**Requirements for the Bank System Application**

**Functional Requirements**

1. **Account Management**:
   * Users should be able to create new accounts.
   * Retrieve account details by account ID.
   * View a list of all accounts.
2. **Transaction Management**:
   * Deposit money into an account.
   * Withdraw money from an account (with sufficient balance).
   * Transfer money between two accounts.
   * View transaction history for any account.
3. **Validation**:
   * Ensure withdrawal and transfer operations are valid based on available balance.
   * Prevent negative or zero-value transactions.
4. **Authentication**:
   * Users should log in securely to access their account.
   * Ensure that only authorized users can perform account operations.
5. **Error Handling**:
   * Return meaningful error messages for invalid operations (e.g., insufficient balance, invalid account ID).

**Non-Functional Requirements**

1. **Performance**:
   * The system should handle up to 100 concurrent users with minimal latency.
   * Transactions should be processed within 1 second.
2. **Scalability**:
   * The application should be scalable to accommodate future growth in user base and transaction volume.
3. **Security**:
   * Encrypt sensitive data (e.g., passwords, account details).
   * Prevent unauthorized access using authentication mechanisms like JWT.
4. **Reliability**:
   * The application should ensure data consistency, even during system crashes.
   * Implement retries for failed transactions.
5. **Maintainability**:
   * Code should be modular and follow best practices to ensure easy updates and debugging.
   * Include proper documentation for APIs and core modules.
6. **Usability**:
   * Provide a user-friendly interface for accessing and managing accounts (optional if API-only).

**Technical Requirements**

1. **Programming Language**: Java 11 or higher.
2. **Framework**: Spring Boot.
3. **Database**: MySQL (or H2 for development and testing).
4. **Build Tool**: Maven.
5. **Testing**: JUnit and Postman.
6. **Containerization**: Docker (optional for deployment).
7. **Version Control**: Git for source code management.
8. **API Documentation**: Swagger (for easy exploration of endpoints).

**Prerequisites**

1. **Environment**:
   * Java 11 installed.
   * Maven installed.
   * MySQL database setup.
   * Docker (optional, if containerizing).
2. **Setup**:
   * Basic knowledge of RESTful APIs.
   * IDE (e.g., IntelliJ IDEA, Eclipse) for development.