



**Stu. Name**

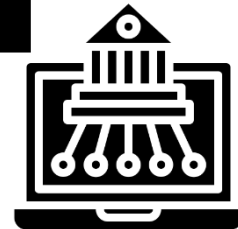
**Ahmed Abd El-Salam**

**Stu. ID: 202302445**

**Course Title: Software Engineering**

**system Of**

**BANK**



## 1. Introduction

### Need for the System

The banking system is needed to simplify and automate banking tasks like account management, transactions, and reporting. It connects customers and administrators to perform banking functions efficiently and securely. The system will work with existing platforms like ATM networks and online payment gateways. It supports the organization's goals by improving customer service and reducing manual work.

---

## 2. User Requirements

1. **Customers** should log in securely and view their account details.
  2. **Customers** need to send and receive money easily.
  3. **Customers** want to apply for loans online.
  4. **Customers** should get SMS/email alerts for transactions.
  5. **Administrators** need to manage user accounts and monitor activities.
  6. **Administrators** should generate and view system reports.
  7. **Customers and administrators** should access transaction histories.
- 

## 3. Functional Requirements

### 1. Secure Login

- **Description/Action:** Allow users to log in with a username and password securely.
- **Inputs:** Username, password.
- **Source:** User input.
- **Pre-condition:** User must have an account.
- **Post-condition:** User gains access to the system.
- **Output:** Login success or failure message.



## 2. Account Overview

- **Description/Action:** Show customers their account balance and details.
- **Inputs:** Customer ID.
- **Source:** System database.
- **Pre-condition:** Logged-in user.
- **Post-condition:** Display account information.
- **Output:** Account balance and details.

## 3. Transaction Processing

- **Description/Action:** Process money transfers between accounts.
- **Inputs:** Sender and receiver account IDs, transfer amount.
- **Source:** User input.
- **Pre-condition:** Sufficient balance in sender's account.
- **Post-condition:** Transfer is completed, balances updated.
- **Output:** Confirmation of transfer.

## 4. Loan Applications

- **Description/Action:** Enable customers to apply for loans.
- **Inputs:** Loan amount, repayment period.
- **Source:** User input.
- **Pre-condition:** User meets loan eligibility criteria.
- **Post-condition:** Loan request is submitted.
- **Output:** Application status.



## 5. Notifications

- **Description/Action:** Notify users of account activity via SMS or email.
- **Inputs:** Transaction details.
- **Source:** System-generated logs.
- **Pre-condition:** Valid contact details.
- **Post-condition:** Notification is sent.
- **Output:** SMS or email confirmation.

## 6. User Management

- **Description/Action:** Allow administrators to manage user accounts.
- **Inputs:** User details.
- **Source:** Administrator input.
- **Pre-condition:** Admin access rights.
- **Post-condition:** User account updated.
- **Output:** Confirmation of changes.

## 7. Reporting

- **Description/Action:** Generate financial or activity reports for administrators.
- **Inputs:** Date range or filters.
- **Source:** System database.
- **Pre-condition:** Administrator access.
- **Post-condition:** Report generated.
- **Output:** Report in PDF or CSV format.



## 8. Database Management

- **Description/Action:** Maintain and update the system database to ensure data integrity and security.
- **Inputs:** Data updates, backups, and recovery plans.
- **Source:** IT staff.
- **Pre-condition:** Database administrator access.
- **Post-condition:** Database remains operational and secure.
- **Output:** Confirmation of updates or recovery completion.

## 9. IT Staff Responsibilities

- **Description/Action:** Oversee system maintenance, troubleshoot issues, and ensure software and hardware stability.
  - **Inputs:** Error logs, system performance metrics.
  - **Source:** System monitoring tools and reports.
  - **Pre-condition:** Access to IT infrastructure.
  - **Post-condition:** System remains functional and efficient.
  - **Output:** Resolved issues and system performance reports.
- 



## 4. Non-Functional Requirements

---

### 1. Security

The system must use encryption (e.g., AES) for all sensitive data.

