Module: 1 (SDLC)

1. What is software? What is software engineering?

Software: - Software is a set of programs, Set of Instruction Used to execute specific task or Function.

- Two Type of software: - **1**. Application Software
  2. System Application
  3. Driver Software
  4. Middleware Software
  5. Programming Software

1) Application Software : These are programs designed to perform specific tasks for users. They can be further divided into categories like.

2) System Software: Software that provides a platform for running application software and manages hardware resources. Examples include device drivers, utility programs, and software development tools.

3) Driver Software: A driver in software provides a programming interface to control and manage specific lower-level interfaces that are often linked to a specific type of hardware, or other low-level service. In the case of hardware, the specific subclass of drivers controlling physical or virtual hardware devices are known as device drivers.

4) Middle ware Software: Software that acts as an intermediary between different software applications or components. It helps facilitate communication and data exchange. Examples include web servers, application servers, and messaging middleware.

5) Programming Software: Programming software is a software which helps the programmer in developing other software. Compilers, assemblers, debuggers, interpreters etc. are examples of programming software. Integrated development environments (IDEs) are combinations of all these software.

Software Engineering: -

- Software Engineering is a Technique Through Which We Can developed or created. Software for computer system and any other electronic device.
- In other words, Software engineering is a process in which user needs are analysed and software is design base on there needs.
- In software engineering the development of software using well define scientific principal method and procedures.
- Software engineering build these software and application by using designing and programming languages.
- In order to a complex software, we should use Software engineering technique as well as to reduce the complexify we should use abstraction and decomposition where abstraction describes only important part of the software and leave irrelevant things for later stage of development so the requirement of software become simple.
- And decomposition breakdown the software into no of module where each module produces a well define independent task.

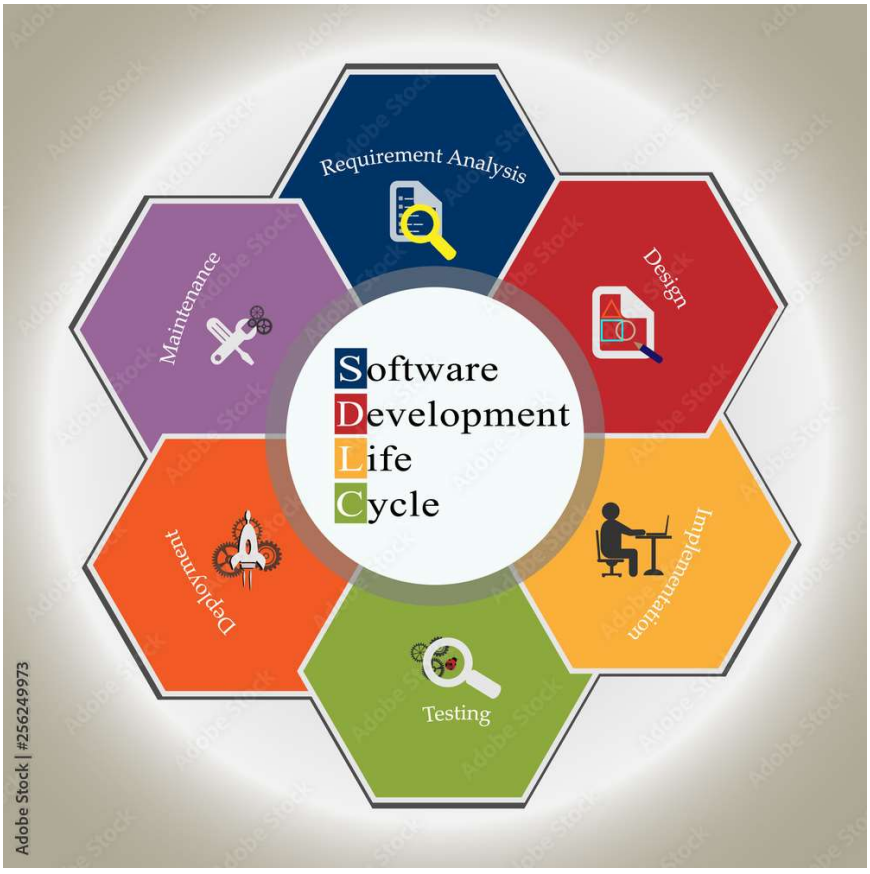
❖ Purpose of Software Engineering

-
- To manage large Software
- For Greater Scalability
- To manage the cost
- To manage the dynamic nature of the software
- For batter quality management

