PRACTICAL – 3

AIM: Study software requirement engineering. Student should include SRS document for current semester project.

SRS (Software requirement specification)

NAME OF THE SYSTEM: *Banking system*

TABLE OF CONTENTS:

1. Introduction

1.1 Purpose

*The purpose of this document is to define the software requirements for the development of a **Banking System**. This system will allow customers to manage their banking accounts, view transactions, transfer funds, pay bills, and access other banking services through a secure, user-friendly interface.*

1.2 Scope

This system will include the following core functionalities:

- *Customer Registration and Authentication*
- *Account Management (view balance, transaction history)*
- *Fund Transfer (internal, external, bill payments)*
- *Loan Management*
- *Notification Services (alerts, reminders)*
- *Security Features (multi-factor authentication, encryption)*

This system will be designed to be used by retail banking customers, bank employees, and administrators. It will be accessible via web and mobile platforms.

1.3 Definitions, Acronyms, and Abbreviations

- ***Customer**: A person who holds an account with the bank.*
- ***Account**: A banking account such as a savings or checking account.*
- ***Transaction**: Any movement of money into or out of an account (deposit, withdrawal, transfer).*

- ***MFA**: Multi-factor authentication.*
- ***API**: Application Programming Interface.*
- ***KYC**: Know Your Customer, a standard for verifying customer identity.*
- ***OTP**: One-Time Password.*

2. Overall Description

2.1 Product Perspective

The banking system will serve as an interface between customers and the bank's backend financial services. It will interact with various APIs to fetch live data, provide transaction updates, and handle payment processing. The system will have multiple modules: customer registration, account management, transactions, bill payments, notifications, and user management.

2.2 Product Functions

The main functions of the Banking System include:

- ***Customer Authentication**: Secure login, registration, and identity verification.*
- ***Account Management**: View balances, statements, and transaction history.*
- ***Fund Transfer**: Internal transfers (within the bank) and external transfers (to other banks or accounts).*
- ***Bill Payments**: Pay utility bills, credit card bills, loan payments, etc.*
- ***Loan Management**: Apply for loans, track loan repayments.*
- ***Notifications**: Alerts for transactions, low balance, due payments, etc.*
- ***Admin Interface**: Bank staff can manage user accounts, perform audits, and approve loan applications.*

2.3 User Characteristics

- ***Customers**: Regular users who need to manage their personal accounts, make transfers, pay bills, and receive alerts.*
- ***Bank Employees**: Bank staff who manage customer accounts, verify transactions, approve loans, etc.*
- ***Administrators**: Bank administrators who monitor system performance, enforce security policies, and oversee the operations.*

2.4 Constraints

- *The system must comply with financial regulations (e.g., PCI-DSS, GDPR, SOX).*
- *It must be secure against fraud and cyber threats.*
- *The system must handle a large number of simultaneous users and transactions.*
- *It must support multiple devices (desktop, mobile) with responsive design.*
- *The system must be scalable to support future expansion (more customers, features).*

2.5 Assumptions and Dependencies

- *The system will rely on third-party APIs for payment gateways, fraud detection, and other services.*
- *The bank has existing infrastructure and backend services for handling financial transactions.*

- *Customers will have access to the internet for using the system.*

---

3. System Features

3.1 User Registration and Authentication

***Description**: Allows customers to register, login, and authenticate securely.*

- ***Functional Requirements**:*
    1. *Users can register by providing basic details like name, email, phone number, and address.*
    2. *Users must verify their email or phone number (OTP-based verification).*
    3. *Customers can log in using a username/password or via multi-factor authentication (MFA).*
    4. *Passwords should be securely stored using encryption.*
- ***Non-Functional Requirements**:*
    1. *Passwords must be encrypted using strong algorithms (e.g., bcrypt).*
    2. *The system should support CAPTCHA or other anti-bot measures during registration.*
    3. *Login sessions must time out after a specified period of inactivity.*

3.2 Account Management

***Description**: Customers can view and manage their bank accounts, including checking balances and transaction history.*

