Name: Devangi Dave
City: Nadiad

Module 1 (Fundamental)

1.What is SDLC?
Ans.: SDLC is the Software development life cycle. SDLC is the cost-effective and time-efficient process that development teams use to design and build high-quality software. The goal of SDLC is to minimize project risks through forward planning so that software meets customer expectations during production.

2.What is software testing?
Ans.: Software testing is the process of evaluating and verifying that a software product or application does what it is supposed to do. The benefits of testing include preventing bugs, reducing development costs, and improving performance.

3.What is Agile methodology?
Ans.: The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders' and continuous improvement at every stage. Once the work begins, teams' cycle through a process of planning, executing, and evaluating.

4.What is SRS?
Ans.: A software requirements specification (SRS) is a complete description of the behavior of the system to be developed. It also describes the functionality the product needs to fulfill the needs of all business & users.

5.What is Oop?
Ans.: Oop meaning is an Object-Oriented Programming. OOP is a computer programming model that organizes software design around data, or objects, rather than functions and logic. An object can be defined as a data field that has unique attributes and behavior.

6.Write Basic concepts of Oop.
Ans.:

- Classes/Objects
- Encapsulation/Data Hiding
- Inheritance
- Polymorphism
- Interface/Methods

7.What is object?
Ans.: Objects are the things you think about first in designing a program and they are also the units of code that are eventually derived from the process. That is both data & function that operate on data are bundled as a unit called as object.

8.What is Class?
Ans.: Class is blueprint for an object. This does not actually define any data, but it does define what the class name means, that is, what an object of the class will consist of and what operations can be performed on such an object. A class represents an abstraction of the object and abstracts the properties and behavior of that object. Class can be considered as the blueprint or definition or a template for an object and describes the properties and behavior of that object, but without any actual existence.

9.What is encapsulation?
Ans.: Encapsulation is the practice of including in an object everything it need hidden from other object. The internal state is usually not accessible by other object. Encapsulation is placing the data and the functions that work on that data on the same place.

