Table of Contents

1. Problem Statement

The project aims to create a API for banking system where users can easily manage accounts, deposits, withdrawals, and balance inquiries.

2. Requirements/Features

- **Functional Requirements**
  - **Account Creation**
  - Users can create a new customer account by providing their name, email, and password.
  - **Deposit**
  - Users can deposit funds into their account by specifying the account number and the deposit amount.
  - **Withdrawal**
  - Users can withdraw funds from their account by specifying the account number and the withdrawal amount.
  - **Balance Inquiry**
  - Users can check the balance of their account by providing the account number.

- **Non-Functional Requirements**
  - **Performance:** The application should provide a fast and responsive user experience.
  - **Security:** User data should be encrypted and stored securely. Access to accounts should require authentication.
  - **Scalability:** The application should be designed to handle an increasing number of users and transactions.
  - **Usability:** The user interface should be intuitive and user-friendly.

3. Technologies

- Front-End: React.js
- Back-End: Django and Django REST framework (Python)
- Database: MySQL
- Authentication: JSON Web Tokens (JWT)

4. APIs to be Developed

- Create a new customer account
- Deposit funds into an account
- Withdraw funds from an account
- Check the account balance by providing the account number

5. Modules

Front-End: React components and views

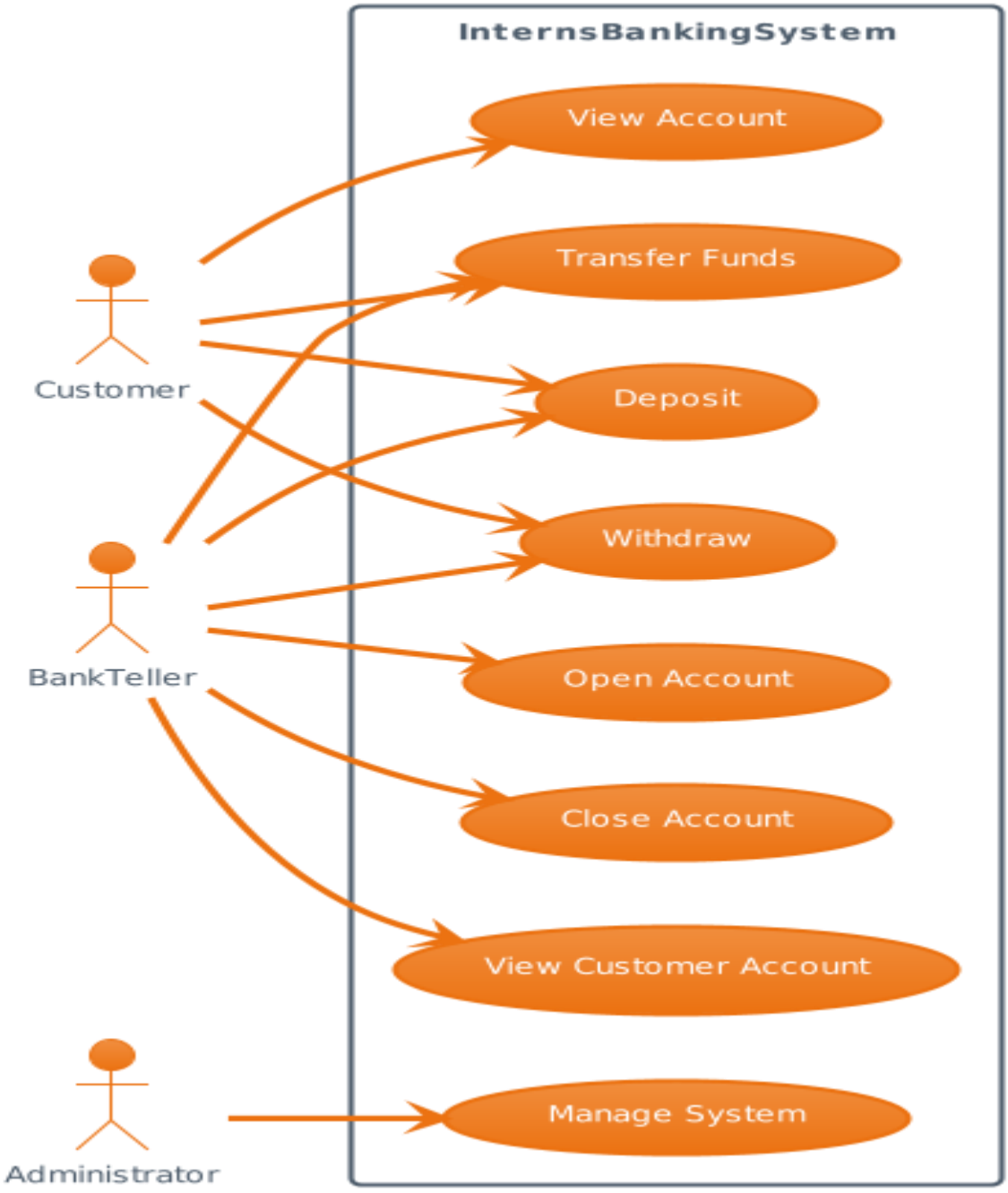
Back-End: Django REST framework for routes, controllers, and database interactions

