2024

Bank Accounting System

SW Masters

Software Engineering

5/20/2024

Bank Accounting System:

Embark on a Revolutionary Journey in Banking with Our Bank Accounting System

Born out of a commitment to enhancing financial transactions and streamlining banking operations, our Bank Accounting System project aims to create a cutting-edge platform that seamlessly connects clients, employees, and administrators. This project is designed to transform traditional banking processes, making them more efficient, secure, and user-friendly.

Project Description:

The Bank Accounting System aims to optimize and streamline the entire banking process, providing a comprehensive platform for clients, employees, and administrators to interact seamlessly. The system will focus on managing transactions such as withdrawals and deposits, user management, and administrative oversight. Additionally, it will incorporate robust security measures to ensure data privacy and integrity. The project seeks to create an innovative and user-friendly solution that fosters efficiency and reliability within the banking community.

Project Objectives:

1. **Efficient Transaction Management:**
    - Develop a robust system that empowers employees to manage client transactions effortlessly, from verifying requests to updating account details.
2. **Intuitive Client Interaction:**
    - Design a user-friendly interface allowing clients to request transactions, view their account status, and receive instant confirmations.
3. **Comprehensive Client Management:**
    - Provide employees with tools to add, remove, and search for client information, ensuring up-to-date and accurate client records.
4. **Administrative Oversight:**
    - Implement features that enable administrators to monitor all transactions, manage employees, and maintain overall system integrity.
5. **User-Friendly Interface:**

- Create an intuitive and visually appealing user interface for clients, employees, and administrators, ensuring a seamless experience in navigating the system.

6. **Security and Privacy Focus:**
    - Prioritize the security and privacy of user data by implementing advanced encryption and customizable privacy settings.

Key Components

1. **Client Module:**
    - User registration, login, and profile management functionalities.
    - Transaction request and confirmation features.
2. **Employee Module:**
    - Employee profile creation and management tools.
    - Client account management, including adding, removing, and searching client details.
3. **Transaction Module:**
    - Efficient handling of transaction requests, including withdrawals and deposits.
    - Real-time transaction processing and confirmation.
4. **Administration Module:**
    - Monitoring and reporting of all transactions.
    - Employee management capabilities, including hiring and firing.
5. **Security Module:**
    - Implementation of robust security measures to safeguard user data.
    - Customizable privacy settings for users.

Constraints

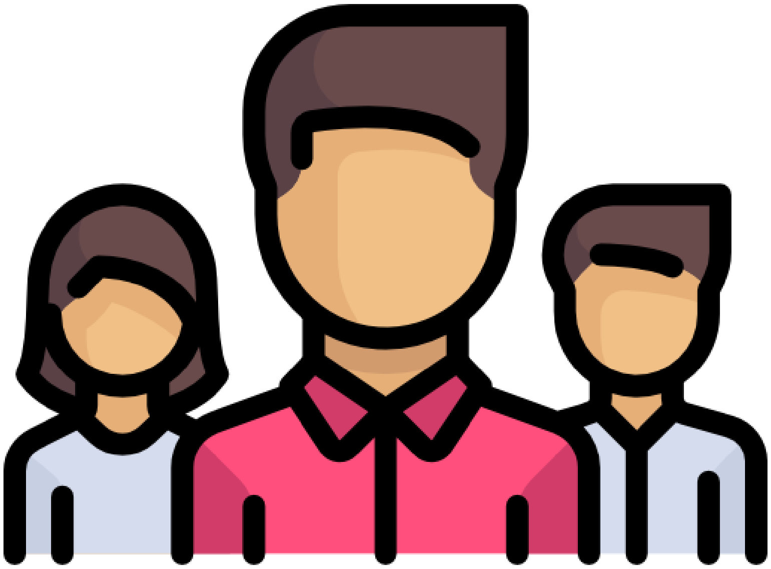
1. **Budget:**
    - The project will be executed within the allocated budget.
2. **Timeline:**
    - The development timeline is fixed, and the project must be delivered on schedule.

Expected Outcomes

1. A fully functional Bank Accounting System that simplifies the banking workflow for clients, employees, and administrators.
2. A secure and efficient platform for managing client transactions and internal operations.
3. Enhanced user experience through regular updates and feature enhancements.

By implementing this Bank Accounting System, we aim to revolutionize the banking experience, ensuring that financial transactions and administrative processes are handled with the utmost precision and security.