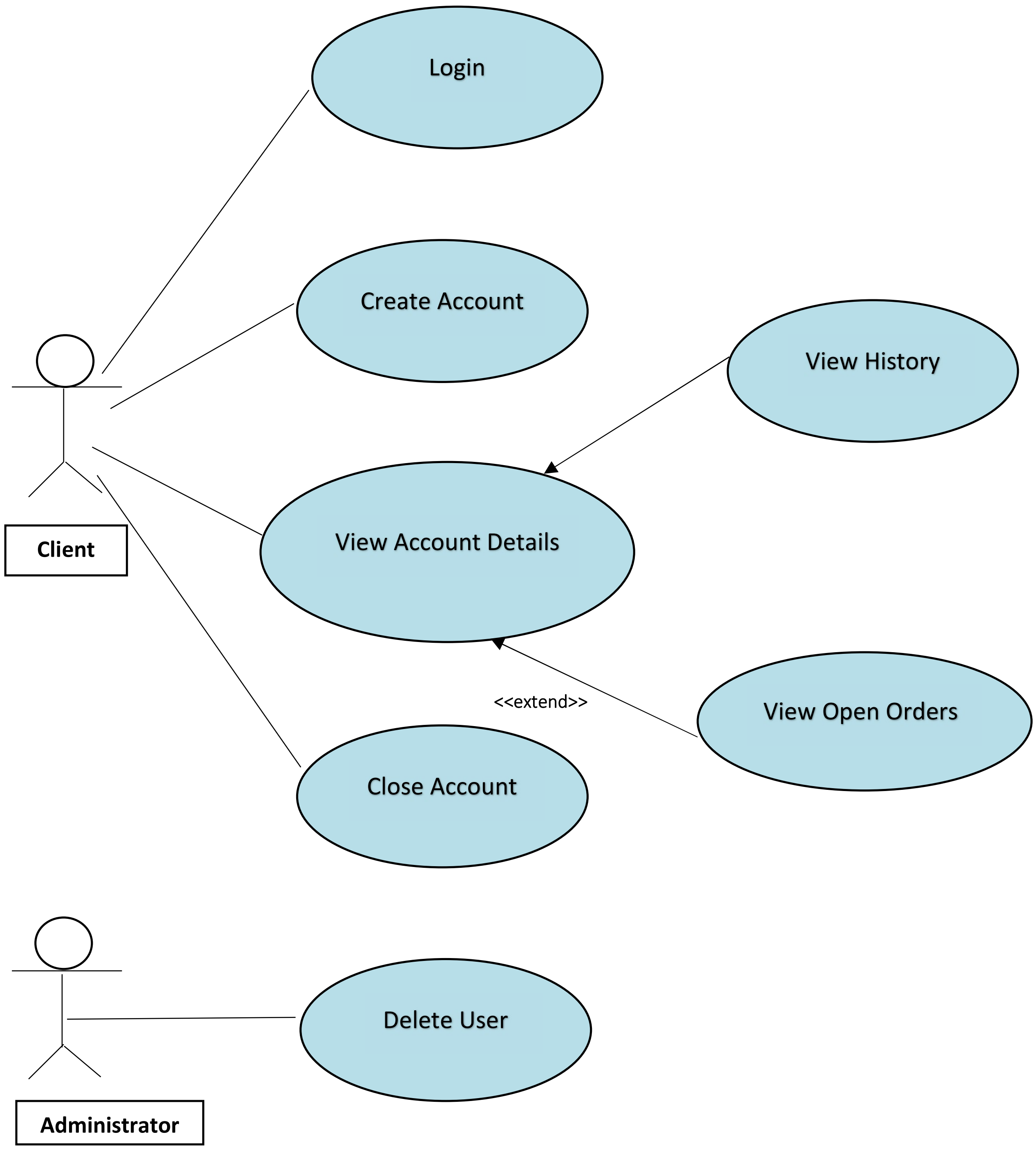
- Encapsulation in Java is the process of wrapping up of properties and behaviors of an object into a single unit and the unit here is a class.

10.What is Inheritance?
Ans.: Inheritance means that one class inherits the characteristics of another class. This is also called a "**is a"** relationship. One of the most useful aspects of object-oriented programming is code reusability. As the name suggests 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 is a very important concept of object-oriented programming since this feature helps to reduce the code size.

