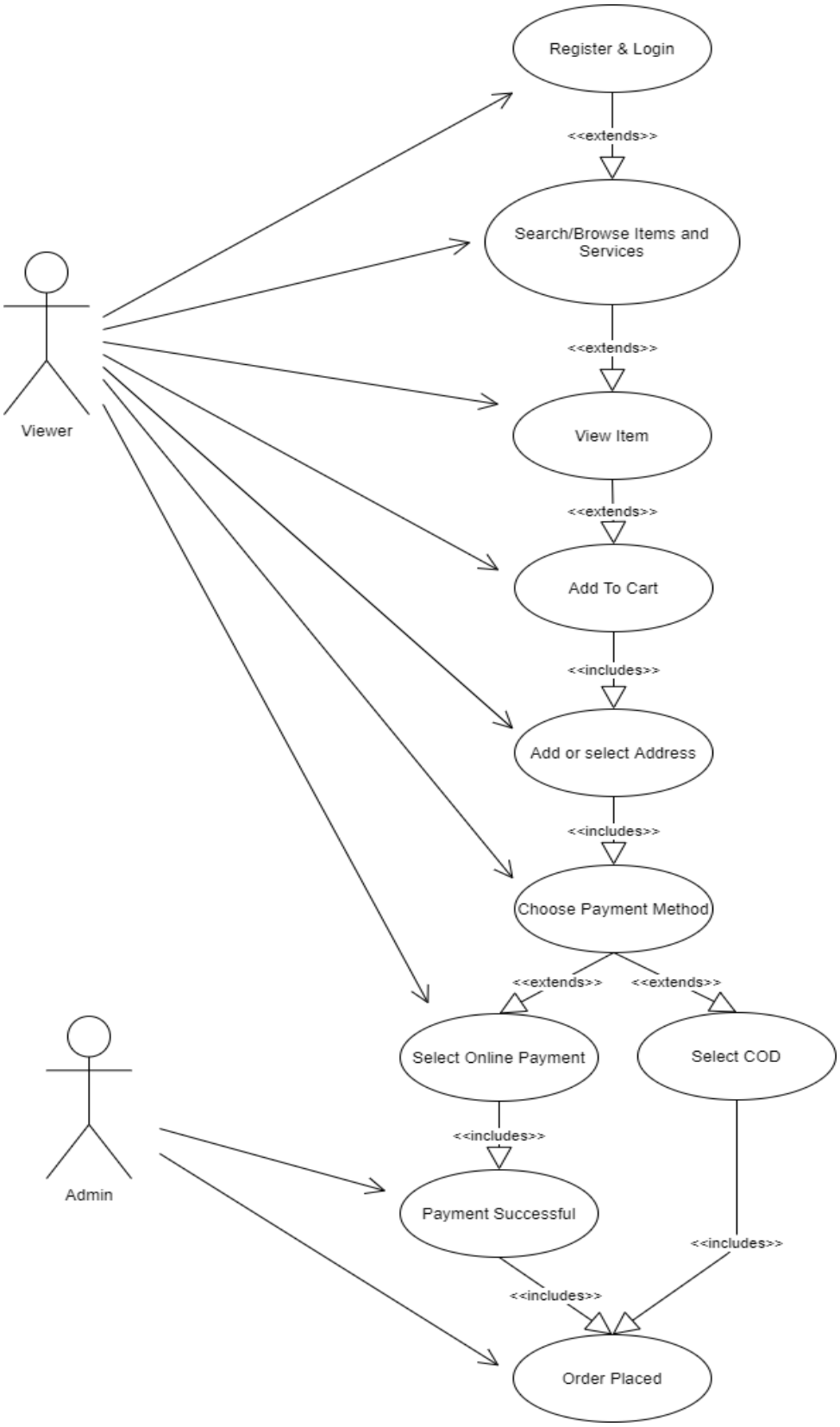
1. Functionality can be rapidly developed and demonstrated to the customer.
2. A flexible model for the environments that change steadily.
3. This model has less rules and documentation is easily employed.
4. This method gives flexibility to developers.

Cons:

1. Since requirements keep changing, it's difficult to predict deadlines and budget.
2. Needs constant involvement from developers, testers, management, stakeholders, etc, which is an issue.
3. Project might keep expanding for no reason because requirements are flexible.

Q20. Draw usecase on OTT Platform.

Q21. Draw usecase on E-commerce application



Q22. Draw usecase on Online shopping product using payment gateway.