Software Requirements Specification (SRS)

Bank Management System

Table of Contents

1. Introduction

1.1 Purpose:

The purpose of this document is to provide a comprehensive description of the Banking Management System (BMS) — a software solution aimed at simplifying and automating core banking operations. The document outlines the system's objectives, functional and non-functional requirements, user roles, and operational scope. It serves as a foundation for communication between all project stakeholders, including developers, testers, managers, clients, and end users.The Banking Management System is intended to handle various customer-related financial services such as account creation, money deposits, withdrawals, fund transfers, and transaction tracking. It will also support secure login features to protect user data and generate detailed reports for bank officials for administrative and regulatory purposes.This SRS ensures that the system is built in alignment with the clients' expectations and business goals, while also providing a clear and structured guideline for the development, testing, deployment, and maintenance of the software.

1.2 Intended Audience :

This document is intended for the following stakeholders:

1.2.1 Software Developers:

Developers will use this document to understand the exact requirements of the system so they can design, code, and implement the features accordingly. It provides both the functional details and technical constraints necessary for development.

1.2.2 Quality Assurance (QA) and Testers:

Testers will refer to this document to prepare test cases and validate that the system functions correctly and meets all specified requirements. It helps ensure that all modules are verified against the expectations.

1.2.3 Project Managers:

Project managers will use the document to oversee the development lifecycle, assign tasks, estimate timelines and resources, and monitor progress toward key milestones.

1.2.4 Bank Staff / End Users:

This group includes employees such as cashiers, accountants, and other banking personnel who will use the system in their daily operations. The document provides them with an understanding of how the system functions and how it will improve their workflows.

.

1.2.5 Clients / System Owners:

This includes the financial institution or individuals who have commissioned the development of the BMS. They will use the document to ensure that all business requirements are being addressed and implemented in the system.

1.3 Intended Use

The Banking Management System (BMS) is designed to be used in real-world banking environments to support a wide range of daily transactions and administrative functions. Its primary uses include:

1.3.1 .Customer Account Management :

The system will allow bank personnel to create new customer profiles, update personal information, and manage various types of accounts (e.g., savings, current, fixed deposits). This ensures centralized and structured data handling for all customer-related information. Financial Transactions (Deposits, Withdrawals, Transfers) Users will be able to perform core banking transactions: Deposits: Add money to their accounts either manually (bank staff) or digitally (online banking). Withdrawals: Withdraw funds securely through the system interface. Fund Transfers: Transfer money between accounts within the same bank or to external banks, with proper authorization and tracking.

1.3.2 Transaction History and Balance Inquiry:

Customers and bank officials can view detailed logs of all transactions, including date, time, amount, and transaction type. Account holders can also check real-time balance updates, promoting financial awareness and transparency.

1.3.3 Secure User Authentication and Login:

The system will enforce strong login protocols to ensure that only authorized individuals can access banking services. It may include features such as encrypted passwords, security questions, and two-factor authentication (2FA) to prevent unauthorized access and protect sensitive financial data.

1.4 Product Scope:

The Banking Management System (BMS) is a secure, scalable, and user-friendly software application designed to support the digital transformation of traditional banking processes. The system will streamline customer onboarding, manage account operations, facilitate secure financial transactions, and generate analytical reports for managerial insights.

The scope of the system includes:

- Customer account registration and profile management
- Deposit, withdrawal, and inter/intra-bank fund transfer services
- Real-time transaction history and account balance inquiry
- Role-based access control with secure login and data encryption
- Automated report generation for audits and performance monitoring

The BMS aims to enhance operational efficiency, reduce human errors, improve customer satisfaction, and maintain regulatory compliance through digitization of routine banking tasks.

1.5 Risk Definitions:

The following risks are considered during the development and deployment of the Banking Management System:

1.5.1 Data Breach Risk:

Sensitive customer data could be exposed due to system vulnerabilities or unauthorized access. This risk can be fixed through encryption, access controls, and regular audits.

1.5.2 Transaction Failure Risk:

Unexpected system failures during transactions could lead to financial discrepancies or customer dissatisfaction. Implementation of atomicity mechanisms and transactional integrity checks will reduce this risk.

1.5.3 Usability Risk:

If the interface is too complex or unintuitive, users may face difficulty operating the system. This can be fixed through UI/UX testing and incorporating user feedback.

1.5.4 Compliance Risk:

Failure to align with banking regulations can result in legal and financial penalties. The system will be built to support regulatory reporting and validation mechanisms.

2. Overall Description

2.1 User Classes and Characteristics:

2.1.1 Administrators:

- Have full access to manage system configurations, users, permissions, and data backups
- Responsible for maintaining system uptime and security updates
- Manage audit logs and ensure compliance with regulatory standards

2.1.2 Bank Employees (Tellers, Accountants, Managers):

- Perform daily transactions (deposits, withdrawals, transfers)
- Register new accounts, verify identity documents, and manage customer records
- Generate and review operational reports

2.1.3 Customers (End Users):

- Use the system (web or mobile interface) to check balances, transfer funds, and view transaction history
- Must authenticate using secure login mechanisms
- Receive notifications and digital receipts for all activities

2.2 User Needs

2.2.1 Bank Customer Needs

As a bank customer,

• I want to log in securely so that I can access my bank account without worrying about unauthorized access.