Team:



Arwa Hassan	225176
Mahmoud Maher	225200
Youmna Alaa	225023
Akram Mohamed	225186

Functional Requirements:

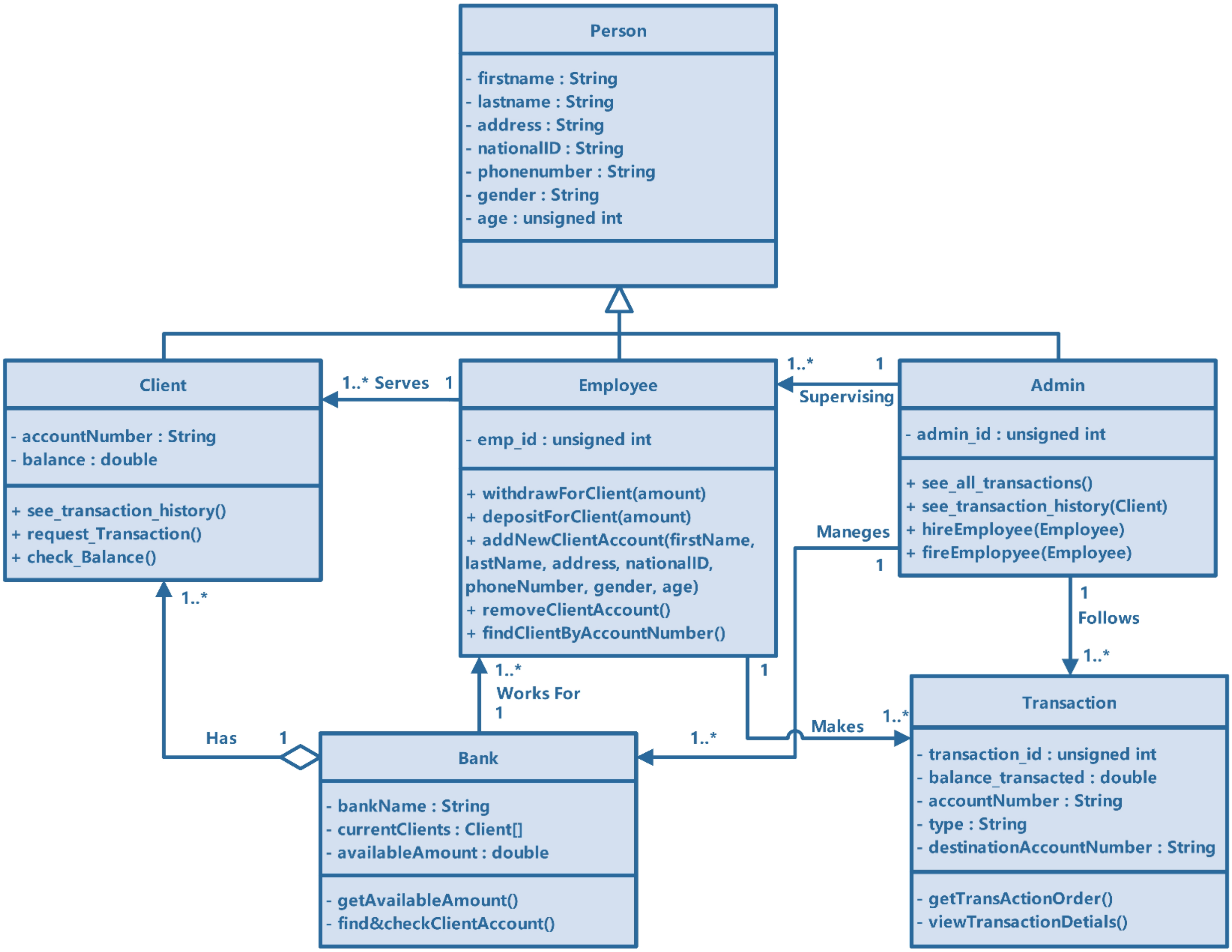
1. **User Registration and Authentication:**
   - **The system shall allow clients to register and create accounts.**
   - **The system shall authenticate users during login using a username and password.**
2. **Client Transaction Requests:**
   - **The system shall enable clients to request transactions, including withdrawals and deposits.**
   - **The system shall allow clients to view the status of their transaction requests.**
3. **Transaction Processing:**
   - **The system shall enable employees to process transaction requests from clients.**
   - **The system shall verify the feasibility of transactions before processing.**
   - **The system shall update client account balances after transaction processing.**
4. **Client Management:**
   - **The system shall allow employees to add new clients.**
   - **The system shall allow employees to remove clients.**
   - **The system shall enable employees to search for and view client information.**
5. **Administrative Functions:**
   - **The system shall provide administrators with an overview of all transactions.**
   - **The system shall allow administrators to hire and fire employees.**
   - **The system shall enable administrators to generate reports on transactions and employee performance.**
6. **Security and Privacy:**
   - **The system shall encrypt sensitive data such as passwords and transaction details.**
   - **The system shall provide customizable privacy settings for clients and employees.**