❖ What is SDLC? Full Explanation

- ❖ SDLC Stands for “Software Development Life Cycle” Model It describes the sequence of phases or Steps to develop any Software
- ❖ In simple word “Entire life time of Software From Beginning To ending.”
- ❖ It Contain Three Main Stages
    1. Conception stage
    2. Implementation Stage
       Maintenance
- ❖ The SDLC Model is classified into three Category based on three Advantages and Dis-advantages.
    1. Water Fall (Classical Model)
    2. Prototype Model
    3. Spiral Model

• Below is the diagrammatic representation of the SDLC cycle:



• Full Explanation Of SDLC:

Requirement Analysis:

❖ This phase involves gathering and understanding the project's requirements from stakeholders, clients, and end-users. The goal is to define what the software needs to achieve, its features, functionalities, and constraints. Requirements are documented in detail to serve as a foundation for the entire project.

Planning:

❖ In this phase, the development team creates a project plan that outlines the scope, timeline, resources, budget, and risks associated with the project. The plan also includes a breakdown of tasks, responsibilities, and milestones. Planning sets the overall direction for the project and helps manage expectations

Design:

❖ During the design phase, the software architecture is defined, including high-level system components, databases, modules, interfaces, and data flow. Design decisions also consider factors like scalability, maintainability, and performance. This phase translates the requirements into a technical blueprint for the development team to follow.

Implementation (Coding):

❖ In this phase, the actual coding of the software takes place. Developers write code based on the design specifications, using programming languages and coding practices that align with the project's requirements. Code is written in small, manageable units and is reviewed by peers to ensure quality and adherence to coding standards.

Testing:

❖ The testing phase involves systematically evaluating the software's functionality, performance, and security to identify defects and ensure it meets the specified requirements. Different testing levels include unit testing (testing individual units/modules), integration testing (testing interactions between units/modules), system testing (testing the entire system), and user acceptance testing (validating with end-users).

Deployment:

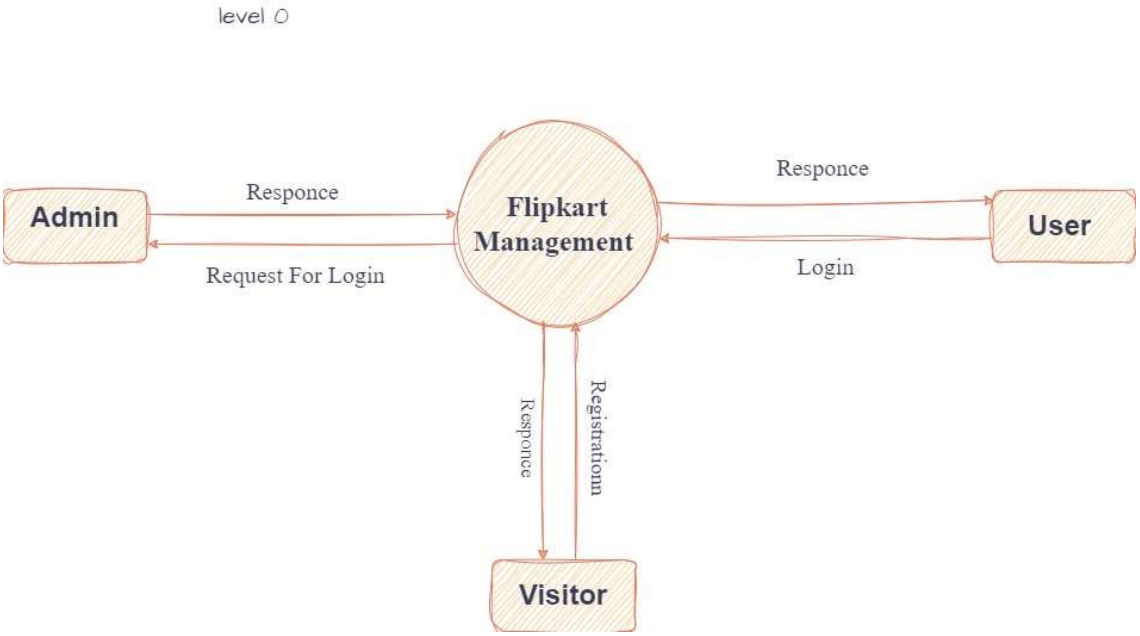
❖ Once the software passes all testing phases, it's ready for deployment. Deployment involves releasing the software to the production environment, making it accessible to end-users. This phase requires careful planning to ensure a smooth transition and minimal disruption.