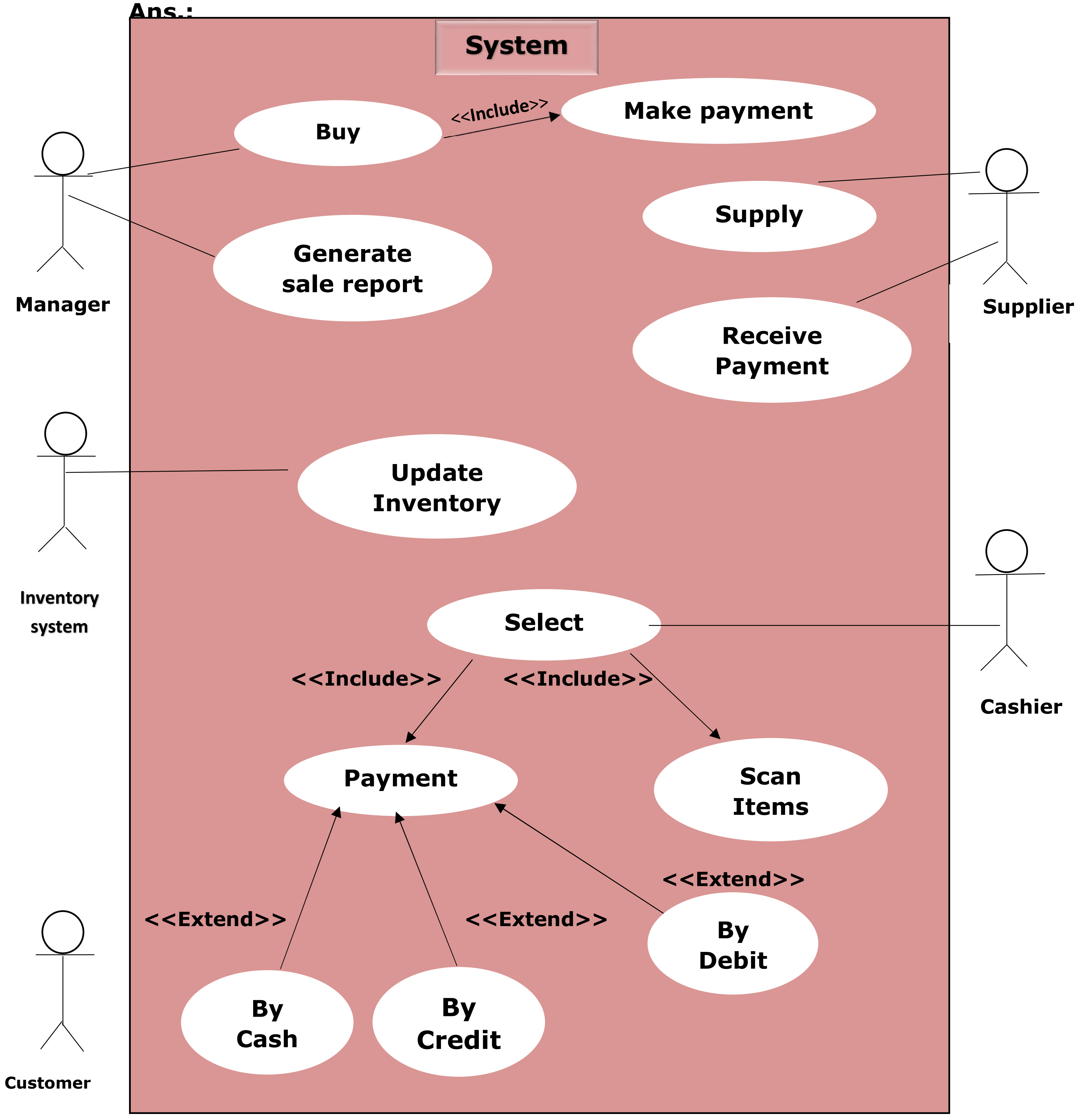
11.What is Polymorphism?
Ans.:
- Polymorphism means "**having many forms."** It allows different objects to respond to the same message in different ways, the response specific to the type of object.
- The most important aspect of an object is its behavior. A behavior is initiated by sending a message to the object.
- The ability to use an operator or function in different ways in other words giving different meaning or functions to the operators or functions is called Polymorphism.