Database: MySQL database for account data storage

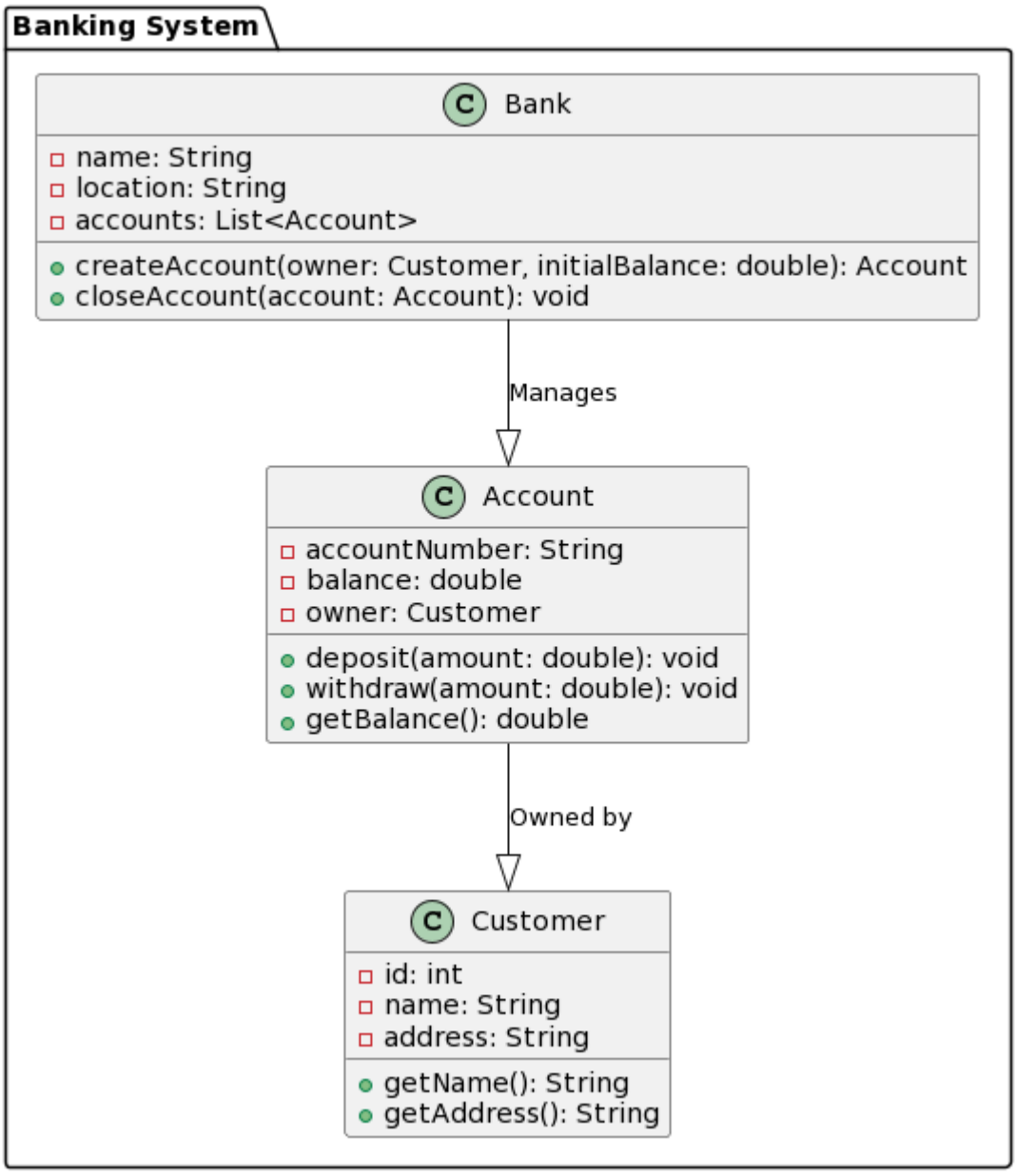
Authentication: JWT-based secure user authentication

