**Software Requirements Specification**

**for**

**Banking System**

**Version 1.0**

**KMIT**

**18-10-2024**

**Group Members:**

1)D Siddharth – 22bd1a056m

2)D Sanjay – 22bd1a056l

3)Parjan – 22bd1a057p

4)Ambhikesh – 22bd1a0564

5)Shireesh – 22bd1a057f

6)Uday D – 22bd1a056j

**s1. Introduction**

* 1. **Purpose**

This document defines the software requirements for the **Banking System**, covering core functionalities such as account management, transactions, customer services, and security protocols. This SRS applies to version 1.0 of the banking system software and focuses on delivering secure, user-friendly financial services.

* 1. **Document Conventions**
* **Bold**: Primary system components
* *Italics*: Examples
* Priority levels are indicated as **High**, **Medium**, or **Low** for individual requirements.
  1. **Intended Audience and Reading Suggestions**
* **Developers**: For understanding functional and nonfunctional requirements.
* **Project Managers**: For scope and progress tracking.
* **Testers**: To create test cases based on system features and requirements.
* **End Users**: To understand available features.

Start by reviewing the **Product Scope**, then proceed through specific sections relevant to each audience, such as **System Features** for developers and testers, and **Performance Requirements** for project managers.

* 1. **Definitions, Acronyms, and Abbreviations**
* **BMS**: Bank Management System
* **KYC**: Know Your Customer
* **DBMS**: Database Management System

**API**: Application Programming Interface

* 1. **Product Scope**

The **Banking System** aims to provide comprehensive banking solutions, including account management, secure financial transactions, and reporting features. The system will enhance customer experiences by providing online access to accounts and implementing strict security measures. It aligns with the bank's strategic goal of offering a modern, secure, and user-friendly platform.

* 1. **References**
* [ISO/IEC 27001:2013 Information Security Standard]
* [Company Policy Document on Financial Transactions]
* [User Interface Guidelines v2.0]

**2. Overall Description**

**2.1 Product Perspective**

The **Banking System** is a new product, designed to replace the bank’s legacy system. It will integrate with the bank's existing customer databases and financial reporting tools. The system includes mobile and web applications that interact through secure APIs.

**2.2 Product Functions**

* **Account Management**: Create, modify, and close accounts.
* **Transaction Processing**: Deposits, Withdrawals, fund transfers, and bill payments.
* **Customer Services**: Loan Management, Balance inquiries, statement generation, and support requests.
* **Security**: User authentication, encryption, fraud detection.

**2.3 User Classes and Characteristics**

* **Customers**: General account holders with basic to advanced financial knowledge.
* **Bank Employees**: Administrators and customer service representatives with access to internal tools.
* **System Administrators**: Manage software updates, security, and database integrity.
* **Third Party** : Visa cards, Master card

**2.4 Operating Environment**

* **Operating System**: Windows 10+, macOS, Linux
* **Software Requirements**: Web browsers (Chrome, Firefox), Mobile platforms (iOS, Android)
* **Network**: Internet connection for web-based access, secure VPN for internal operations.

**2.5 Design and Implementation Constraints**

* Compliance with **PCI-DSS**.
* **Database**: Oracle 19c.
* **Languages**: Java for backend, JavaScript (React) for frontend.

**2.6 User Documentation**

* Online help guides.
* User manuals for banking personnel.
* FAQs for customers.

**2.7 Assumptions and Dependencies**

* The system relies on the existing customer database.
* Integration with third-party payment gateways is required.

**3. External Interface Requirements**

**3.1 User Interfaces**

* Web-based dashboard for customers.
* Mobile app for Android/iOS with touch and face authentication.
* Admin panel for bank staff

.

**3.2 Hardware Interfaces**

* Mobile devices: Fingerprint and face recognition for authentication
* Servers should have the capacity to handle large volumes of data transactions.
* ATM machines for real-time account interactions.

**3.3 Software Interfaces**

# Integration with KYC verification systems (e.g., Aadhaar or SSN validation).

* The system must interact with external APIs (payment gateways, third-party blanks).

**3.4 Communications Interfaces**

* **HTTPS** for all communication between the client and server.
* Email/SMS notifications for transaction alerts.