Maintenance and Support:

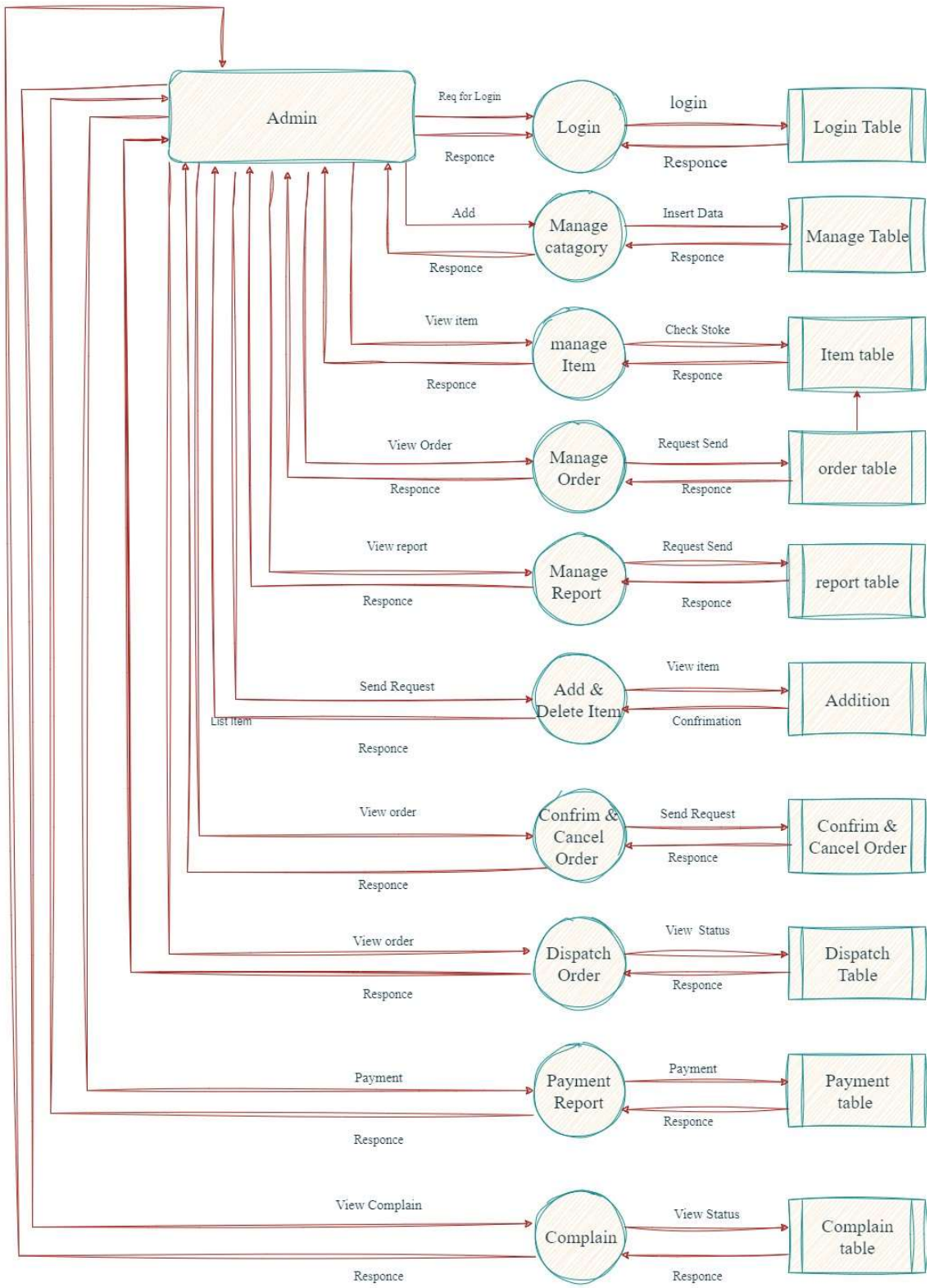
- After deployment, the software enters the maintenance phase. This involves monitoring the software for issues, fixing bugs, providing updates, and making improvements based on user feedback. Maintenance can be corrective (bug fixes), adaptive (supporting changes in the environment), perfective (adding new features), or preventive (proactively preventing issues).

