

# **Document**

## **ATM Management System**



### **Submitted To**

Sir Zeeshan

### **Submitted By**

Sadia Sher (2022-SE-07)

Rimsha Aamir(2022-SE-13)

**Department of Computer Science**

University of Engineering and Technology, Lahore, New-Campus

## Contents

|                                  |    |
|----------------------------------|----|
| <b>Introduction:</b> .....       | 5  |
| <b>Background:</b> .....         | 5  |
| <b>Existing Systems:</b> .....   | 5  |
| <b>Our System:</b> .....         | 5  |
| <b>Advantages:</b> .....         | 5  |
| <b>Functions:</b> .....          | 5  |
| <b>Admin:</b> .....              | 5  |
| <b>User:</b> .....               | 5  |
| <b>UML Diagrams:</b> .....       | 6  |
| <b>Use Case Diagram:</b> .....   | 6  |
| <b>Use Cases:</b> .....          | 6  |
| <b>Admin:</b> .....              | 7  |
| <b>Use case 1:</b> .....         | 7  |
| <b>Use case 2:</b> .....         | 7  |
| <b>Use case 3:</b> .....         | 8  |
| <b>Use case 4:</b> .....         | 9  |
| <b>Use case 5:</b> .....         | 10 |
| <b>User:</b> .....               | 11 |
| <b>Use case 1:</b> .....         | 11 |
| <b>Use case 2:</b> .....         | 11 |
| <b>Use case 3:</b> .....         | 12 |
| <b>Use case 4:</b> .....         | 13 |
| <b>Use case 5:</b> .....         | 14 |
| <b>Class Diagram:</b> .....      | 15 |
| <b>Object Diagram:</b> .....     | 15 |
| <b>Sequence Diagram:</b> .....   | 16 |
| <b>Admin Functions:</b> .....    | 16 |
| <b>Log-in:</b> .....             | 16 |
| <b>Add users:</b> .....          | 16 |
| <b>View User Detail:</b> .....   | 17 |
| <b>Update User Detail:</b> ..... | 17 |
| <b>Delete User Detail:</b> ..... | 18 |
| <b>User Functions:</b> .....     | 18 |
| <b>Log-in:</b> .....             | 18 |
| <b>User deposit money:</b> ..... | 19 |

|                              |    |
|------------------------------|----|
| User withdraw money:.....    | 19 |
| Check balance: .....         | 20 |
| View statement: .....        | 20 |
| Collaboration Diagram: ..... | 21 |
| Admin Functions:.....        | 21 |
| Log-in: .....                | 21 |
| Add users: .....             | 21 |
| View User Detail: .....      | 21 |
| Update User Detail:.....     | 22 |
| Delete User Detail: .....    | 22 |
| User Functions: .....        | 22 |
| Log-in: .....                | 22 |
| Check balance: .....         | 23 |
| User deposit money:.....     | 23 |
| View statement: .....        | 23 |
| User withdraw money:.....    | 24 |
| Activity Diagram:.....       | 24 |
| Admin Functions:.....        | 24 |
| User Functions: .....        | 25 |
| State Machine Diagram:.....  | 26 |
| Admin Functions:.....        | 26 |
| Log-in: .....                | 26 |
| Update User Detail:.....     | 27 |
| Add users: .....             | 28 |
| Delete User Detail: .....    | 28 |
| View User Detail: .....      | 29 |
| User Functions: .....        | 29 |
| Log-in: .....                | 29 |
| User withdraw money:.....    | 30 |
| User deposit money:.....     | 30 |
| Check balance: .....         | 30 |
| View statement: .....        | 31 |
| Packages Diagram: .....      | 31 |
| Component Diagram: .....     | 31 |
| Deployment Diagram: .....    | 32 |
| SYSTEM ARCHITECTURE: .....   | 32 |

|                          |    |
|--------------------------|----|
| <b>Diagram:</b> .....    | 32 |
| <b>Advantages:</b> ..... | 33 |
| <b>Drawbacks:</b> .....  | 33 |

## Introduction:

## Background:

An ATM (Automated Teller Machine) Management System is a comprehensive software solution designed to efficiently manage and control the operations of Automated Teller Machines. ATMs are widely used for providing 24/7 banking services to customers, allowing them to perform various financial transactions without visiting a physical bank branch.

## Existing Systems:

Prominent players in the field of ATM Management Systems include established entities such as NCR Corporation and Diebold Nixdorf.

## Our System:

Our system ATM Management System a user-friendly interface, ensuring a seamless experience for users as they effortlessly navigate and execute various functions based on their specific needs. In contrast to other complex systems, our system prioritizes simplicity while remaining highly functional.