Non-Functional Requirements:

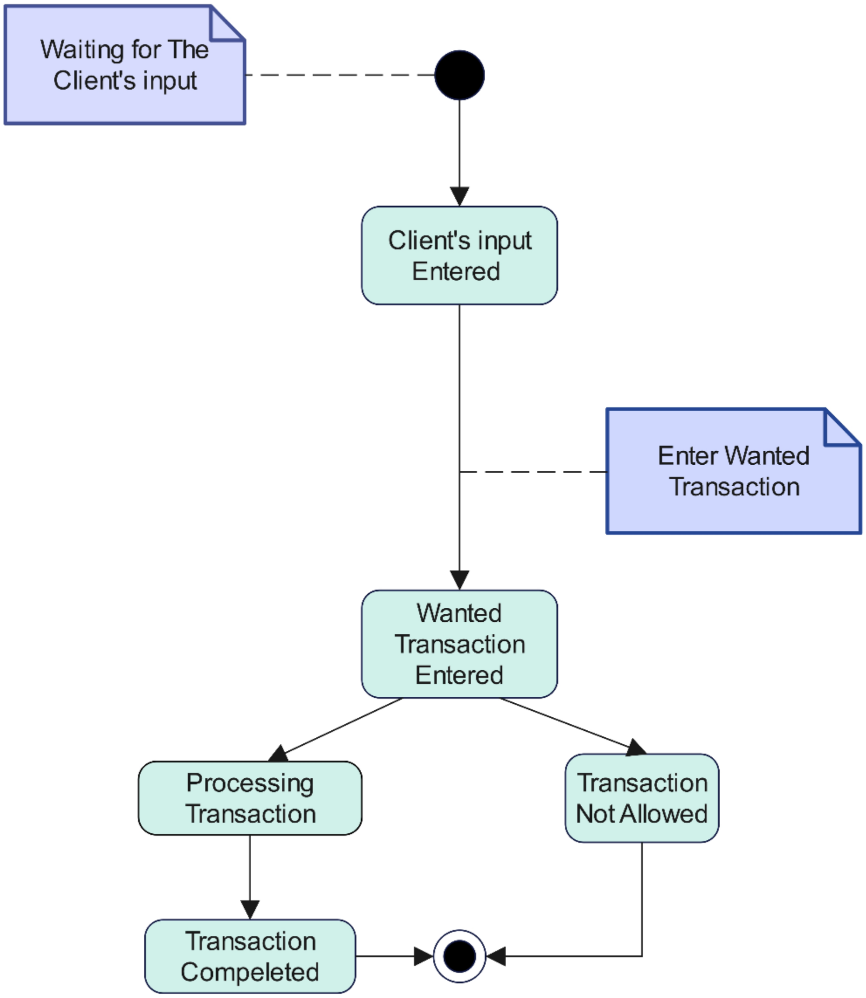
1. **Performance:**
   - **The system shall process transaction requests within 2 seconds.**
   - **The system shall support concurrent access by at least 100 users without performance degradation.**
2. **Usability:**
   - **The system shall have an intuitive and user-friendly interface.**
   - **The system shall provide clear and concise error messages and guidance for users.**
3. **Reliability:**
   - **The system shall have an uptime of 99.9% to ensure availability.**
   - **The system shall implement failover mechanisms to handle system failures.**
4. **Scalability:**
   - **The system shall be scalable to accommodate future growth in the number of users and transactions.**
5. **Security:**
   - **The system shall comply with industry standards for data security and privacy (e.g., PCI DSS, GDPR).**
   - **The system shall log all access and changes to data for auditing purposes.**
6. **Maintainability:**
   - **The system shall be designed with modular components to facilitate easy maintenance and updates.**
   - **The system shall include comprehensive documentation for developers and users.**
7. **Compatibility:**
   - **The system shall be compatible with major web browsers (e.g., Chrome, Firefox, Edge, Safari).**
   - **The system shall be accessible from various devices, including desktops, tablets, and smartphones.**

By adhering to these functional and non-functional requirements, the Bank Accounting System will provide a robust, secure, and user-friendly platform for managing banking operations efficiently.