Flipkart DFD

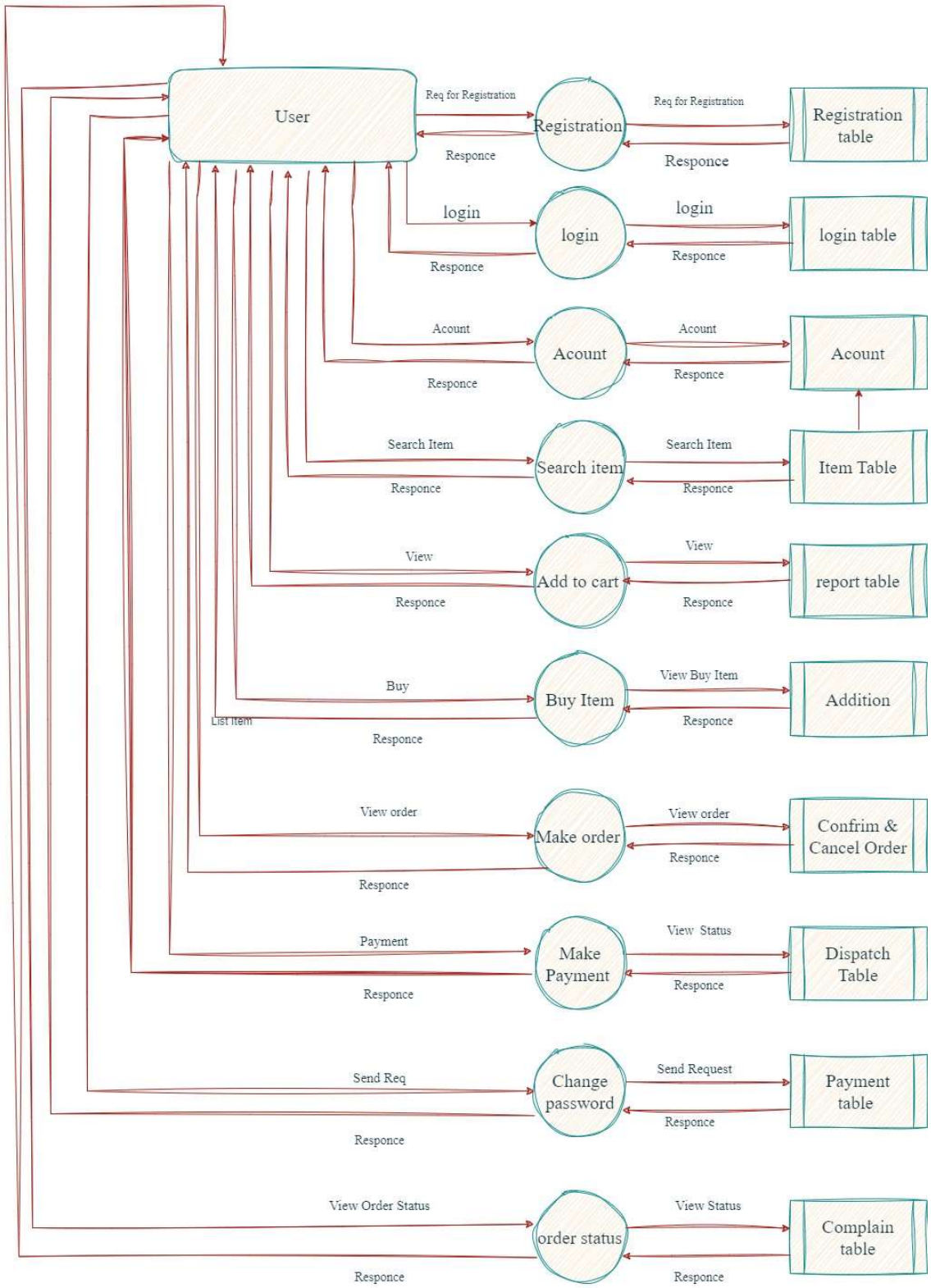
0 Level DFD



1 Level DFD:


Admin
Req for Login
Responce
Login
login
Responce
Login Table
Add
Responce
Manage catagory
Insert Data
Responce
Manage Table
View item
Responce
manage Item
Check Stoke
Responce
Item table
View Order
Responce
Manage Order
Request Send
Responce
order table
View report
Responce
Manage Report
Request Send
Responce
report table
Send Request
List Item
Responce
Add & Delete Item
View item
Confirmation
Addition
View order
Responce
Confrim & Cancel Order
Send Request
Responce
Confrim & Cancel Order
View order
Responce
Dispatch Order
View Status
Responce
Dispatch Table
Payment
Responce
Payment Report
Payment
Responce
Payment table
View Complain
Responce
Complain