12. Draw use case on Online Book Shopping.
Ans.:

Online book store



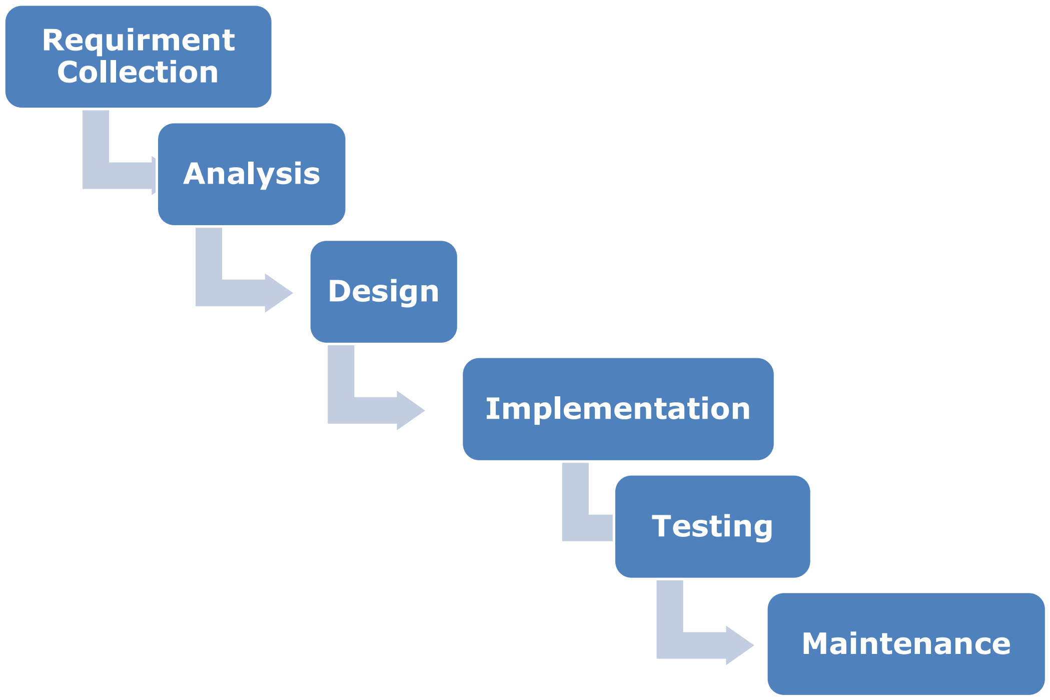
13. Draw Use case on online bill payment system (paytm).
Ans.:



14.Write SDLC phases with basic introduction.
Ans.:

SDLC Phases	Security Processes
Requirements	Security analysis for requirements and check abuse/misuse cases
Design	Security risk analysis for designing. Development of test plan including security tests
Coding and Unit Testing	Static and Dynamic Testing and Security white box testing
Integration Testing	Black box Testing
System Testing	Black box testing and Vulnerability scanning
Implementation	Penetration Testing, Vulnerability Scanning
Support	Impact analysis of Patches

15.Explain Phases of the waterfall model.
Ans.:

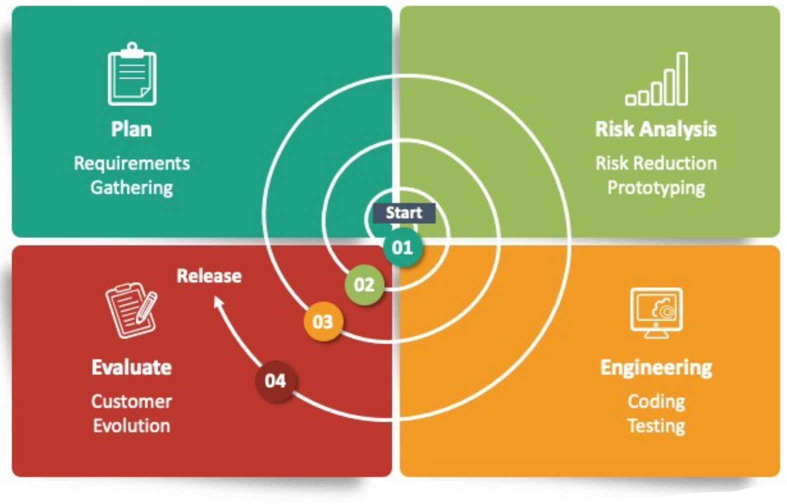


- **Requirement Collection:** All possible requirements of the system to be developed are captured in this phase and documented in a requirement specification document.

- **Analysis:** The requirement document delivered by the requirement phase and maps the requirement.
- **Design:** The requirement specifications from first phase are studied in this phase and the system design is prepared. This system design helps in specifying hardware and system requirements and helps in defining the overall system architecture.
- **Implementation:** With inputs from the system design, the system is first developed in small programs called units, which are integrated in the next phase. Each unit is developed and tested for its functionality, which is referred to as Unit Testing.
- **Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit. Post integration the entire system is tested for any faults and failures.
- **Maintenance:** There are some issues which come up in the client environment. To fix those issues, patches are released. Also, to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.

16.Write phases of Spiral model.