- **Explanation:** The system must protect sensitive user data, such as passwords and transaction details, from unauthorized access. Encryption techniques like **AES (Advanced Encryption Standard)** ensure that data remains secure during storage and transmission. Additionally, the system should implement secure login mechanisms, such as multi-factor authentication (MFA), to prevent breaches.
- 

### 2. Performance

The system should process transactions in under 2 seconds.

- **Explanation:** This requirement ensures that the system operates efficiently. For instance:
    - ✓ **Transaction Speed:** Users should experience minimal delay, with transactions being processed within **2 seconds** to maintain user satisfaction.
    - ✓ **Load Times:** The user interface (UI) should load quickly even under heavy usage.
- 

### 3. Reliability

The system must have 99.9% uptime and backup mechanisms.

- **Explanation:** The system should be highly available, with a **99.9% uptime** (around 8.76 hours of downtime per year). Reliability ensures:
    - ✓ Users can access the system whenever needed.
    - ✓ Backup and disaster recovery mechanisms protect data from unexpected failures
- 



#### 4. Usability

The interface should be simple and support both desktop and mobile access.

- **Explanation:** This ensures the system is intuitive and easy to use for both customers and administrators. Examples include:
    - ✓ **Mobile-Friendly Design:** A responsive interface for smartphones and tablets.
    - ✓ **Accessibility:** Compliance with standards like WCAG (Web Content Accessibility Guidelines) for users with disabilities
- 

