TASK-7

Object Diagram and Software Architecture Diagram

Aim:

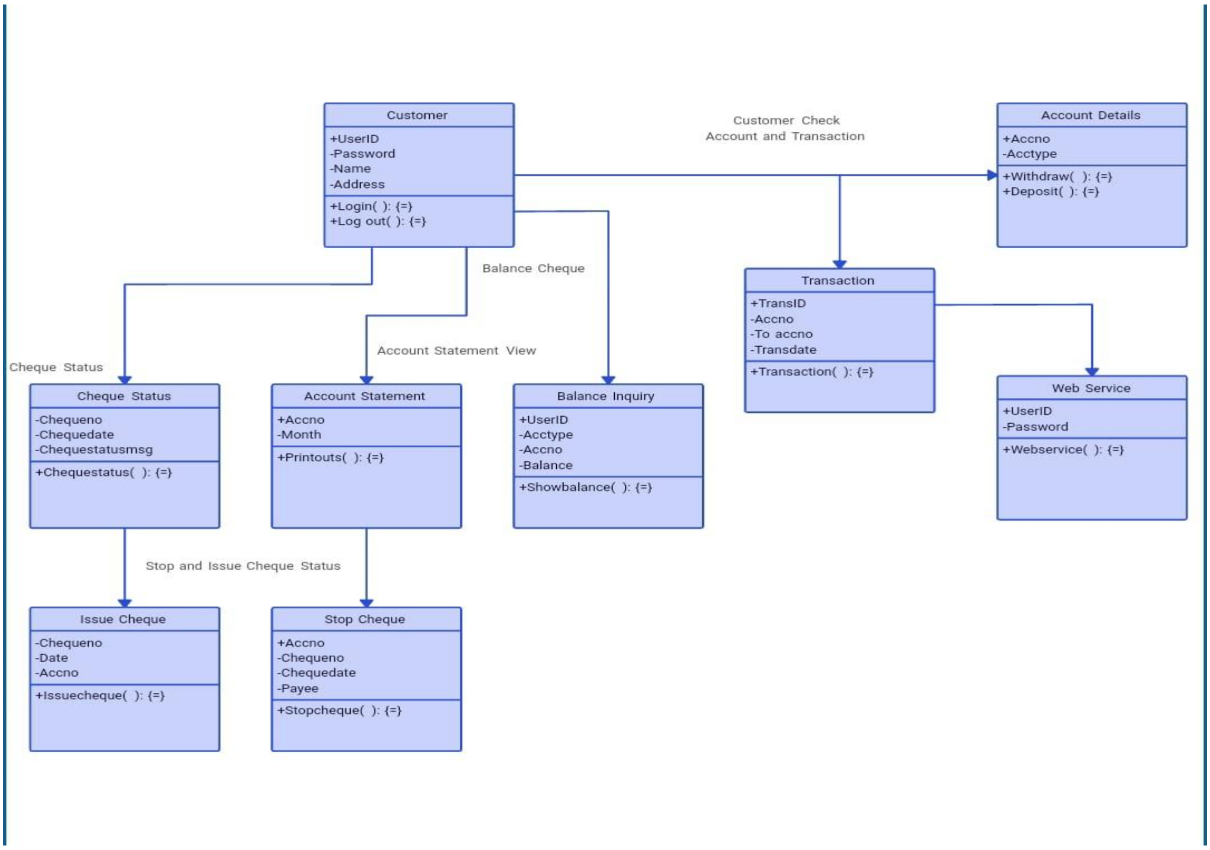
The aim is to implement partial layered, object-oriented, and logical software architecture diagrams to visualize the high-level design of the Online Banking System**.**

Object Diagram:

A UML Object Diagram is an instance-level representation of classes, showing objects and their relationships at a specific point in time. It helps in understanding how different entities interact in a real system scenario.

Object Diagram Symbols and Notations:

1. Object Name
2. Object Attributes
3. Links (Relationships between Objects**)**



OBJECT DIAGRAM

Diagram Explanation:

In the Online Banking System, the main classes are Bank, Account, Customer, and Transaction.

- The Bank object manages a collection of Customer and Account objects. It provides services for account creation, fund transfer, and report generation.
- Each Customer object has attributes such as *Customer ID*, *Name*, *Address*, *Contact Info*, and is associated with one or more Account objects. The customer can perform actions such as login, check balance, and transfer funds.
- Each Account object has attributes like *Account Number*, *Account Type*, *Balance*, and maintains links to Transaction objects. It supports operations like *deposit()*, *withdraw()*, and *updateBalance()*.
- The Transaction object records details such as *Transaction ID*, *Date*, *Amount*, *Type (Debit/Credit)*, and *Status*. It is linked to the corresponding Account and Customer that initiated it.

