**Aim- Prepare the SRS document in line with the IEEE standard format**

**2-use case diagram**

**Objective**

**Theory**

**IEEE Standard SRS Template**

1. **Introduction**
   1. Purpose

The purpose of this document is to define the requirements for the **Banking Management System (BMS)**. This system aims to provide a secure, efficient, and user-friendly platform for managing banking operations, including account management, transactions, customer support, and reporting.

* 1. Scope

The Banking Management System will:

* Allow customers to create and manage bank accounts (savings, current, fixed deposit).
* Enable customers to perform transactions (deposit, withdrawal, fund transfer).
* Provide administrative functions for bank staff to manage customer accounts and transactions.
* Generate reports for transactions, account balances, and customer details.
* Ensure secure access through authentication and authorization mechanisms.

The system will be accessible via a **web-based interface** and a **mobile application**.

* 1. Definitions, acronyms & abbreviations
* **BMS**: Banking Management System
* **API**: Application Programming Interface
* **UI**: User Interface
* **DBMS**: Database Management System
* **KYC**: Know Your Customer
* **OTP**: One-Time Password
* **SSL**: Secure Sockets Layer
* **ATM**: Automated Teller Machin
  1. References
* IEEE 830-1998: IEEE Recommended Practice for Software Requirements Specifications.
* Banking Regulations and Compliance Guidelines.
* Data Protection and Privacy Laws (e.g., GDPR).
  1. Overview

This document is organized into four main sections:

1. **Introduction**: Provides an overview of the system.
2. **Overall Description**: Describes the system's context, functionality, and user characteristics.
3. **Specific Requirements**: Details the functional and non-functional requirements.
4. **Supporting Information**: Includes appendices and additional resources.

**2. Overall description**

2.1.Product perspective

The Banking Management System is a standalone system that integrates with external banking services (e.g., payment gateways, credit bureaus).

2.1.1. System interfaces

* **External Systems**: Payment gateways, credit card networks, and regulatory systems.
* **Internal Systems**: Database, authentication server, and reporting tools.

2.1.2. User interfaces

* **Customer Interface**: Web and mobile app for account management and transactions.
* **Admin Interface**: Web-based dashboard for managing accounts, transactions, and reports

2.1.3. Hardware interfaces

* Supports interaction with hardware devices such as card readers, biometric scanners, and ATMs.

2.1.4. Software interfaces

* Database management system (MySQL, PostgreSQL, or Oracle).
* APIs for third-party integrations.

2.1.5. Communications interfaces

* HTTPS for secure data transmission.
* Email and SMS notification services.

2.1.6. Memory constraints

* Scalable cloud storage for transaction history.
* Efficient caching mechanisms for quick retrieval.

2.1.7. Operations

* Customer registration, account management, and financial transactions.
* Loan and credit management.

2.1.8. Site adaptation requirements

* Configurable for different banking institutions.
* Adaptable for multiple languages and currencies.

2.2. Product functions

* **Customer Functions**:
  + Account creation and management.
  + Deposit, withdrawal, and fund transfer.
  + View transaction history and account balance.
* **Admin Functions**:
  + Manage customer accounts.
  + Approve loans and credit requests.
  + Generate reports and analytics.

2.3. User characteristics

* Customers: Basic computer literacy.
* Bank Staff: Trained to manage accounts and transactions.
* Administrators: Advanced technical knowledge.

2.4. Constraints

* Compliance with banking regulations and data protection laws.
* High-security requirements for handling sensitive financial data

2.5. Assumptions and dependencies

* Reliable internet connectivity for users.
* Integration with third-party services (e.g., payment gateways).

2.6. Apportioning of requirements

* Phase 1: Core functionality (account management, transactions).
* Phase 2: Advanced features (loan management, reporting).

**3. Specific Requirements**

3.1 External interface requirements

3.1.1 User interfaces

* **Customer UI**: Intuitive design with options for account management and transactions.
* **Admin UI**: Dashboard with tools for managing accounts and generating reports.

3.1.2 Hardware interfaces

* ATM integration for withdrawals and deposits.
* Biometric authentication for security.

• Integration with card readers and biometric devices for secure login.

3.1.3 Software interfaces

* API support for third-party banking services.
* Database connections for transaction logging.
* APIs for payment processing, KYC verification, and credit scoring
  + 1. Communication interfaces
  + Secure communication via HTTPS and SSL/TLS.

3.2 Specific requirements

* + 1. Sequence diagrams
* Sequence diagrams for key processes (e.g., fund transfer, account creation).

3.2.2 Classes for classification of specific requirements

* **Customer Class**: Attributes include name, account number, balance.
* **Transaction Class**: Attributes include transaction ID, amount, date.

3.3 Performance requirements

* System response time: < 2 seconds for most operations.
* Support for up to 10,000 concurrent users.

3.4 Design constraints

* Use of secure coding practices to prevent vulnerabilities.
* Compliance with accessibility standards (e.g., WCAG).

3.5 Software system attributes

**3.5.1 Reliability**

* System uptime: 99.9%.
* Data backup and recovery mechanisms.

**3.5.2 Availability**

* 24/7 availability with minimal downtime.

**3.5.3 Security**

* Multi-factor authentication (MFA) for login.
* Encryption of sensitive data at rest and in transit.
  + 1. **Maintainability**
    2. Modular design for easy updates and maintenance.

3.6 Other requirements

* Support for multi-currency transactions.
* Integration with mobile banking apps

**4. Supporting information**

4.1 Table of contents and index

4.2 Appendixes