

TASK -2

Software Requirement Specification (SRS) Document

Online Banking System

1. Introduction

1.1 Purpose

The purpose of this document is to define the requirements for the Online Banking System. It provides a detailed description of the functional and non-functional requirements. The document serves as a reference for developers, testers, and stakeholders to ensure that the final product aligns with the client's expectations and meets industry standards for online financial systems.

1.2 Scope

The Online Banking System allows customers to perform various banking operations online, such as viewing account balances, transferring funds, paying bills, managing beneficiaries, and requesting bank services without visiting a branch. The system will be accessible through web and mobile interfaces for both customers and administrators, offering high security and ease of use.

1.3 Definitions, Acronyms, and Abbreviations

Term	Description
Customer	Registered user who performs banking operations online.
Admin	Bank personnel responsible for managing accounts, transactions, and system operations.
Database	Stores customer details, account information, and transaction records.
OTP	One-Time Password used for secure authentication.

1.4 References

- i. www.rbi.org.in
- ii. www.wikipedia.org
- iii. IEEE Software Requirements Specification Std. 30-1993

1.5 Overview

- Chapter 1 describes the purpose and scope.
- Chapter 2 discusses the system's overall functionalities and constraints.
- Chapter 3 defines specific requirements to design and implement the system.

2. System Description

2.1 Product Perspective

The Online Banking System replaces traditional manual banking processes by offering an automated digital platform. It enables customers to access banking services anytime, anywhere. The system ensures secure data storage, fast transactions, and user-friendly access through the web and mobile applications. Administrators can monitor transactions, manage user accounts, and generate reports in real time.

2.2 Product Functions

There are two main users of this system:

1. Customer
2. Administrator

System Functions Include:

- User authentication and registration.
- Viewing account details and transaction history.
- Online fund transfer between accounts and banks.
- Bill payment and loan management.
- Admin dashboard for transaction monitoring and fraud detection.
- Secure communication using encryption and OTP-based verification.

2.3 User Functions

a) Administrator:

- View, verify, and manage user accounts.
- Approve new registrations and loan requests.
- Monitor suspicious transactions and generate financial reports.
- Manage service configurations such as interest rates or transaction limits.

b) Customer:

- Register and log in to the banking portal.
- View account balance and recent transactions.
- Transfer funds and pay utility bills.
- Manage beneficiaries and update personal details.
- Download account statements and request cheque books or cards.

2.4 System Constraints

- **Compatibility:** Must support major browsers and mobile OS (Chrome, Firefox, Android, iOS).
- **Security:** Must comply with RBI guidelines and industry encryption standards.
- **Budget:** Should operate cost-effectively with limited infrastructure.
- **Connectivity:** Requires a stable internet connection for all operations.
- **Hardware:** Should function on standard hardware with moderate processing power.
- **Compliance:** Must follow data privacy laws and PCI DSS standards.
- **Backup:** Regular automatic data backup should be maintained.

2.5 System Dependencies

- Relies on a robust database (e.g., MySQL or Oracle) for secure storage.
- Network infrastructure for real-time transaction processing.
- Integration with third-party APIs (e.g., UPI, payment gateways).
- Encryption libraries for secure communication.
- SMS and email APIs for OTP and notifications.

3. Specific System Requirements

3.1 Functional Requirements

1. **User Management** – Users must be able to register and securely log in using OTP-based authentication.
2. **Account Management** – The system should display account balance, transaction history, and profile details.
3. **Funds Transfer** – Enable fund transfers between same-bank and inter-bank accounts with confirmation alerts.
4. **Bill Payment** – Allow customers to pay utility bills and schedule recurring payments.
5. **Loan and Card Requests** – Customers can apply for loans, debit/credit cards, and track their status.
6. **Admin Management** – Admins can manage users, approve transactions, and view detailed reports.
7. **Notification System** – Send alerts via email/SMS for transactions, logins, and suspicious activity.

3.2 Non-Functional Requirements

- **Performance:** The system must support 1000+ users concurrently.

- Reliability: System uptime should be at least 99.9%.
- Usability: Interface should be simple, responsive, and accessible for all users.
- Security: Data encryption (SSL/TLS), two-factor authentication, and audit logs must be implemented.
- Maintainability: System should be modular and easily upgradable.
- Backup & Recovery: Automatic daily backups and disaster recovery mechanisms.

3.3 External Interfaces

3.3.1 User Interface:

The homepage should display quick access to account overview, fund transfer, bill payment, and support. The dashboard should present real-time balance, recent transactions, and personalized alerts.