Error Handling: Robust error handling and security measures

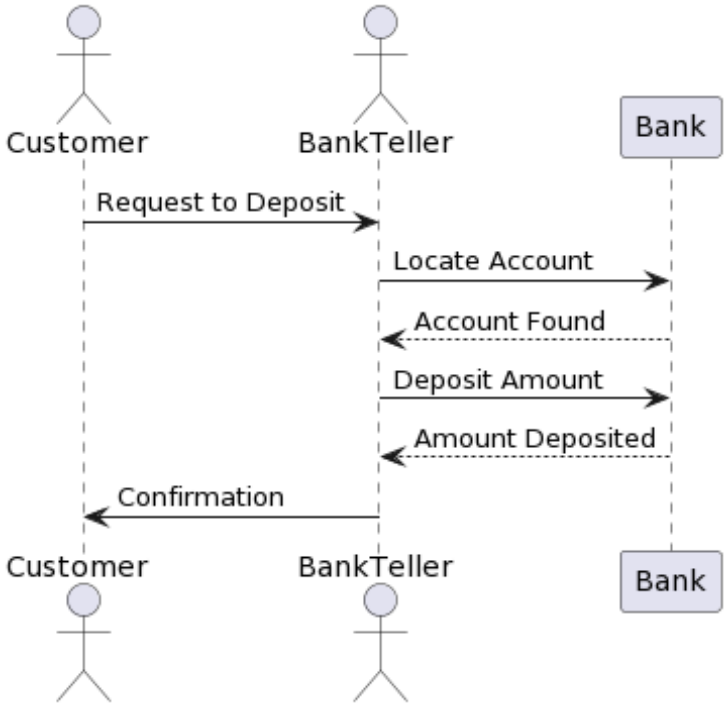
6. Use Case Diagram



7. Class Diagram



8. Sequence Diagram



9. Future Scope

- **Fund Transfer:** Implement the capability for users to transfer funds between accounts, enabling payments and transactions.
- **Multi-Currency Support**: Extend the system to support multiple currencies, enabling international transactions.
- **Enhanced Security:** Implement advanced security measures, such as two-factor authentication (2FA), to further protect user accounts.
- **Mobile Application:** Create a mobile application that provides access to the banking system, making it more convenient for users on the go.
- **Notifications:** Integrate a notification system to alert users about account activities, security updates, and promotions.
- **Credit and Loan Services:** Introduce credit and loan services, allowing users to apply for loans or credit cards.
- **Personal Finance Management:** Develop tools for personal finance management, such as budgeting, expense tracking, and financial planning.
- **Integration with Third-Party Services:** Enable integration with external financial services like payment gateways, investment platforms, or tax preparation software.