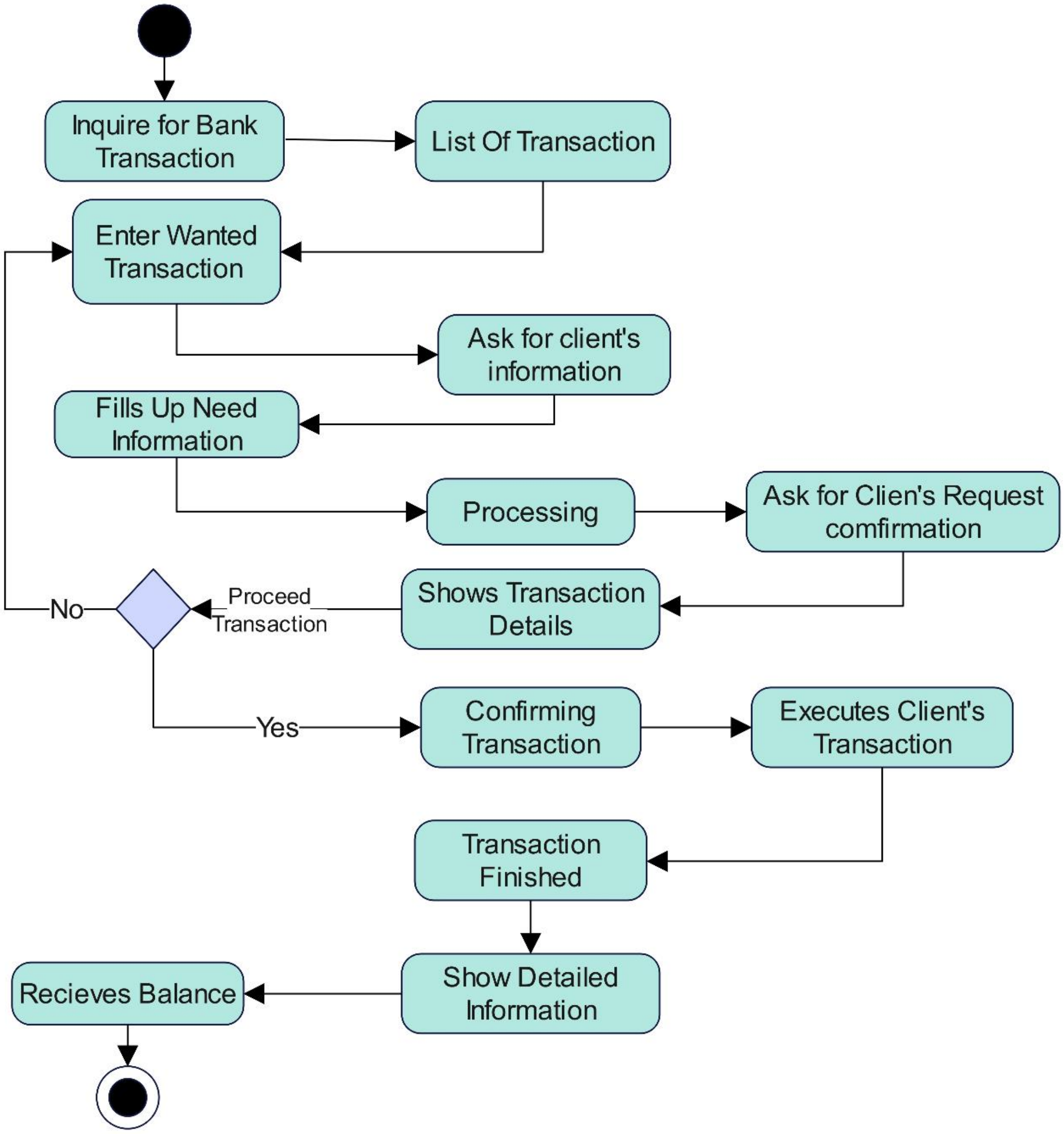
Class Diagram:



