# **Requirement Specification Document**

**Project Name:** Banking Transaction System  
**Prepared By:** Arun Kumar R  
**Date:** 09/06/2025

## **Git : https://github.com/aruntokumar/Atlas-Docs-Arun-Kumar/tree/main/Project-Banking\_Transaction\_System\_Arun%20Kumar%20R**

## **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 and outline the technical and business requirements for the Banking Transaction System. This application will enable customers to perform essential banking operations, such as money transfers, while ensuring the generation of accurate audit logs for every transaction. The system is intended to be secure, efficient, and compliant with banking standards. It is targeted toward financial institutions seeking improved operational transparency and accountability.

## **2. Problem Statement**

Traditional banking systems often suffer from performance bottlenecks, insufficient audit tracking, and outdated security protocols. With rising customer expectations for real-time fund transfers and increasing regulatory scrutiny, there is a need for a robust banking solution.

Customers expect:

* Fast and secure money transfers
* Transparency in account transactions
* Immediate feedback on account operations

Banks require:

* Accurate logs for auditing and compliance
* Systems that can handle large volumes with high reliability
* Secure platforms to mitigate fraud and cyber threats

The Banking Transaction System addresses these challenges by offering a modern, secure, and auditable platform for financial transactions.

## **3. Functional Requirements**

**3.1 Money Transfer**

* The system shall allow customers to transfer money from one account to another using valid account numbers.
* The system shall verify the sender’s account balance before executing any transfer to prevent overdrafts.
* The system shall update the balances of both sender and receiver accounts immediately after the transaction.
* The system shall generate a unique transaction ID for each money transfer.
* The system shall support multiple modes of bank transfers, including:
  + NEFT (National Electronic Funds Transfer)
  + RTGS (Real-Time Gross Settlement)
  + IMPS (Immediate Payment Service)
  + Internal transfers (between accounts within the same bank)
* The system shall restrict transfer modes based on bank business rules and time windows, such as:
  + NEFT transfers are allowed only during banking hours (e.g., 8 AM to 6 PM on working days).
  + RTGS transfers are allowed for transactions above a specified minimum amount (e.g., ₹2,00,000).
  + IMPS transfers are allowed 24/7 but may have a daily transaction limit (e.g., ₹2,00,000).
* The system shall display applicable transfer modes based on the user’s transaction context (amount, time, recipient bank).
* The system shall record the selected transfer mode in the audit logs along with the transaction.
* In the case of scheduled transfers (e.g., NEFT requests outside service hours), the system shall queue the transaction and process it during the next valid window.
* The system shall log balance inquiries made before transfers as part of the audit trail, including timestamp, account number, and user ID.

**3.2 Audit Logs**

* The system shall maintain a persistent audit log that records every financial operation, including deposit, withdrawal, transfer, and balance inquiries.
* The system shall log non-financial user activities such as login and logout events to enhance traceability and session-level auditing.
* Each audit log entry shall include:
  + Event Type (e.g., deposit, withdrawal, transfer, balance check, login, logout)
  + Transaction ID (if applicable)
  + Timestamp (date and time of the event)
  + Amount (if applicable)
  + Sender Account (if applicable)
  + Receiver Account (if applicable)
  + User ID or Username
  + IP Address or Device ID (if available)
* The system shall allow authorized users (e.g., compliance officers, auditors, system administrators) to securely access audit logs for regulatory and internal review purposes.
* The system shall store audit logs in both:
  + Relational database (SQL) to maintain ACID-compliant transaction integrity
  + NoSQL-based storage (e.g., DynamoDB, MongoDB) for scalable, high-speed querying and archival
* Audit logs shall be immutable and tamper-proof; once recorded, they must not be altered or deleted by unauthorized users.
* The system shall support filtering and exporting logs based on event type, date range, account number, or user ID.
* The system shall retain audit logs in accordance with applicable regulatory policies (e.g., minimum 5 years as per banking standards).

## **4. Non-Functional Requirements**

**4.1 Security**

* All financial transactions must be encrypted using industry-standard protocols (e.g., HTTPS/TLS).
* Users must authenticate securely before accessing any financial operation.
* Role-based access control (RBAC) shall be implemented to restrict sensitive data access to authorized personnel only.

**4.2 Performance**

* Transactions should be processed within 2 seconds under standard load.
* The system must support concurrent operations from multiple users without delays or failures.

**4.3 Reliability**

* Transactions must be atomic; either fully completed or not executed at all (ACID-compliant).
* The system should guarantee data consistency, even during system failures or power loss.
* Transaction failures must automatically trigger rollback procedures to ensure no partial updates.

**4.4 Availability**

* The application should maintain 99.9% uptime with robust failover and recovery strategies.
* Regular backups shall be scheduled to safeguard against data loss.
* Disaster recovery procedures must be tested and documented.

**4.5 Compliance**

* The system must adhere to financial regulations and frameworks including:
  + KYC (Know Your Customer)
  + AML (Anti-Money Laundering)
  + Data retention and privacy standards (e.g., GDPR, RBI norms)

## 

## **5. Deliverable**

| **Deliverable** | **Description** |
| --- | --- |
| Requirement Specification Document | This document outlining functional, non-functional, and regulatory needs. |
| Use Case Diagram (Optional) | A diagram showing user interactions such as transfer, view log, etc. |
| System Workflow (Optional) | High-level overview of money transfer and audit logging processes. |
| Approval Sign-Off (Optional) | Stakeholder confirmation of documented requirements. |