# Mobile applications should use APIs for interaction with the backend, with JSON data formats.

**4. System Features**

**4.1 Account Management**

**4.1.1 Description and Priority**

* **Priority: High**  
  Users must be able to create new bank accounts, update account details (e.g., contact information, address), and view their account balance and transaction history. These are core functionalities critical to the system's operation.
* **Priority: Medium**  
  Features such as the ability to link external accounts (e.g., accounts from other banks) or set up joint accounts could be included. These features enhance user convenience but are not immediately essential for basic account management.
* **Priority: Low**  
  A feature allowing users to set up account preferences (e.g., custom account names, account categorization, or notification settings) could be implemented. These are additional options that improve user experience but are not crucial to the core account management functionality.

**4.1.2 Stimulus/Response Sequences**

* User selects “Create Account”; system prompts for details.
* User inputs account details; system validates and creates an account.

**4.1.3 Functional Requirements**

* Account Creation: The system shall allow the creation of new accounts.
* Validation (Sign-Up): The system shall validate user data before account creation.
* Creation via Email: The system shall notify users of account creation via email.

**4.2 Transaction Management**

**4.2.1 Description and Priority**

This feature allows users to perform financial transactions such as fund transfers and bill payments.

* **Priority: High**Users need to execute core transactions like fund transfers and bill payments reliably and securely.
* Enhancements such as scheduled transactions or automated bill payments could be included.
* **Priority: Medium**These features improve user convenience by allowing automation and scheduling of regular payments.
* A feature allowing users to set transaction preferences (e.g., notifications for every bill payment or fund transfer) could be implemented.
* **Priority: Low**This feature enhances customization and user control but is not critical to core functionality.

**4.2.2 Stimulus/Response Sequences**

* **Fund Transfer**
  + User selects "Transfer Funds"; system requests source and destination account details.
  + User confirms the transfer; system processes and sends confirmation.

**4.2.3 Functional Requirements**

* **Bank Transfers**

The system shall support intra-bank and inter-bank fund transfers.

* **Bill Payments**

The system shall allow users to pay bills to registered vendors or service providers.

* **Notifications**

The system shall send email/SMS notifications for each transaction.

* **Detection and Recovery**

The system shall apply real-time fraud detection and enable recovery for failed or suspicious transactions**.**

**4.3 Loan Management**

**4.3.1 Description and Priority**

This feature allows users to manage loans, including applying for loans, viewing loan details, and making loan payments.

**Priority: High**Users must be able to apply for and manage loans, including viewing outstanding balances and payment schedules.

Enhancements such as automatic loan payment scheduling could be included.

**Priority: Medium**These features will allow users to automate loan payments or receive reminders for upcoming payments.

A feature allowing users to track loan application progress or submit additional documents could be implemented.

**Priority: Low**This feature would enhance user convenience but is not essential for initial functionality.

**4.3.2 Stimulus/Response Sequences**

* **Loan Application**
  + User selects "Apply for Loan"; system requests details such as loan type, amount, and tenure.
  + User submits the application; system processes and returns either an approval, rejection, or request for further information.
* **Loan Payment**
  + User selects "Pay Loan"; system requests loan details and payment amount.
  + User confirms the payment; system processes and sends confirmation.

**4.3.3 Functional Requirements**

* **Loan Applications**

The system shall allow users to apply for various types of loans (e.g., personal loans, home loans, auto loans).

* **Loan Management**

The system shall display loan details, including outstanding balances, payment schedules, and interest rates.

* **Loan Payments**

The system shall allow users to make loan payments and provide receipts for successful transactions.

* **Loan Status Tracking**

The system shall allow users to track the status of their loan applications, including approvals, rejections, or requests for additional documentation.

* **Notifications**

The system shall send email/SMS notifications for each loan-related event, such as approval, payment reminders, and overdue notices.

**5. Other Nonfunctional Requirements**

**5.1 Performance Requirements**

* The system shall handle up to 100,000 transactions per second.
* All user interactions should have a response time of less than 2 seconds.

**5.2 Safety Requirements**

* The system must ensure that transactions are irreversible after confirmation.
* Critical data backups must be performed daily.

