Assignment-01

Student Name: Tanay Manish Nesari **UID:** 23BCS13761
Branch: BE-CSE **Section/Group:** KRG_2B
Semester: 6th **Date of Performance:** 30/1/26
Subject Name: System Design
Subject Code: 23CSH-314

Q1. Explain the role of **interfaces and enums** in software design with proper examples.
(2.5 marks)

Ans) **Interface:** An interface in software design is used to define a common set of methods that different classes must implement, ensuring uniform behavior across the system. Interfaces help in achieving abstraction and flexibility by allowing the implementation to vary independently from the usage.

For example, in a **Payment Processing Application**, an interface `PaymentMethod` can define a method `payAmount()`, which is implemented by classes such as `CreditCardPayment`, `DebitCardPayment`, and `UPIPayment`. This allows the application to process payments through different modes without changing the core business logic, making the system easier to maintain and extend.

Enums: Enums are used in software design to represent a fixed set of predefined constants, ensuring consistency and type safety in applications. They help avoid errors caused by invalid values and improve code readability.

For example, in an **Order Management System**, an enum `OrderStatus` can contain values such as `PENDING`, `CONFIRMED`, `SHIPPED`, and `DELIVERED`. Using an enum ensures that the order status can only take one of these valid states, reducing bugs and making the application logic clearer and more reliable.

Q2. Discuss how **interfaces enable loose coupling** with an example.
(2.5 marks)

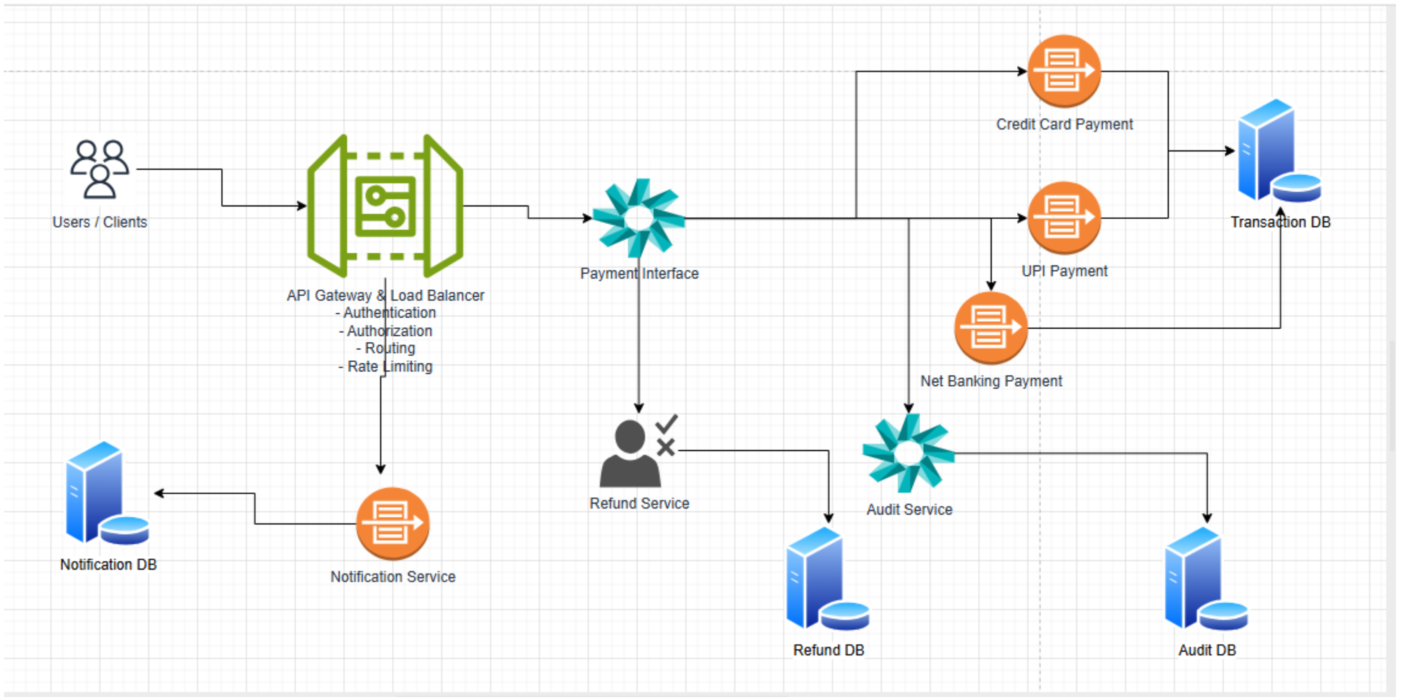
Ans) Interfaces enable loose coupling by allowing a program to depend on abstractions rather than concrete implementations.

- Interfaces define *what* an object can do, not *how* it does it.
- The calling class depends only on the interface, not the actual implementation.
- Changes in implementation do not affect the dependent classes.
- New implementations can be added easily without modifying existing code.

For example, In a **Notification System**, an interface `NotificationService` defines a method `sendNotification()`. This interface is implemented by classes such as `EmailNotification` and `SMSNotification`. The main application communicates only with the `NotificationService` interface. As a result, notification methods can be changed or extended without altering the core system, achieving loose coupling and better maintainability.

Q3. Design a **High-Level Design (HLD)** for a **Payment Processing System**, showing where **interfaces** would be used

Ans)


Users / Clients
API Gateway & Load Balancer
- Authentication
- Authorization
- Routing
- Rate Limiting
Payment Interface
Credit Card Payment
UPI Payment
Net Banking Payment
Transaction DB
Refund Service
Audit Service
Notification DB
Notification Service
Refund DB
Audit DB