- ***Functional Requirements**:*
    1. *Users can view their account balance and details.*
    2. *Users can view their transaction history (deposits, withdrawals, transfers).*
    3. *Users can request account statements for a specified period.*
- ***Non-Functional Requirements**:*
    1. *The system should provide real-time balance updates.*
    2. *Transaction history should be searchable by date and transaction type.*

3.3 Fund Transfers

***Description**: Users can transfer funds between accounts, to other banks, and pay bills.*

- ***Functional Requirements**:*
    1. *Users can transfer money between their own accounts.*
    2. *Users can send money to external bank accounts (using IFSC or SWIFT codes).*
    3. *Users can pay utility bills such as electricity, water, and mobile bills.*
    4. *Transfers will require multi-factor authentication (OTP sent via SMS/email).*
- ***Non-Functional Requirements**:*
    1. *The system must confirm transfer success or failure in real time.*
    2. *Transfers should be processed with minimal delay.*
    3. *The system should support large transaction volumes without slowing down.*

3.4 Loan Management

***Description**: Users can apply for personal loans, view loan status, and track repayments.*

- ***Functional Requirements**:*
    1. *Users can apply for loans by providing necessary documentation and financial details.*
    2. *Users can view the status of their loan applications.*
    3. *Loan repayments can be tracked and updated in real-time.*
- ***Non-Functional Requirements**:*
    1. *Loan processing should be fast and efficient, with feedback to the customer on the status of their application.*
    2. *The system should be integrated with credit scoring and risk assessment tools.*
    3. *Loan records must be securely stored and accessible only to authorized personnel.*

3.5 Notifications and Alerts

***Description**: The system will send notifications to customers about important events.*

- ***Functional Requirements**:*
    1. *Notifications will be sent for successful transactions, low balance warnings, bill payments due, etc.*
    2. *Customers can choose to receive notifications via SMS, email, or in-app alerts.*
- ***Non-Functional Requirements**:*
    1. *Notifications must be delivered in real-time.*
    2. *The system should handle high volumes of notifications during peak times.*

3.6 Security Features

***Description**: Ensure the system is secure against unauthorized access and fraudulent activities.*

- ***Functional Requirements**:*
    1. *Multi-factor authentication (MFA) will be required for accessing sensitive features (e.g., fund transfers).*
    2. *The system will log and monitor suspicious activities, like multiple failed login attempts or large transfers.*
- ***Non-Functional Requirements**:*
    1. *All sensitive data must be encrypted at rest and in transit using strong encryption algorithms (e.g., AES-256).*
    2. *The system should comply with security standards like PCI-DSS for handling payment data.*
    3. *The system must provide role-based access control (RBAC) to ensure proper authorization for different user roles.*

---

4. External Interface Requirements

4.1 User Interfaces

- *Web-based interface for customers, employees, and administrators.*
- *Mobile apps for iOS and Android.*
- *Responsive design to ensure usability across devices.*

4.2 Hardware Interfaces

- *The system will operate on cloud infrastructure and does not require specific hardware for end users.*

4.3 Software Interfaces

- *Payment gateway APIs for transferring funds to external banks.*
- *SMS/Email services for sending OTPs and notifications.*

4.4 Communication Interfaces

- *REST APIs for communication between the banking system and external services (e.g., payment gateways, fraud detection).*

---

5. Non-Functional Requirements

5.1 Performance Requirements

- *The system should support at least 100,000 concurrent users during peak times.*
- *Transactions must be processed in under 5 seconds.*

5.2 Reliability & Availability

- *The system must have 99.9% uptime.*
- *The system should have automatic failover in case of hardware or software failure.*

5.3 Scalability

- *The system should be designed to scale horizontally to accommodate increasing users and transaction volumes.*

5.4 Usability

- *The system should be user-friendly and intuitive, with easy navigation for customers of all ages.*

5.5 Security

- *Implement robust security measures such as encryption, firewalls, and intrusion detection systems.*

- *Data backups should be performed daily and stored securely.*

ad