View Status
Responce
Complain table

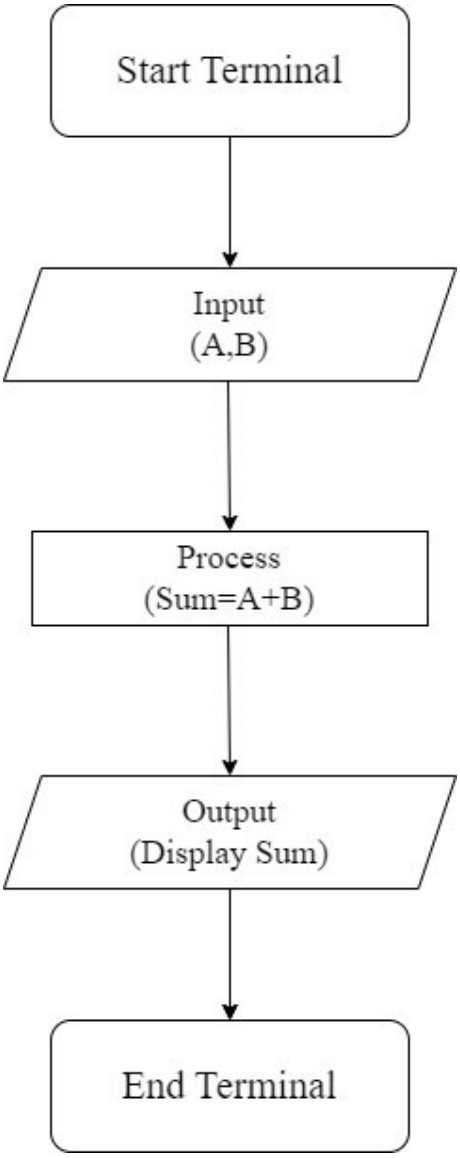
2 Level DFD:


User
Req for Registration
Responce
Registration
Req for Registration
Responce
Registration table
login
Responce
login
login
Responce
login table
Acount
Responce
Acount
Acount
Responce
Acount
Search Item
Responce
Search item
Search Item
Responce
Item Table
View
Responce
Add to cart
View
Responce
report table
Buy
List Item
Responce
Buy Item
View Buy Item
Responce
Addition
View order
Responce
Make order
View order
Responce
Confrim & Cancel Order
Payment
Responce
Make Payment
View Status
Responce
Dispatch Table
Send Req
Responce
Change password
Send Request
Responce
Payment table
View Order Status
Responce
order status
View Status
Responce
Complain table