**5.3 Security Requirements**

* All data must be encrypted during transmission and storage.
* Multi-factor authentication is required for all user logins.

**5.4 Software Quality Attributes**

* **Reliability**: The system should maintain 99.99% uptime.
* **Usability**: The system should be intuitive and require minimal training.

**5.5 Business Rules**

* Only users with verified identities can perform transactions above a defined threshold.
* Account closures must be approved by an administrator.

**6. Other Requirements**

* The system should be localizable for multiple languages.
* Compliance with **GDPR** for handling customer data.

**Appendix A: Glossary**

* **ATM**: Automated Teller Machine
* **PCI-DSS**: Payment Card Industry Data Security Standard
* **GDPR**: General Data Protection Regulation

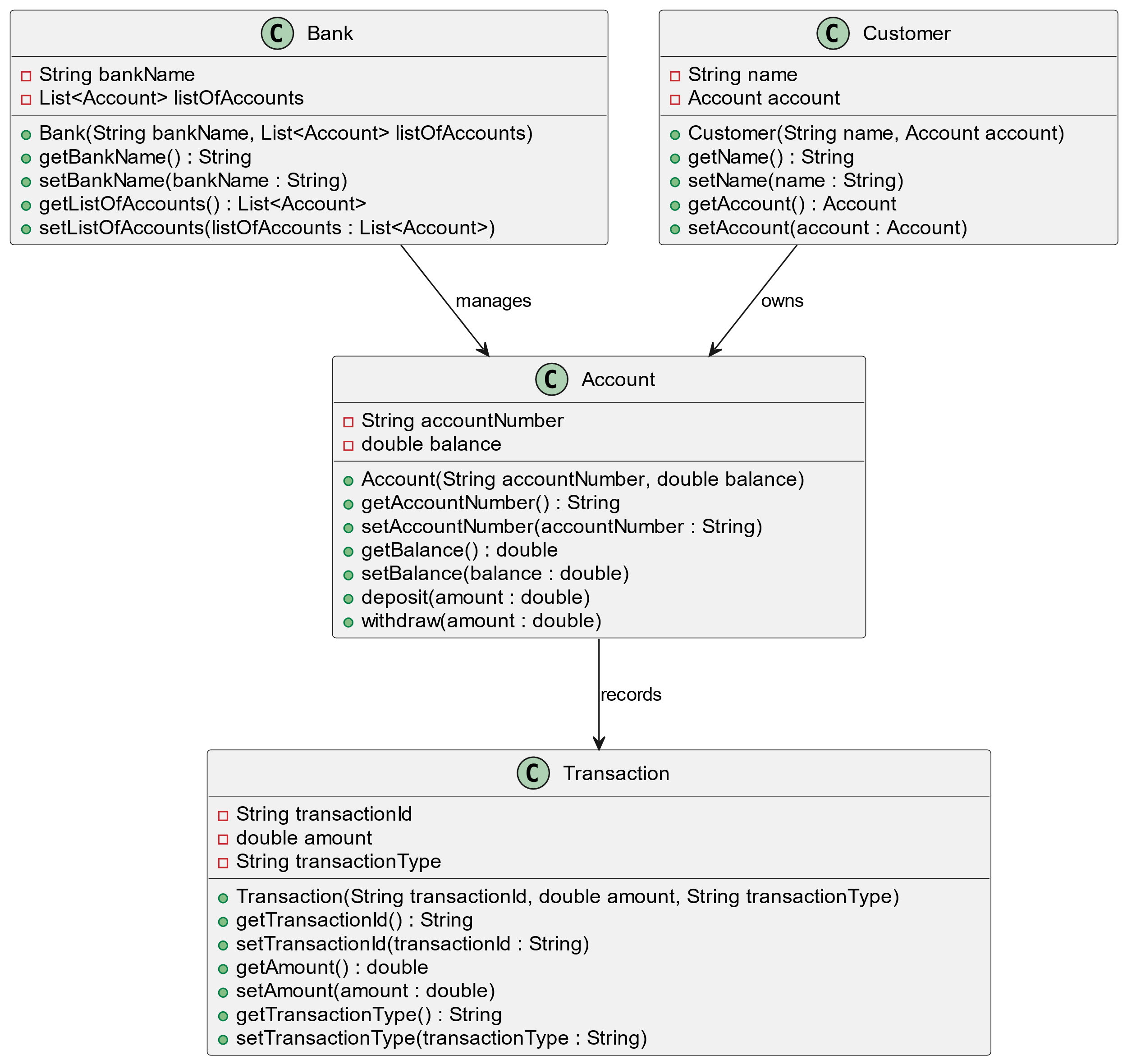
**Appendix B: Analysis Models**

* Data Flow Diagram for the transaction process.

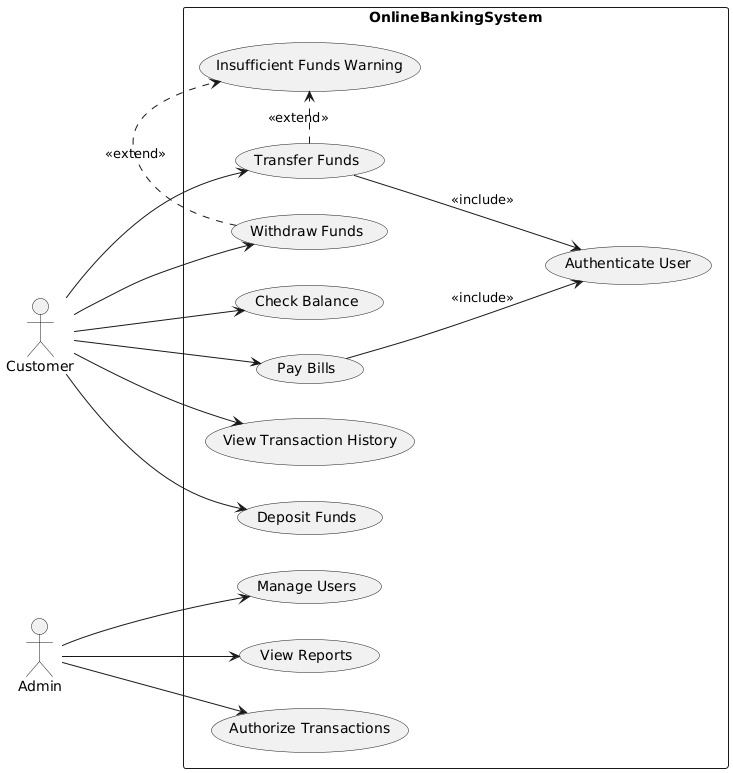
**Appendix C: To Be Determined List**

* TBD: Integration timeline for third-party payment gateways.

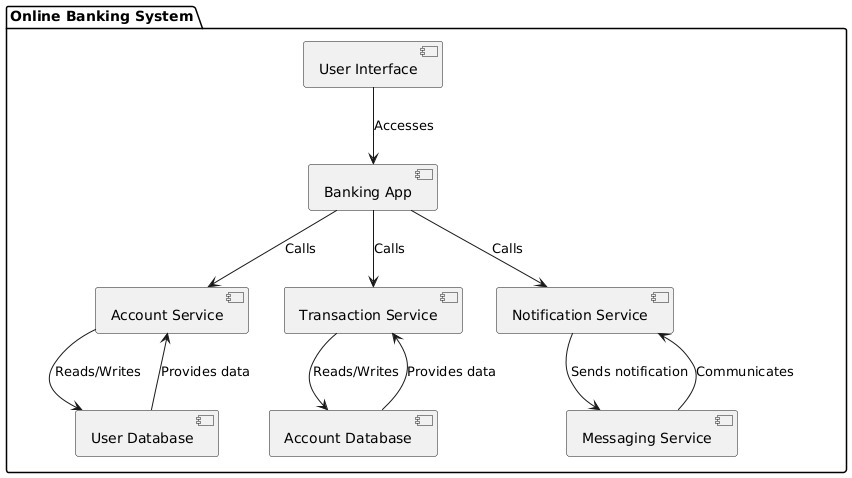
Class Diagram:



Use Case Diagram:



Component Diagram:



Sequence Diagram:

