**Software Requirements Specification (SRS) for Online Banking System**

**1. Introduction**

**1.1 Purpose**

This document outlines the software requirements for the Online Banking System. It details the functionalities, business rules, and validation criteria required to ensure proper operation and user experience.

**1.2 Scope**

The Online Banking System will allow users to create accounts, deposit and withdraw funds, and transfer money between accounts. The system aims to provide a simple and user-friendly interface while ensuring data integrity and security.

**1.3 Definitions, Acronyms, and Abbreviations**

* **SRS**: Software Requirements Specification
* **UI**: User Interface
* **API**: Application Programming Interface
* **Balance**: The amount of money in the user's account.

**2. Overall Description**

**2.1 Product Perspective**

The Online Banking System is a web-based application developed using HTML, CSS, and JavaScript. It operates on modern web browsers and is designed for individual users managing their bank accounts.

**2.2 Product Functions**

The main functions of the system include:

* User account creation
* Deposit funds
* Withdraw funds
* Fund transfer between accounts
* Display account balance

**2.3 User Classes and Characteristics**

* **Customers**: Individuals who will use the system to manage their bank accounts. Users will have basic technical skills to navigate a web application.

**2.4 Operating Environment**

The application will run in any modern web browser, including Chrome, Firefox, and Safari, on desktop and mobile devices.

**2.5 Design and Implementation Constraints**

* The system will be developed using HTML, CSS, and JavaScript.
* User data will be stored in the browser's memory (for simplicity) and will not persist after the session ends.

**3. Functional Requirements**

**3.1 User Account Management**

* **Create Account**:
  + **Input**: Account name, initial deposit (minimum $1000).
  + **Validation**:
    - Account name must be a non-empty string.
    - Initial deposit must be at least $1000.
  + **Output**: Confirmation of account creation and the initial balance.

**3.2 Deposit Funds**

* **Input**: Amount to deposit.
* **Validation**:
  + Deposit amount must be greater than zero.
* **Output**: Updated account balance and confirmation of the successful deposit.

**3.3 Withdraw Funds**

* **Input**: Amount to withdraw.
* **Validation**:
  + Withdrawal amount must be greater than zero.
  + Withdrawal amount must be less than or equal to the current account balance.
* **Output**: Updated account balance and confirmation of the successful withdrawal.

**3.4 Fund Transfer**

* **Input**: Recipient account name and amount to transfer.
* **Validation**:
  + Transfer amount must be greater than zero.
  + Transfer amount must be less than or equal to the current account balance.
  + Recipient account must exist (basic simulation, can be extended in real-world applications).
* **Output**: Updated account balance and confirmation of the successful transfer.

**3.5 Display Account Balance**

* The system shall display the current balance to the user at the top of the dashboard.

**4. Business Rules**

1. **Account Creation**:
   * An account cannot be created without an initial deposit of at least $1000.
2. **Deposit Rules**:
   * Users can deposit any amount greater than zero.
3. **Withdrawal Rules**:
   * Users can only withdraw amounts less than or equal to their current balance.
   * Users cannot withdraw negative amounts.
4. **Transfer Rules**:
   * Users can only transfer amounts less than or equal to their current balance.
   * Transfer transactions should have a valid recipient name.

**5. Non-Functional Requirements**

**5.1 Performance Requirements**

* The system should handle user inputs and transactions within 2 seconds under normal operating conditions.

**5.2 Usability Requirements**

* The UI should be intuitive, allowing users to navigate through different functionalities easily.

**5.3 Security Requirements**

* The application should not expose sensitive information such as account balances to unauthorized users.

**5.4 Reliability Requirements**

* The system should maintain data integrity for all transactions performed by the user.

**6. User Interface Requirements**

**6.1 User Dashboard**

* **Components**:
  + Display of account balance at the top.
  + Sections for depositing funds, withdrawing funds, and transferring money.
  + Notifications for successful transactions.

**7. Assumptions and Dependencies**

* The application assumes that users have a basic understanding of web applications.
* The system relies on the web browser for user interface rendering and local storage.