• I want to see my account details and balance anytime so that I can manage my money properly.

• I want to transfer money to other accounts easily so that I can send funds to friends, family, or other bank accounts.

• I want to check my transaction history so that I can keep track of my expenses and income.

• I want quick support from the bank if I face any problem so that my issues get resolved smoothly.

2.2.2 Bank Staff (Tellers/Officers) Needs:

As a bank employee.

• I want to manage customer information so that I can help customers with account updates or issues.

• I want to handle deposits and withdrawals quickly so that customers don't have to wait long.

• I want to check account balances and transaction history in real-time so that I can provide accurate information to customers.

2.2.3 Administrator Needs:

As an Admin,

• I want to have full control over the system so that I can manage users, accounts, and monitor system activities.

• I want to enforce security policies so that customer data stays safe and protected.

• I want to generate reports for regulatory purposes so that the bank stays compliant with laws.

2.3 Operating Environment

2.3.1 Hardware Environment:

As a user of the system, I can access it from:

• A desktop or laptop with basic configurations (e.g., Intel Core i3/i5, 8GB RAM)

• A mobile device like smartphones or tablets to use banking services on the go

2.3.2 Software Environment :

As a system user, I can access the Banking Management System through:

• Web browsers like Chrome, Firefox, Edge, or Safari • Operating systems like Windows 10/11, macOS, As an Admin or Developer, I can use MySQL or PostgreSQL databases for secure data storage.

2.3.3 Network and Security Environment:

As a user, I need a stable internet connection (minimum 5 Mbps) so that I can use the system without delays. As an Admin, I want the system to use SSL/TLS encryption so that user data stays secure during transactions. As a security team member, I can implement firewalls and multi-factor authentication to protect the system from threats

2.4 Constraints

2.4.1 Technical Constraints:

As a Developer, I can only use secure and approved technologies for building the system to avoid vulnerabilities. As an Admin, I can ensure that the system works with existing banking infrastructure to avoid compatibility issues.

2.4.2 Regulatory Constraints:

As a Bank employee, I must follow all government rules like KYC (Know Your Customer) and AML (Anti-Money Laundering) to stay compliant. As an Admin, I can generate reports to meet legal requirements.

2.4.3 Time Constraints :

As a Project Manager, I can set deadlines for each phase (design, development, testing) so that the project finishes on time.

2.4.4 Budget Constraints:

As an organization, we can only spend within the approved budget for development, deployment, and maintenance.

2.4.5 Resource Constraints:

As a Development Team, we need experienced people in banking software and cybersecurity to complete the project successfully. As an organization, we must ensure proper servers, network equipment, and tools are available to run the system smoothly.

3. Requirements

3.1 Functional Requirements

The functional requirements describe the specific behaviors and operations the Bank Management System must perform to support business needs:

3.1.1 Account Management:

- Allow authorized staff to create, update, deactivate, and close customer accounts.
- Validate customer identity using national ID or official documents during account creation.
- **3.1.2 Customer Information Management:**
- Store, retrieve, and update personal and financial data for each customer.
- Allow advanced search of customer records using account number, name, or NID.

3.1.3 Deposit and Withdrawal Handling:

- Enable customers to deposit or withdraw money via teller, ATM, or online portal.
- Validate account balance during withdrawal and log each transaction with a time stamp.

3.1.4 Fund Transfer:

- Support internal (within the same bank) and external (to other banks) fund transfers.
- Require OTP or 2FA confirmation for each transfer and generate confirmation receipt.

3.1.5 Transaction History and Balance Inquiry:

- Provide customers with real-time access to their balance and transaction logs.
- Include filters for date range, amount, and transaction type.

3.1.6 Authentication and Security:

- Require secure login using username and password.
- Support two-factor authentication and role-based access control.

3.1.7 Report Generation:

- Generate daily, weekly, and monthly financial reports for management.
- Export reports in PDF, Excel, or CSV format.

3.2 Non-Functional Requirements

These define system quality attributes, performance expectations, and operational standards:

3.2.1 Performance Requirements:

- The system should respond to user actions within 2 seconds under normal conditions.
- Capable of handling 1000+ concurrent users without performance degradation.

3.2.2 Security Requirements:

- All sensitive data must be encrypted during transmission and storage.
- Implement firewalls, intrusion detection systems, and regular vulnerability scans.

3.2.3 Usability Requirements:

- Provide an intuitive and responsive user interface accessible via desktop and mobile.
- The system should be operable by users with basic computer literacy.

3.2.4 Availability Requirements:

- Ensure 99.99% uptime with automated failover and recovery systems.
- Perform regular backups and system health checks.

3.2.5 Compatibility Requirements:

- The system should run on all major browsers (Chrome, Firefox, Edge) and OS (Windows, Linux, macOS).
- Support integration with SMS/email notification systems and third-party payment APIs.

3.2.6 Maintainability & Scalability:

- Modular architecture to allow updates and addition of features with minimal downtime.
- Easily scalable to accommodate future branches and growing user base.