These interconnected objects demonstrate real-time interactions in online banking, reflecting how customer activities trigger account and transaction updates.

Architecture Diagram:

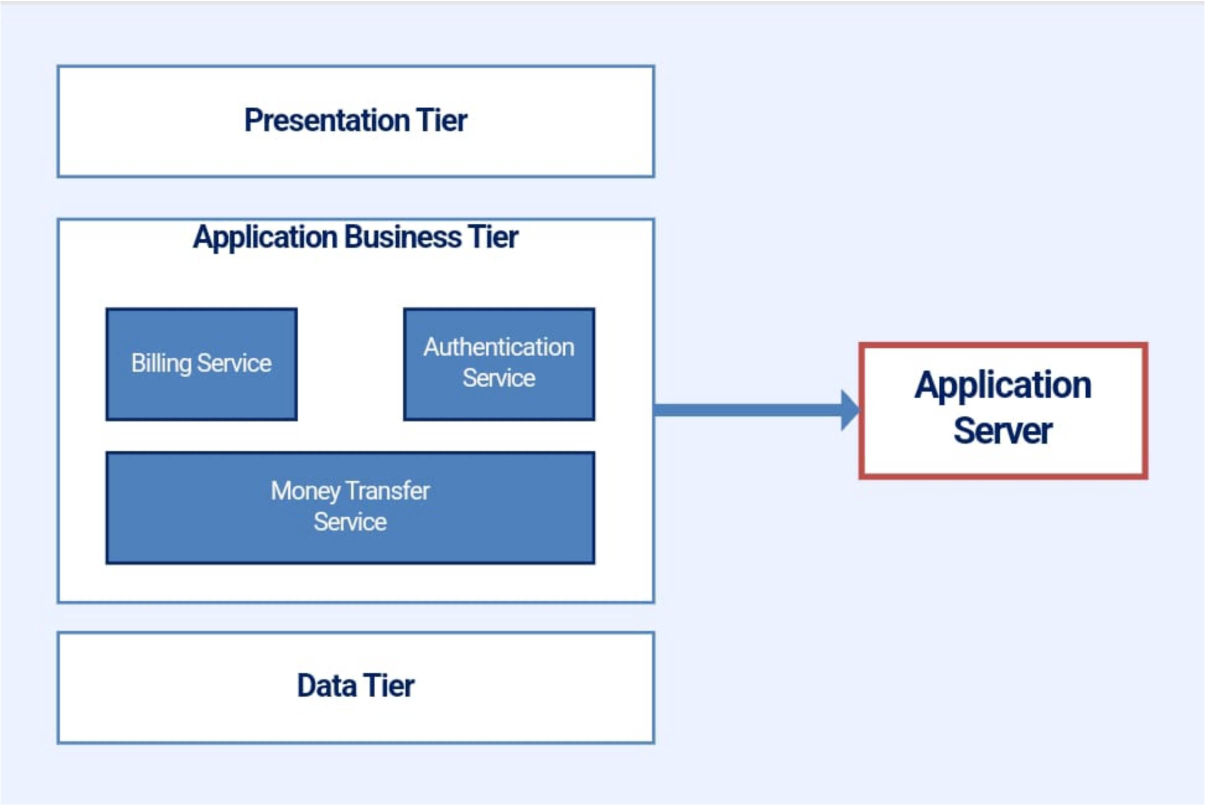
Software Architecture defines the overall structure of the Online Banking System, showing how components and layers interact to achieve secure and efficient operation.

The architecture follows a three-tier layered design:

1. Presentation Layer:
    - User interfaces for customers and administrators (web/mobile apps).
    - Handles login, navigation, and transaction input.
2. Business Logic Layer:
    - Implements core functionalities such as authentication, fund transfer, bill payment, and account management.
    - Enforces business rules, validations, and transaction security.
3. Data Layer:
    - Manages the database containing user information, account details, and transaction histories.
    - Ensures data consistency, backup, and secure storage using encryption.

The layers communicate through well-defined interfaces, ensuring modularity, maintainability, and scalability.

This architecture supports multiple users simultaneously, protects sensitive data through secure APIs, and allows easy integration with other financial systems**.**



ARCHITECTURE DIAGRAM

Result:

Thus, the object-oriented and logical software architecture diagrams for the Online Banking System have been designed to visualize the system's high-level structure and inter-object relationships effectively.