❖ What is Flow chart? Create a flowchart to make addition of two numbers

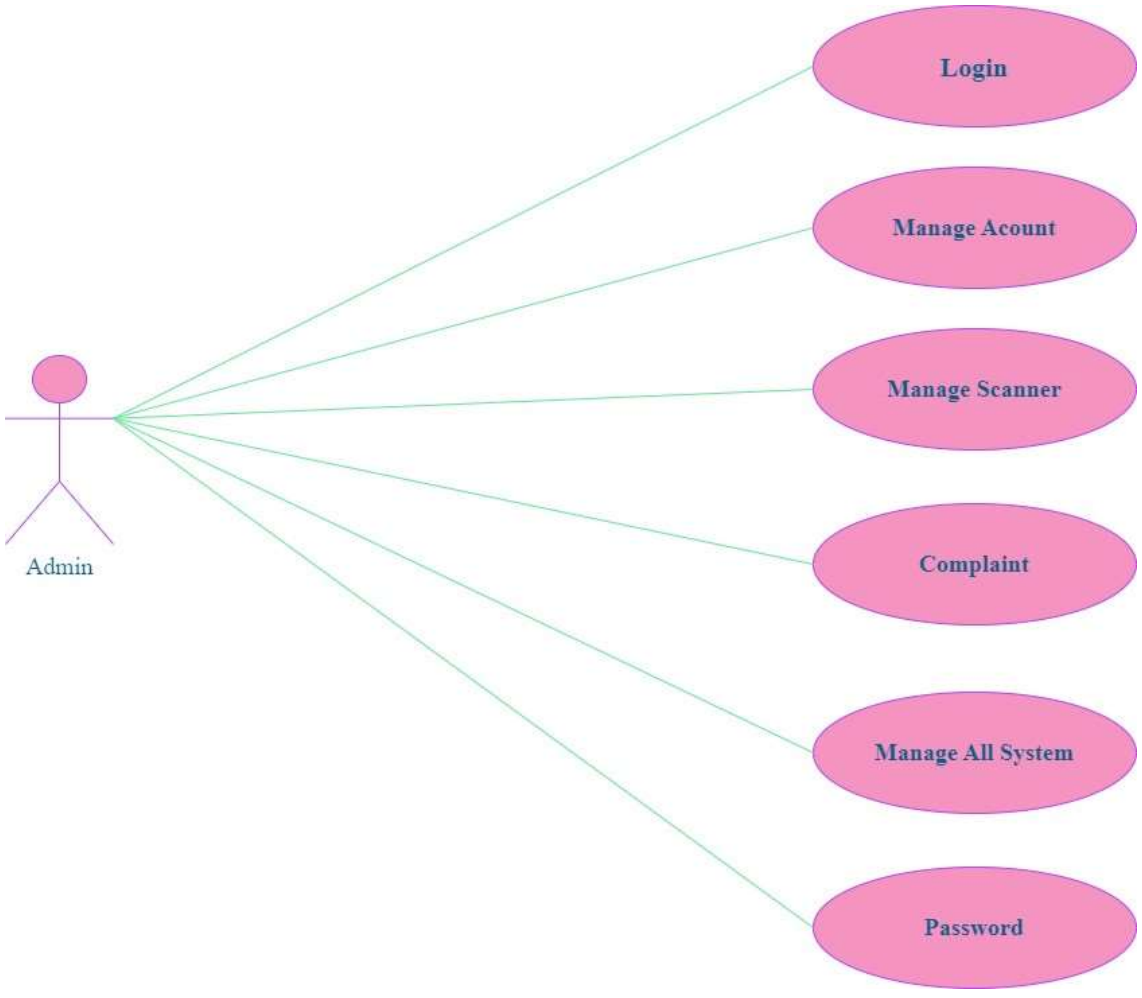
- Ans:-

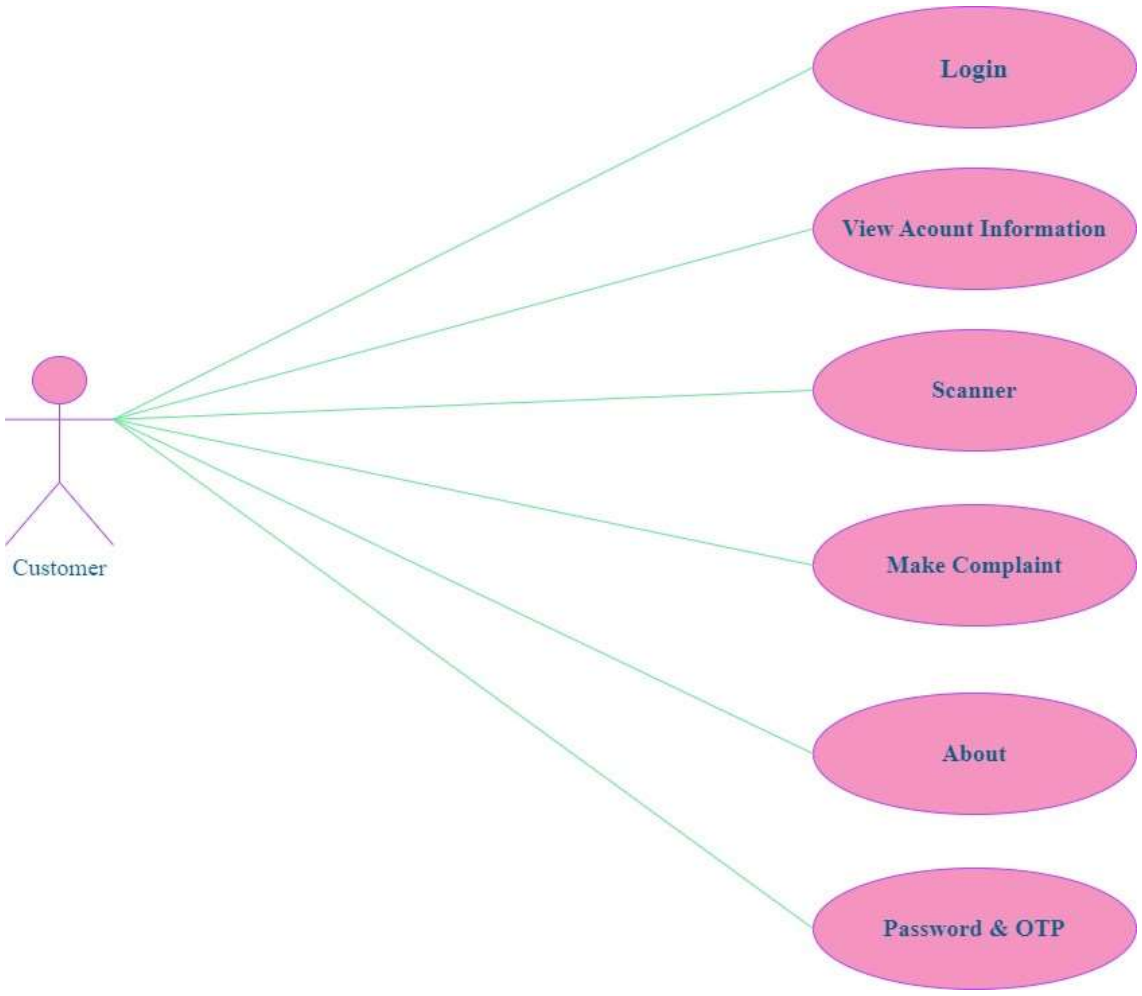
❖ A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.



❖ What is Use case Diagram? Create a use-case on bill payment on Payment.

❖ A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.




Login
View Acount Information
Scanner
Make Complaint
About
Password & OTP
Customer