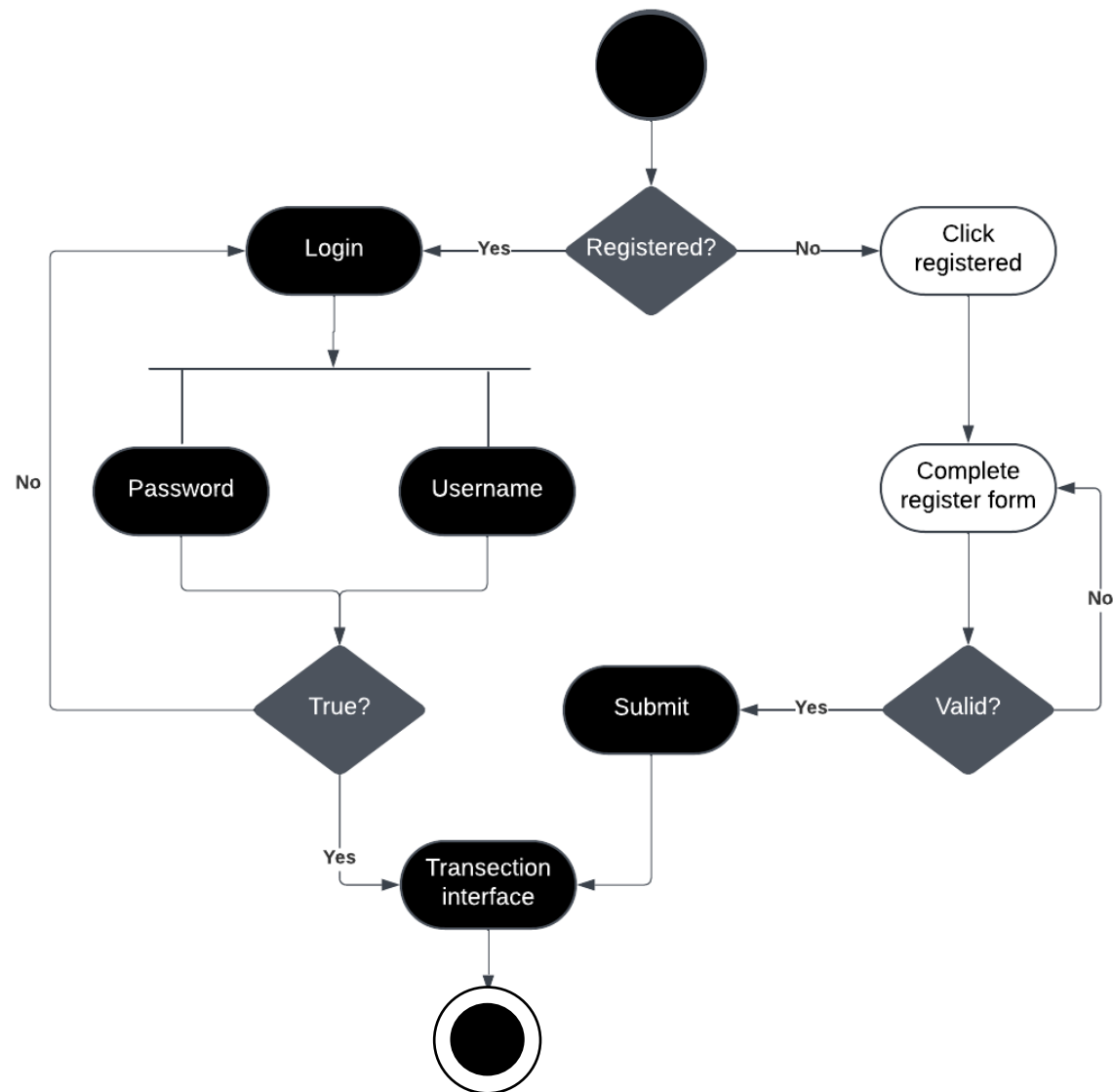
#### 5. Scalability

The system should handle a growing number of users and transactions efficiently.

- **Explanation:** Scalability ensures the system can handle growth in:
    - ✓ **Users:** If the bank acquires more customers, the system should still operate smoothly.
    - ✓ **Transactions:** An increase in daily transactions should not affect performance. This might involve cloud-based infrastructure or load-balancing techniques
- 