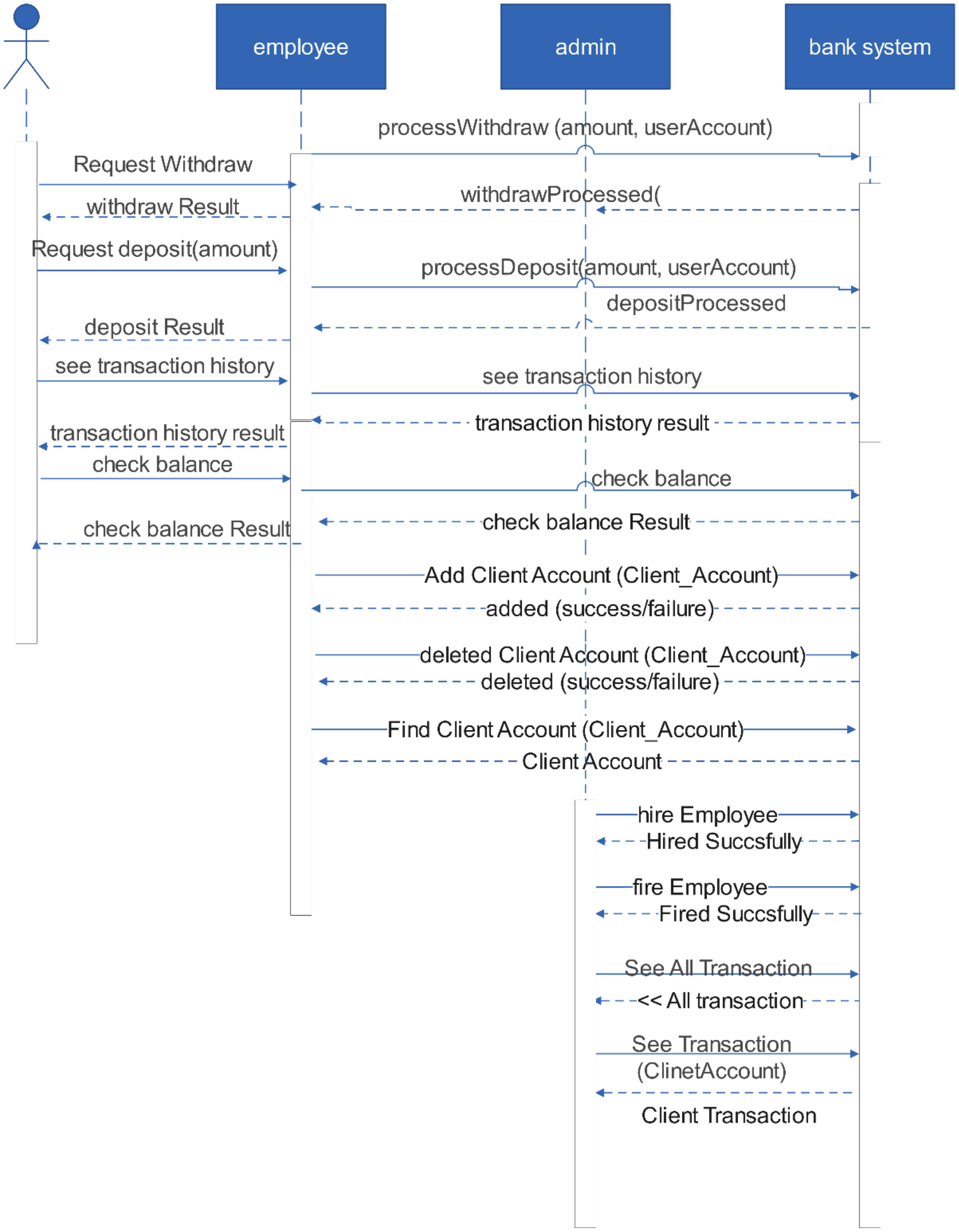
State Diagram:



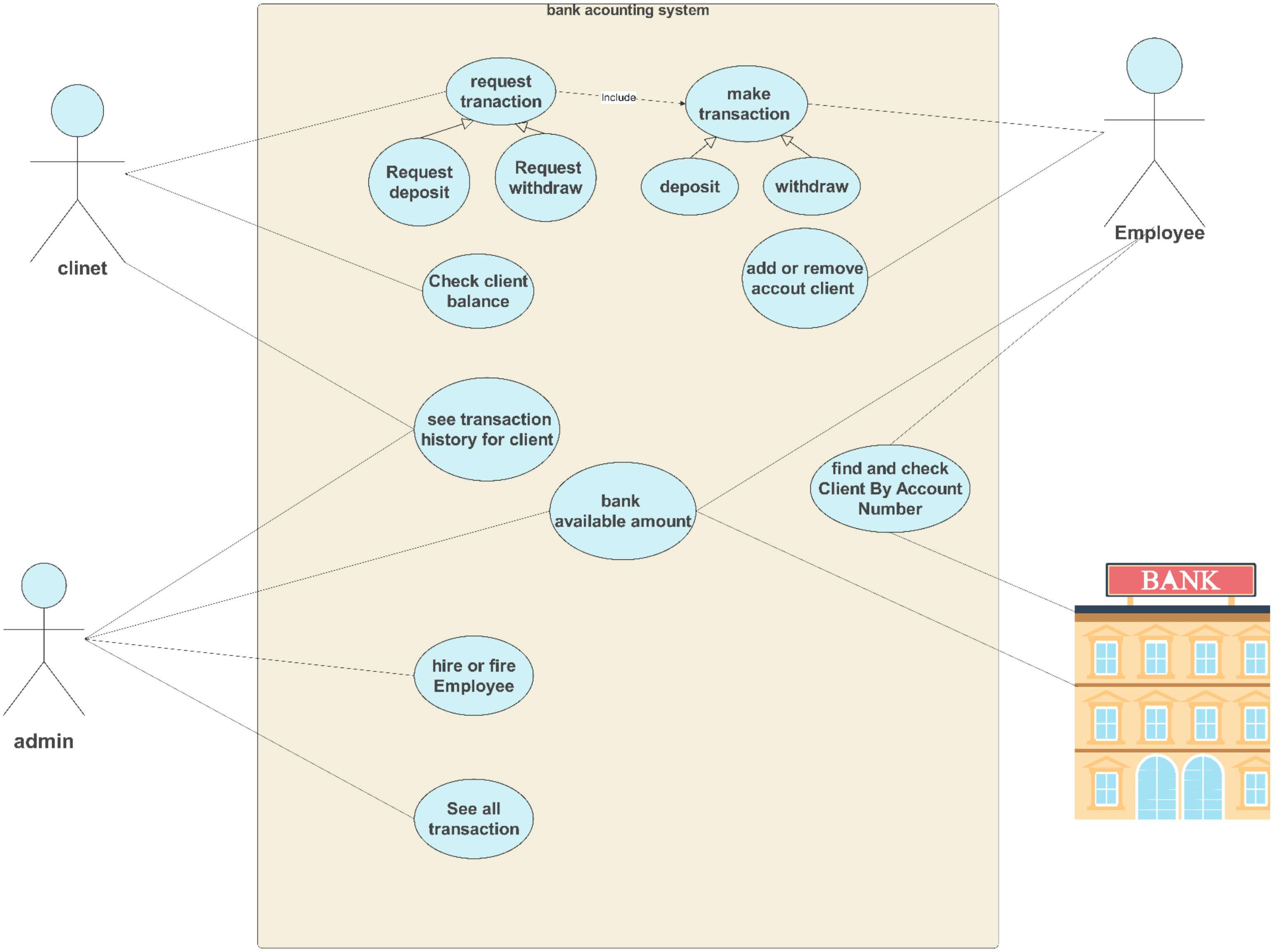
Activity Diagram:


Inquire for Bank Transaction
List Of Transaction
Enter Wanted Transaction
Ask for client's information
Fills Up Need Information
Processing
Ask for Clien's Request comfirmation
No
Proceed Transaction
Shows Transaction Details
Yes
Confirming Transaction
Executes Client's Transaction
Transaction Finished
Recieves Balance
Show Detailed Information

Sequence Diagram:



Use Case Diagram:

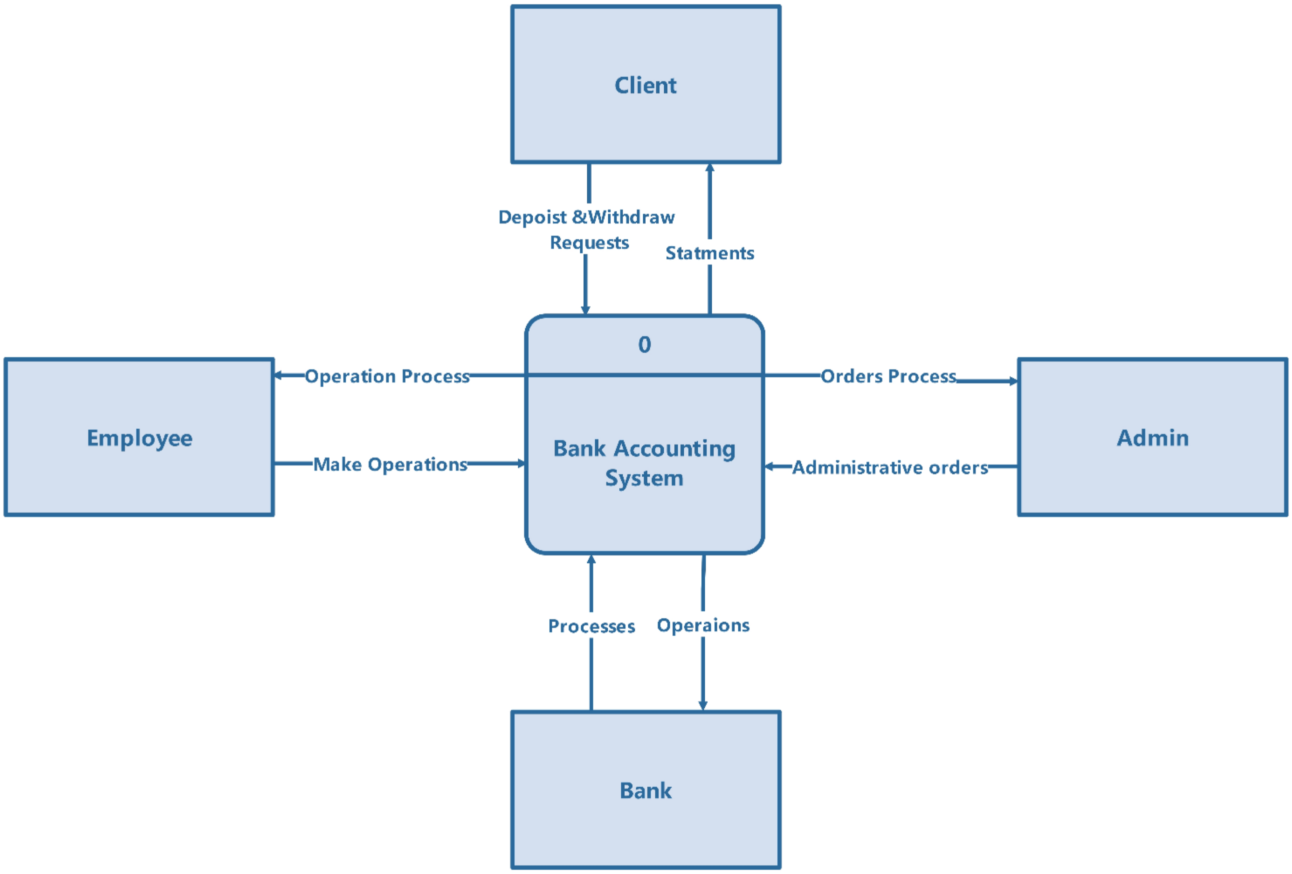


Use Case Scenario:

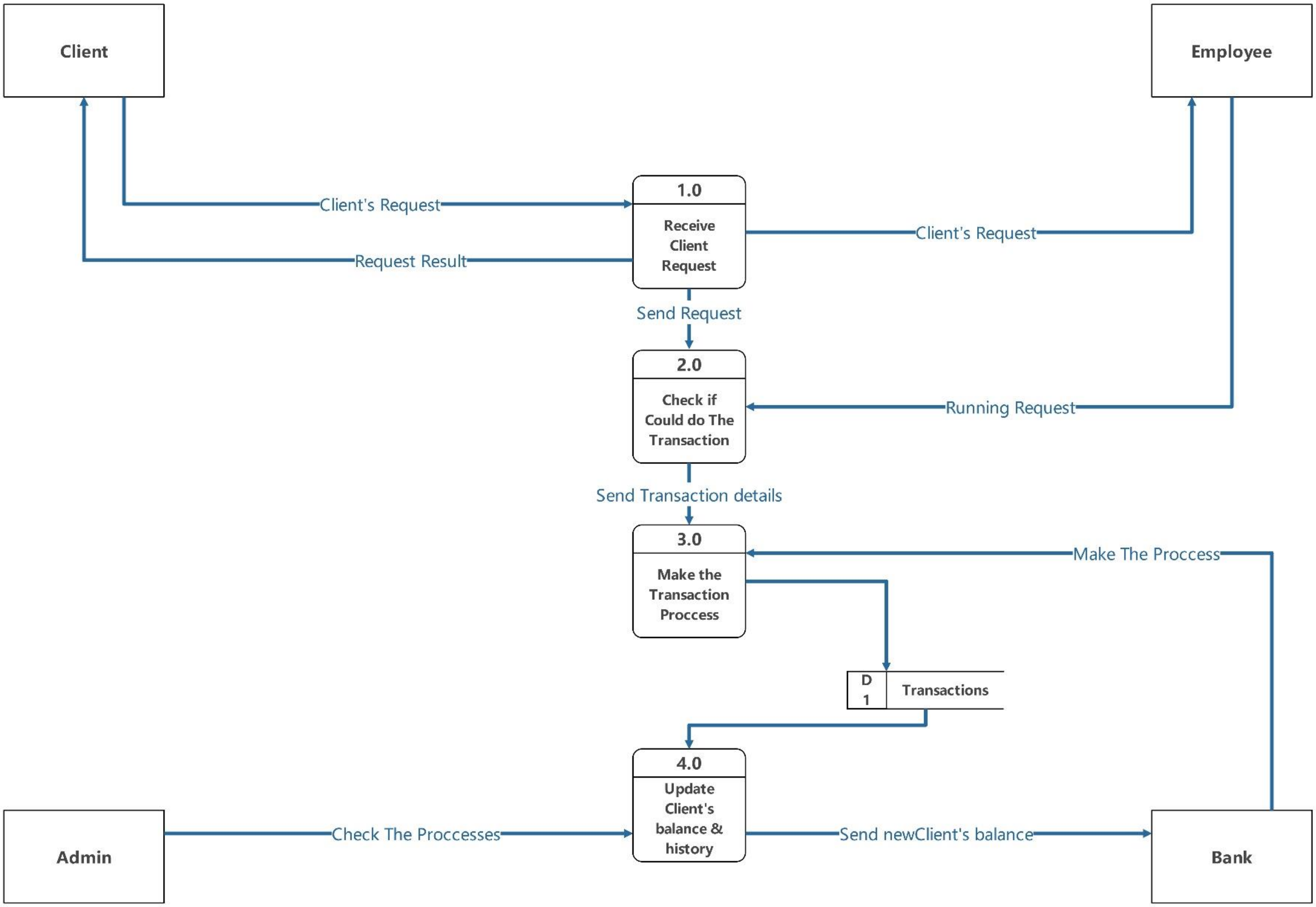
Use Case Name:	Bank Accounting System.	
Actor(s):	Client, Employee, Admin.	
Description:	This use case describes the process of client requesting a transaction process to do withdraw or deposit. then, The Employee take the request and check if he could make this transaction by return to bank and find if client there and check his balance and make this process and save this transaction. Admin could see all transaction history and hire or fire Employees in bank.	
Typical Course of Events:	Actor Action:	System Response:
Events	Step1: **Client sends Transaction Request.** Step2: **Employee will check and make the process.** Step5: **Admin can check all transactions and manage Employee and bank.** Step6: **Client could check his balance.** Step8: **Employee tells Client's balance.**	Step3: **Find and check Clients Account and make the Transaction.** Step4: **Save transaction and update clients history and balance. Request.** Step7: **Return his balance.**
Alternate Events:	Step1: **There's No Account for client in Bank.** Step2: **Client requests a failed Transaction.**	
Precondition:	1- **User Registration.** 2- **System Accessibility.**	
Post condition:	1- **Making client's Transactions.** 2- **Updated client's data.** 3- **Check client's balance.**	
Assumption:	1- **User Authenticity.** 2- **System Reliability.** 3- **Client have the balance to make processes.**	

