**Project Name: Banking Transaction System**

**Submitted By:**

* **Name: Pratheesh K**
* **Amazon ID: ukprath**
* **Batch: 2**

**Table of Contents**

1. Introduction
2. Problem Statement
3. Functional Requirements
4. Non-Functional Requirements
5. Deliverable

**1.Introduction:-**

The purpose of this document is to define the technical and business requirements for the Banking Transaction System enables secure deposits, withdrawals, and money transfers while ensuring accurate account balances. It generates detailed audit logs for every transaction to support transparency and compliance. The system is designed to be efficient, reliable, and aligned with banking standards. Its goal is to deliver data integrity, accountability, and trust for financial institutions.

**2.Problem Statement:-**

Traditional banking systems often face issues like delays, errors, limited audit tracking, and outdated security, leading to customer dissatisfaction and compliance risks.

**Traditional Banking Systems**

* Transactions often delayed due to manual processing
* Prone to human errors and mismatched balances
* Limited or no audit tracking for accountability
* Outdated security protocols vulnerable to threats
* Lack of real-time updates on account operations

**Customer Expectations**

* Fast and secure money transfers
* Accurate and transparent account transactions
* Immediate feedback on all operations
* Strong security to protect financial data
* Trustworthy system with clear audit trails

**3.Project Scope**

* Coding of transaction modules: deposit, withdrawal, and transfer.
* User authentication, PIN management, and session handling.
* Logging and auditing through coded logger utilities.
* Database integration for storing customer accounts and transaction history.
* Unit testing and integration testing

**4.Functional Requirements:-**

**4.1 Money Transfer:-**

* **User Management:** Register new users with essential details (name, email, phone, address) and link them to one or more accounts.
* **PIN Management**: Provide secure creation and updating of PINs to authorize transactions.
* **User Authentication:** Ensure only authorized users can access the system and perform transfers.
* **Money Transfer Functionality:** Allow secure fund transfers to other accounts with validation of account details and confirmation of completion.
* **Transaction Status Tracking**: Provide real-time updates and notifications about the status of money transfer requests.
* **Error Handling and Rollbacks**: Detect failed transactions promptly, revert balances to the pre-transaction state, and notify users.

**4.2 Audit Logs:-**

* **Transaction Logging:** Record every transaction (deposit, withdrawal, transfer) and key system events in an immutable log.
* **Log Details:** Each log entry shall include User ID, Timestamp, Source and Destination Account, Amount, Transaction Type, and Status.
* **Immutable Records:** Ensure audit logs cannot be altered or deleted by unauthorized users.
* **Reporting:** Allow administrators to generate reports from audit logs for monitoring, compliance, and reviews.

**5.Non-Functional Requirements:-**

* **Security**: Encrypt all sensitive data in storage and transit, restrict access to audit logs, and mask confidential fields.  
  Example: Customer PINs stored using AES-256 encryption, and account numbers partially masked in audit logs (e.g., \*\*\**6789).*
* **Performance**: Ensure fast processing of transfers and immediate audit log entries without noticeable delay.  
  Example: A transfer of ₹10,000 between two accounts must complete in under 2 seconds, with the audit log recorded instantly.
* **Reliability**: Guarantee ACID (Atomicity, Consistency, Isolation, Durability) for all transactions.  
  Example: If a ₹50,000 transfer fails midway, the sender’s balance must be rolled back to its original state.
  + - **Atomicity** → A transaction happens fully or not at all.
    - **Consistency** → The system moves from one valid state to another valid state.
    - **Isolation** → Transactions running at the same time don’t affect each other.
    - **Durability** → Once a transaction is confirmed, it stays recorded even after failures.
* **Scalability**: Support increasing users and transactions without performance loss.  
  Example: The system should handle 10,000 concurrent transfers per second during peak hours without degradation.
* **Usability**: Provide an intuitive interface for customers and administrators.  
  Example: A customer can complete a transfer in fewer than 3 clicks, and an auditor can filter logs by date with a single query.
* **Compliance**: Retain logs for the legally required duration and meet financial regulations.
  + KYC (Know Your Customer)
  + AML (Anti-Money Laundering)
  + Data Privacy Compliance

**6. Deliverable:-**

* Defines the Problem Statement.
* Defines the purpose, scope, and requirements of the Banking Transaction System.
* Provides a structured roadmap for the design and development phases.

**Technologies Used**

Language Java

IDE IntelliJ IDEA

Database AWS DynamoDB

Build Tool Maven

Version Control Git, GitHub

Testing JUnit / Cucumber

DevOps Jenkins