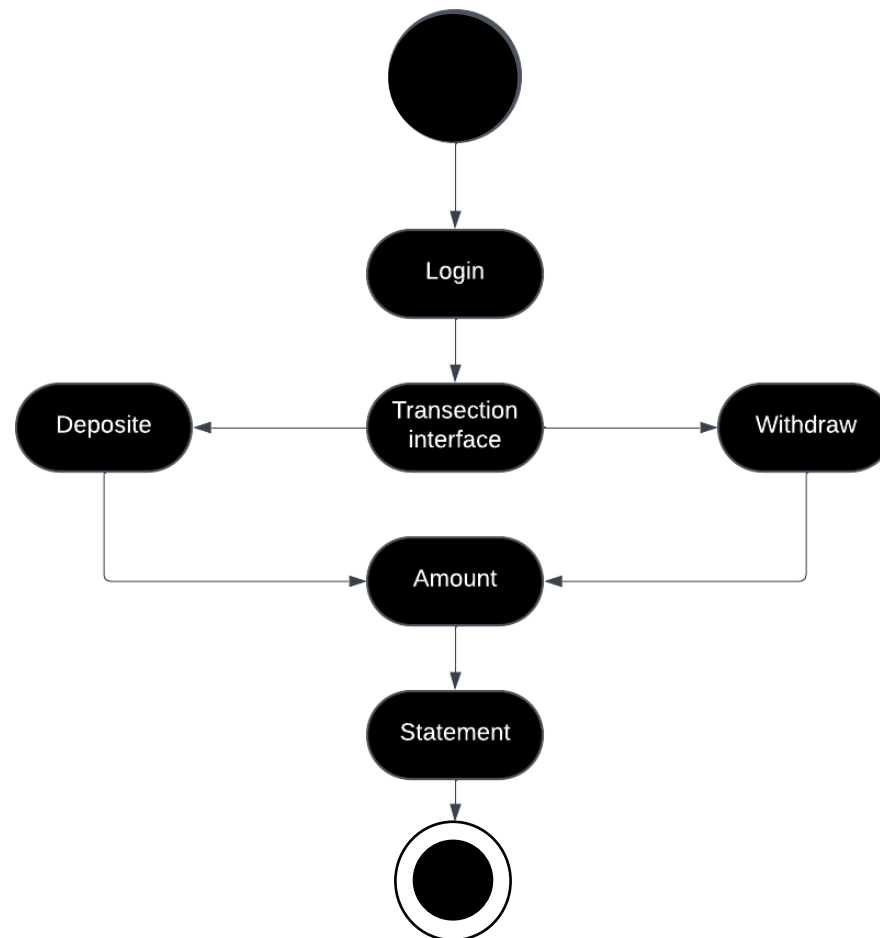


# LOGIN / REGISTER DIAGRAM

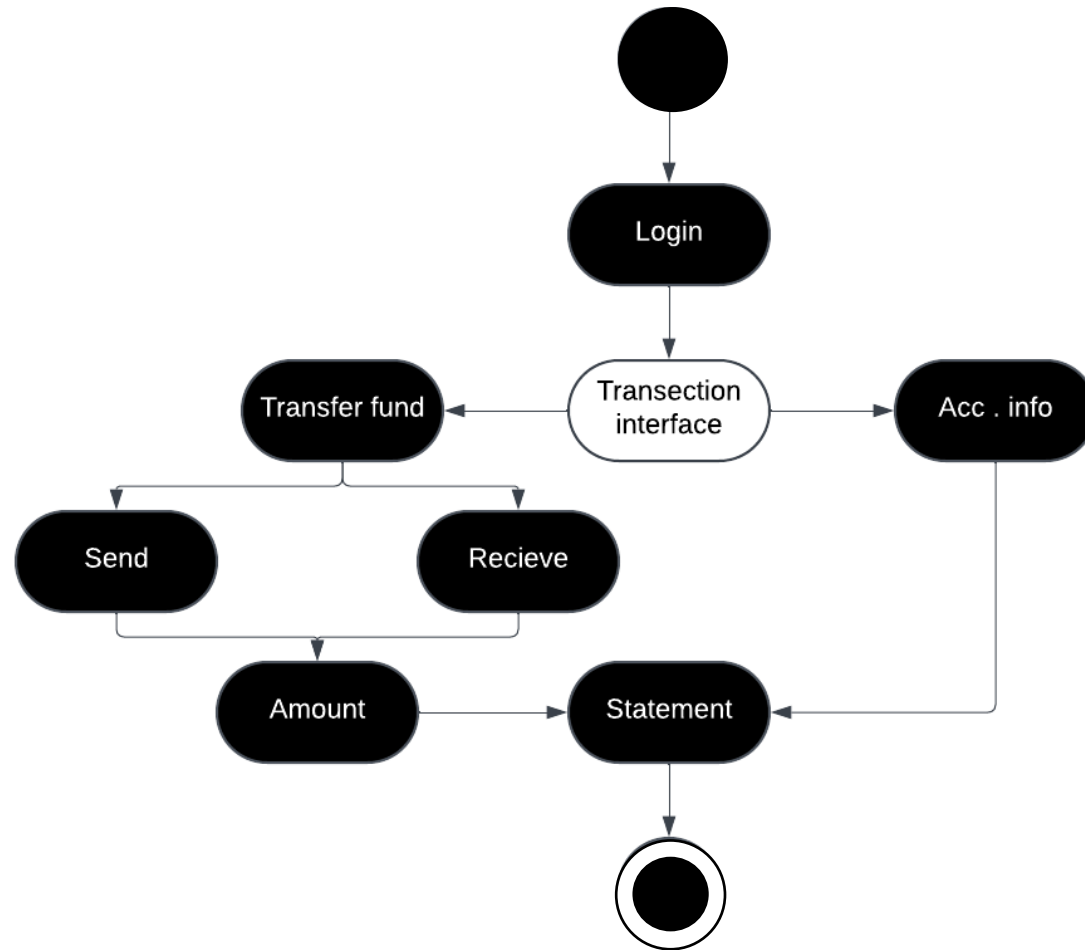