Data Flow Diagram:

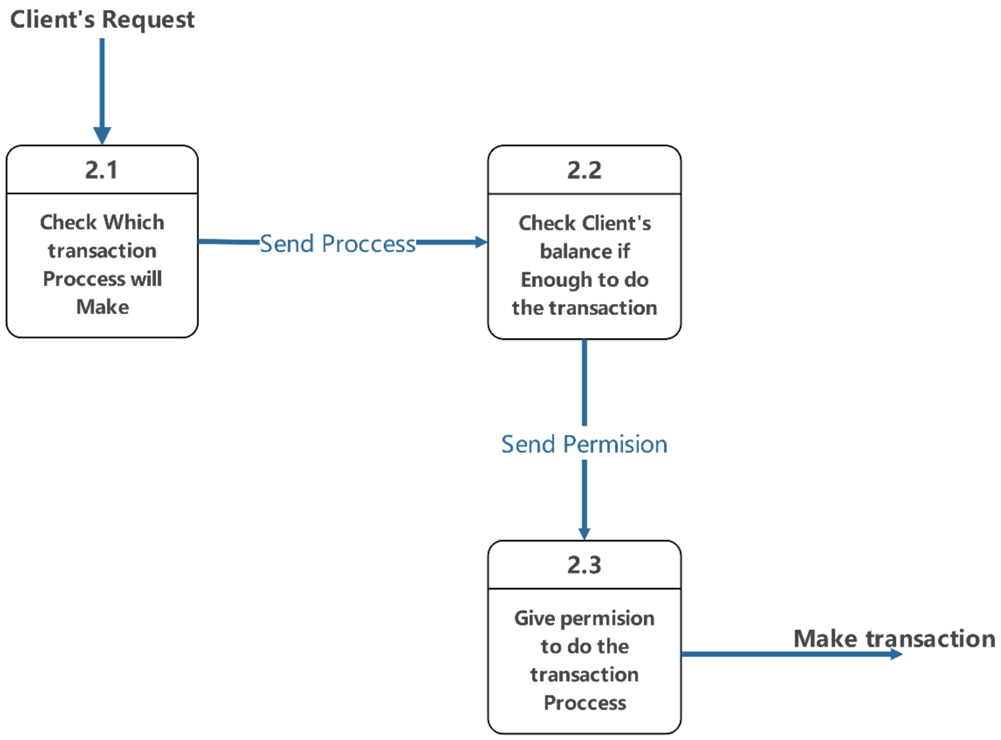
Context Level Diagram:



Level 0 Diagram:



Level 1 Diagram:


Client's Request
2.1
Check Which transaction Proccess will Make
Send Proccess
2.2
Check Client's balance if Enough to do the transaction
Send Permision
2.3
Give permision to do the transaction Proccess
Make transaction