Title:- Transaction Processing System

1.Introduction

A Transaction Processing System is a software application that manages and records transactions in a systematic manner. These systems are integral to a wide range of industries, including banking, retail, and e-commerce, where they handle tasks such as order processing, payment handling, and data management.

The primary goal of this project is to create a robust TPS that can process and manage transactions in real-time, ensuring data integrity and consistency. Key features of the system include transaction recording, error handling, concurrency control, and secure data management. The use of Java provides a strong platform for developing scalable and reliable solutions due to its robust libraries, object-oriented nature, and cross-platform capabilities.

2. Purpose

This SRS covers the requirements for the development of a TPS that supports multiple types of transactions, including but not limited to deposits, withdrawals, transfers, and payments. The system will be designed to handle high volumes of transactions and ensure data integrity, security, and compliance with relevant financial regulations.

2.1 Definitions, Acronyms, and Abbreviations

- *TPS: Transaction Processing System*
- *API: Application Programming Interface*
- *GUI: Graphical User Interface*
- *DBMS: Database Management System*

2.2 Product Perspective

The TPS will be a standalone system that can be integrated with other financial software and systems via APIs. It will be built using Java and a relational DBMS. The system will have both a web-based GUI for user interaction and a backend server for processing transactions.

2.3 Product Functions

- *User authentication and authorization*
- *Transaction initiation and processing*
- *Account management*

- *Transaction history and reporting*
- *Error handling and recovery*
- *System monitoring and auditing*

2.4 Operating Environment

- *Server: Linux-based server with Java Runtime Environment (JRE) 8 or higher*
- *Client: Web browser (Chrome, Firefox, Safari, Edge)*
- *Database: Relational DBMS (e.g., MySQL, PostgreSQL)*

3. System Features

3.1 User Authentication and Authorization

- *Secure login and logout*
- *Role-based access control*

3.2 Transaction Management

- *Initiate transactions (deposits, withdrawals, transfers, payments)*
- *Verify and process transactions*
- *Real-time transaction updates*

3.3 Account Management

- *Create, update, and delete accounts*
- *View account balances and transaction history*

4. External Interface Requirements

4.1 User Interfaces

- *Web-based GUI for user interaction*
- *Admin dashboard for system monitoring*

4.2 Hardware Interfaces

- *Server hardware specifications (CPU, RAM, storage)*
- *Network interface requirements*

4.3 Software Interfaces

- *APIs for integration with external systems*

- *Database connectivity (JDBC)*

4.4 Communication Interfaces

- *Secure communication protocols (HTTPS, SSL/TLS)*
- *Notification systems (email, SMS)*

5. System Requirements

5.1 User Authentication

- *The system shall provide secure login and logout functionality.*
- *The system shall support role-based access control.*

5.2 Transaction Processing

- *The system shall support the initiation, verification, and processing of transactions.*
- *The system shall update transaction status in real-time.*

5.3 Account Management

- *The system shall allow users to create, update, and delete accounts.*
- *The system shall provide account balance and transaction history views.*

6 Non-functional Requirements

6.1 Performance

- *The system shall handle up to 10,000 transactions per second.*
- *The system shall provide a response time of less than 2 seconds for transaction processing.*

6.2 Security

- *The system shall use encryption for data transmission.*
- *The system shall comply with relevant financial regulations.*

6.3 Reliability

- *The system shall have 99.99% uptime.*
- *The system shall provide mechanisms for error handling and recovery.*

7. Other Non-functional Requirements

7.1 Performance Requirements

- *The system shall handle peak loads without performance degradation.*

- *The system shall scale horizontally to accommodate increased transaction volumes.*

7.2 Safety Requirements

- *The system shall ensure data integrity during transaction processing.*
- *The system shall provide backup and recovery mechanisms.*

7.3 Security Requirements

- *The system shall implement multi-factor authentication.*
- *The system shall audit and log all transactions and access attempts.*

7.4 Software Quality Attributes

- *Usability: The system shall provide an intuitive user interface.*
- *Maintainability: The system shall use modular design principles.*
- *Scalability: The system shall support scalability to handle growing transaction volumes.*

Conclusion

The transaction processing system project in Java successfully meets its objectives of managing transactions and maintaining data integrity. The system's design leverages Java technologies and adheres to best practices for performance and reliability. Challenges encountered were effectively addressed, and the system performs well under load. Future enhancements could include scalability improvements and additional features. Overall, the project is a solid implementation with potential for real-world application.