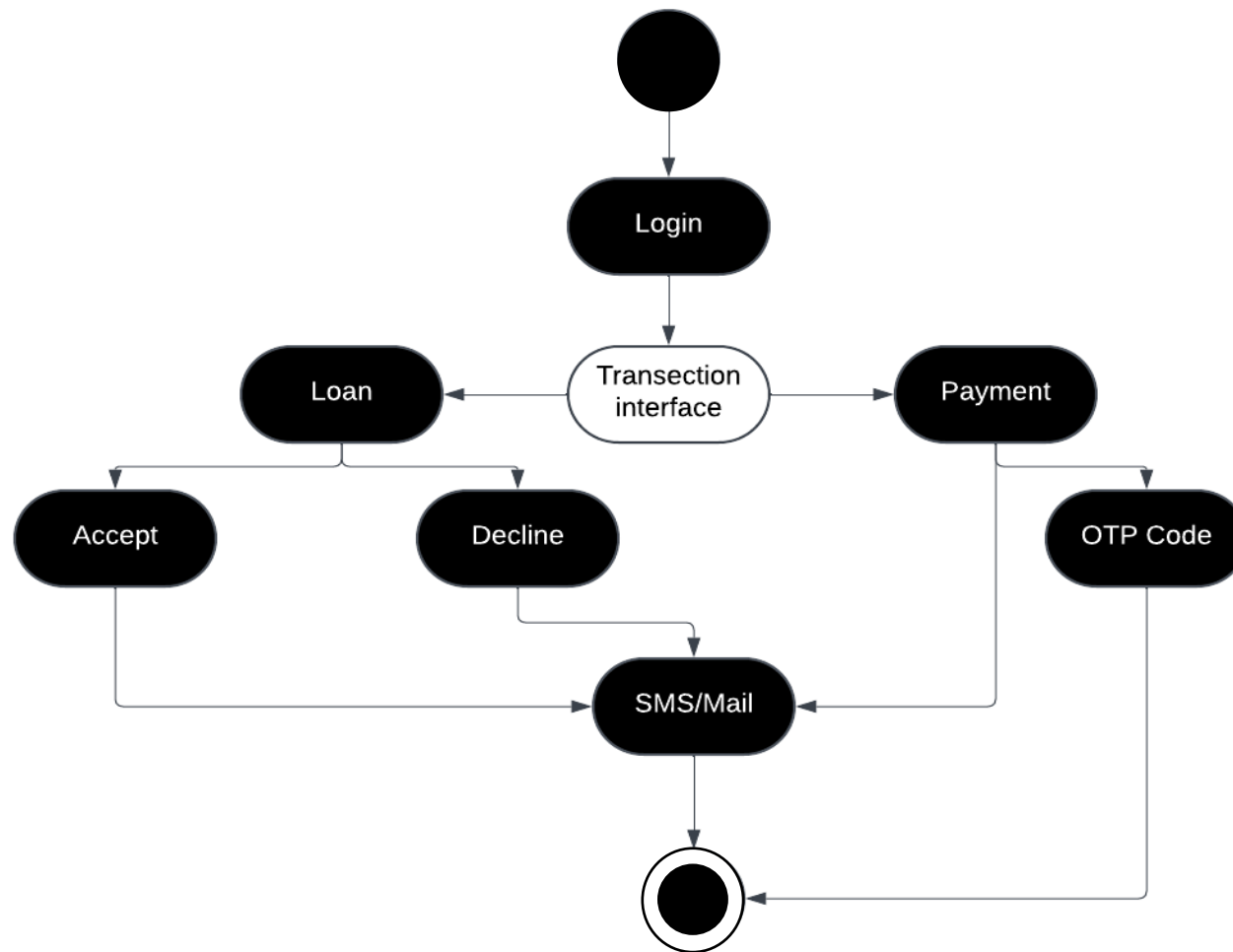
# DEPOSITE / WITHDRAW DIAGRAM



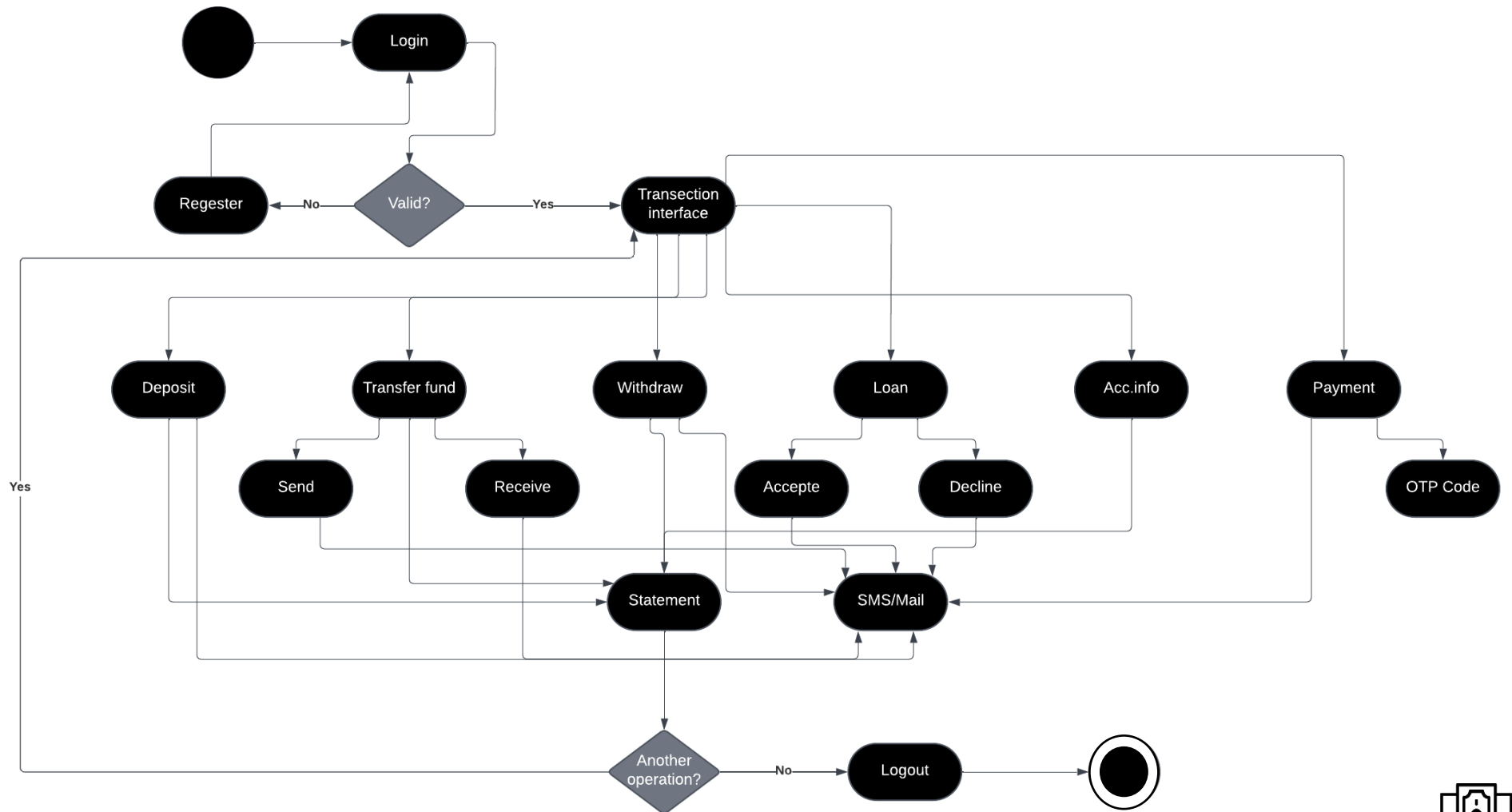