3.3.2 Hardware Interface:

Compatible with desktops, laptops, tablets, and smartphones. Requires stable internet connectivity.

3.3.3 Software Interface:

- Frontend: Angular / React
- Backend: Java / Node.js
- Database: MySQL / Oracle
- Other Tools: MS Office, RESTful APIs

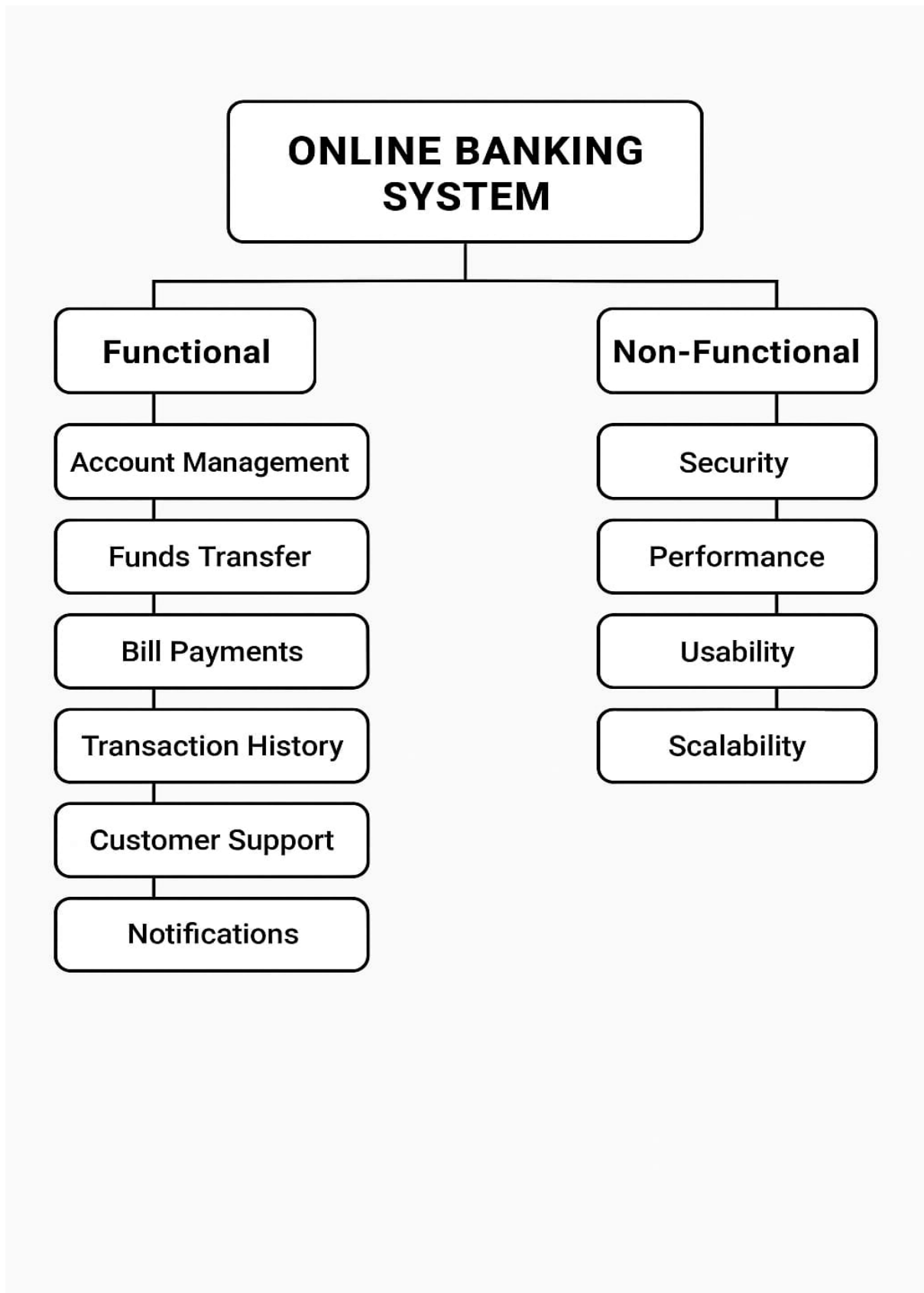
3.3.4 Communication Interface:

Secure HTTPS protocols for data transfer. SMS and email notifications for authentication and updates.

4. Appendices

Appendix A: Definitions of key terms

Appendix B: Database Schema for Account, Transaction, and User tables



Flow Chart Diagram

Result

Thus, the Software Requirement Specification Document for Online Banking System has been prepared successfully.