Day 1

Requirements (Money Transfer & Audit Logs)

**Scope**

* Money transfers
* Balance validation before transfer.
* Failure handling for insufficient funds or invalid accounts.
* Audit log generation for each transfer event.
* Secure authentication of transfer requests.

**Functional Requirements**

1. **Money Transfer**
   * The system shall allow customers to initiate a transfer from a source account to a destination account.
   * The system shall validate sufficient funds before executing the transfer.
   * The system shall debit the source account and credit the destination account atomically.
   * The system shall provide success/failure status to the customer in real-time.
2. **Audit Logs**
   * The system shall create an audit log entry for each transfer (successful or failed).
   * The audit log shall capture:
     + Transaction ID
     + Source account & destination account
     + Amount & currency
     + Before & after balances
     + Status (success/failure)
     + Initiator (user/system)
     + Timestamp
   * Audit logs shall be **immutable** (cannot be altered or deleted).
   * Audit logs shall be accessible to compliance and admin teams for reporting.

**Non-Functional Requirements (NFRs)**

* **Security**: All transfers and logs must be encrypted in transit and at rest.
* **Consistency**: Ensure ACID compliance for balance updates.
* **Performance**: Transfers must complete within **<10 seconds** under normal load.
* **Availability**: System uptime must be at least **99.9%**.
* **Scalability**: Support at least **100,000 transfers/day** initially, with growth potential.
* **Auditability**: Logs must be tamper-proof and easily retrievable for at least **7 years**

**Business Objectives**

* Enable customers to **transfer funds** between their accounts (or to other customers’ accounts).
* Ensure **accuracy and integrity** of account balances during transfers.
* Maintain **comprehensive audit logs** for regulatory compliance and fraud detection.
* Support **scalability** for high transaction volumes.
* Provide **real-time confirmations** for transfers while ensuring data durability.

**Key Classes**

* Customer
* Account
* Transaction
* AuditLog
* TransferService
* AuditService
* Repositories
  + AccountRepository
  + TransactionRepository
  + AuditRepository