# TRANSFER FUND / ACC.INFO DIAGRAM



# LOAN / PAYMENT DIAGRAM



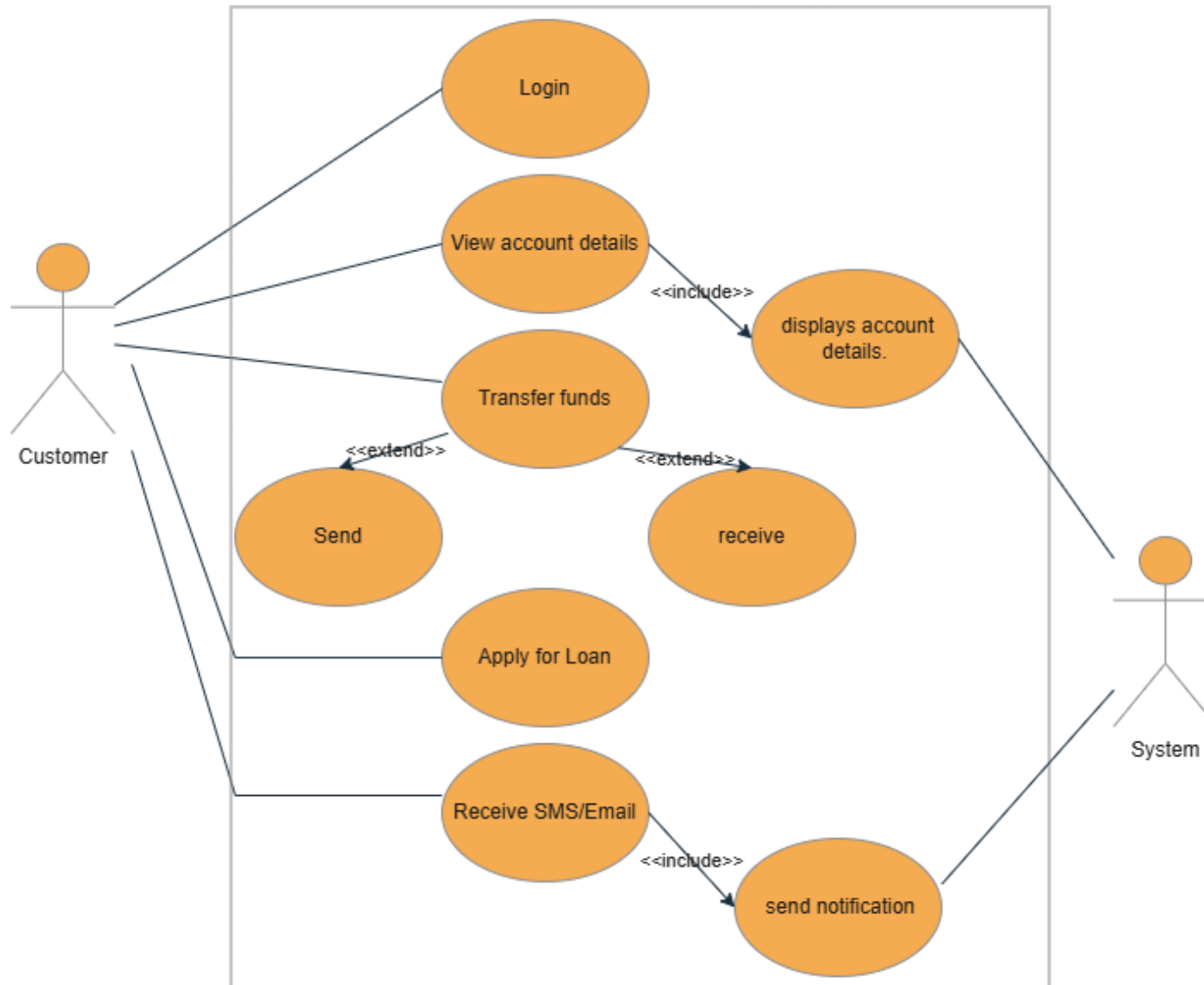