Ans.:



1. **Planning for Requirements Gathering:** Requirements are gathered from the customers and the objectives are identified, elaborated, and analyzed at the start of every phase. Then alternative solutions possible for the phase are proposed in this quadrant.
2. **Risk Analysis:** During the second quadrant, all the possible solutions are evaluated to select the best possible solution. Then the risks associated with that solution are identified and the risks are resolved using the best possible strategy. At the end of this quadrant, the Prototype is built for the best possible solution.
3. **Coding Testing:** During the third quadrant, the identified features are coding and verified through testing. At the end of the third quadrant, the next version of the software is available.
4. **Customer Evolution:** In the fourth quadrant, the Customers evaluate the so far developed version of the software. In the end, planning for the next phase is started.

17.Write Agile Manifesto Principles.
Ans.:

- **Customer satisfaction through early and continuous software delivery**: Customers are happier when they receive working software at regular intervals, rather than waiting extended periods of time between releases.
- **Accommodate changing requirements throughout the development process**: The ability to avoid delays when a requirement or feature request changes.
- **Frequent delivery of working software**: Scrum accommodates this principle since the team operates in software sprints or iterations that ensure regular delivery of working software.
- **Collaboration between the business stakeholders and developers throughout the project**: Better decisions are made when the business and technical team are aligned.
- **Support, trust, and motivate the people involved**: Motivated teams are more likely to deliver their best work than unhappy teams.
- **Enable face-to-face interactions**: Communication is more successful when development teams are co-located.
- **Working software is the primary measure of progress:** Delivering functional software to the customer is the ultimate factor that measures progress.

- **Agile processes to support a consistent development pace:** Teams establish a repeatable and maintainable speed at which they can deliver working software, and they repeat it with each release.
- **Attention to technical detail and design enhances agility:** The right skills and good design ensures the team can maintain the pace, constantly improve the product, and sustain change.
- **Simplicity** – Develop just enough to get the job done for right now.
- **Self-organizing teams encourage great architectures, requirements, and designs:** Skilled and motivated team members who have decision-making power, take ownership, communicate regularly with other team members, and share ideas that deliver quality products.
- **Regular reflections on how to become more effective:** Self-improvement, process improvement, advancing skills, and techniques help team members work more efficiently.

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

Ans.: Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes to deliver specific features for a release.

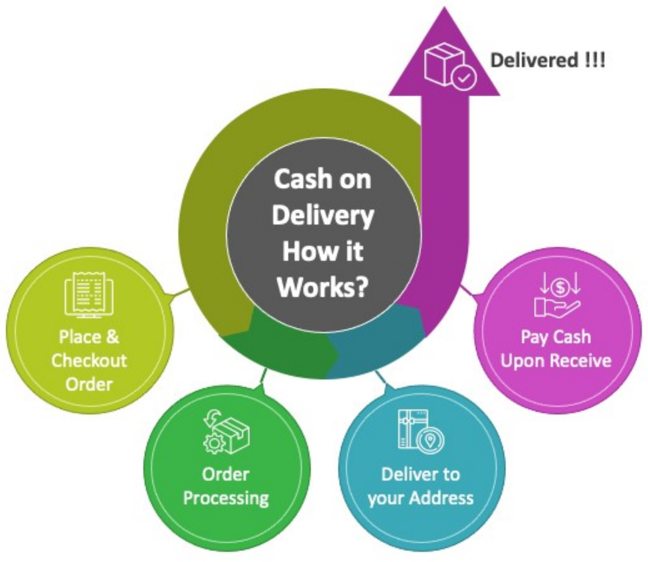
<u>Pros:</u>

- 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:

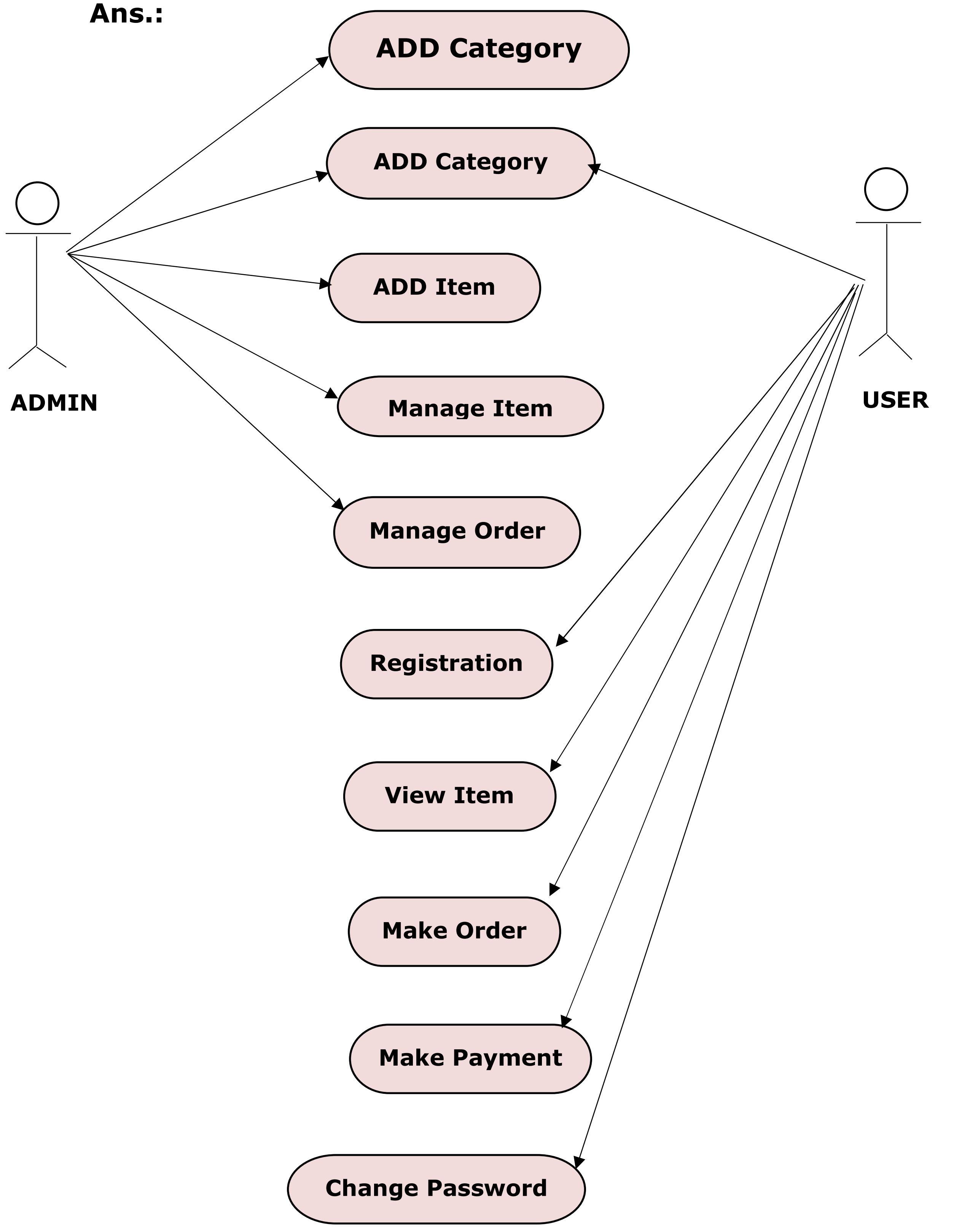
- Not suitable for handling complex dependencies.
- More risk of sustainability, maintainability, and extensibility.
- An overall plan, an agile leader and agile PM practice is a must without which it will not work.
- Strict delivery management dictates the scope, functionality to be delivered, and adjustments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

19.Draw usecase on online shopping product using COD.
Ans.:



20.Draw usecase on Online Shopping product using payment gateway.

Ans.:


ADD Category
ADD Category
ADD Item
Manage Item
Manage Order
Registration
View Item
Make Order
Make Payment
Change Password
ADMIN
USER