# GENERAL ACTIVITY DIAGRAM



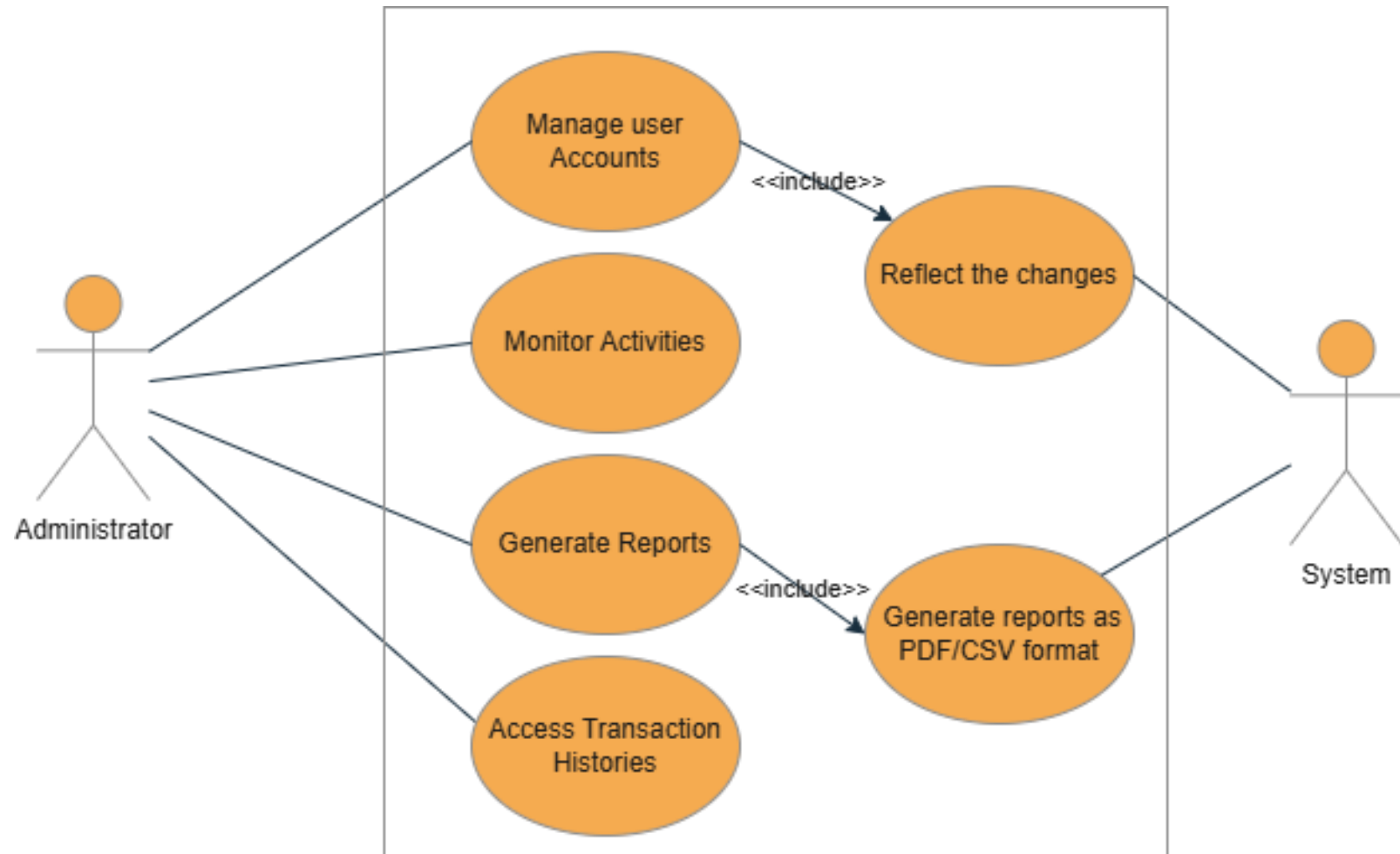
# GENERAL USE CASE DIAGRAM



# CUSTOMER USE CASE DIAGRAM



# ADMINISTRATOR USE CASE DIAGRAM



# GENERAL CLASS DIAGRAM