## Advantages:

Our system offers several advantages, including:

- User Friendly Interface
- Security
- Reliability

## Functions:

Our ATM Management System encompasses key functions to meet user requirements:

### Admin:

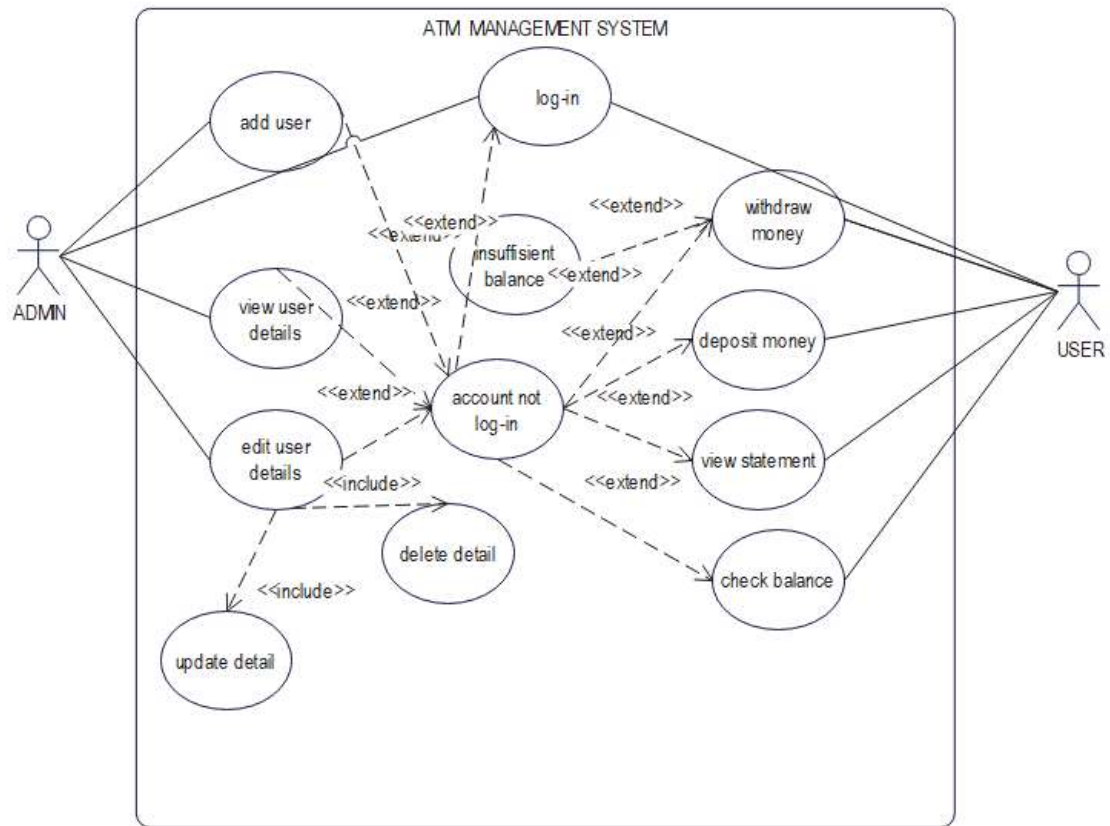
- **Log-in:** Providing secure access for authorized admin.
- **Add User:** Effortlessly add users to system.
- **View Users Detail:** Access a record of users.
- **Delete User Detail:** Effortlessly delete users detail to system
- **Update User Detail:** Effortlessly update users' details to system

### User:

- **Log-in:** Providing secure access for authorized users.
- **Deposit Cash:** Effortlessly add funds to your account.
- **Cash Withdrawal:** Secure and convenient withdrawal of funds.
- **View Statement:** Access a record of transactions.
- **Check Balance:** Instantly view the current account balance.

## UML Diagrams:

### Use Case Diagram:



### Use Cases:

| Actor | Function   |
|-------|--|
| Admin | <ol style="list-style-type: none"><li>1. Log-In</li><li>2. Add User Details</li><li>3. View User Details</li><li>4. Update User Details</li><li>5. Delete User Details</li></ol> |
| User  | <ol style="list-style-type: none"><li>1. Sign-in</li><li>2. Withdraw Money</li><li>3. Deposit Money</li><li>4. Check Balance</li><li>5. View Mini Statement</li></ol>            |

### Admin:

#### Use case 1:

|                |  |
|----------------|--|
| Name           | Log-in to account  |
| Actor          | Admin  |
| Goal           | As a admin, I want to log-in to ATM system.  |
| Trigger        | When admin clicks on log-in button.  |
| Pre-condition  | 1. Admin Must have an account  |
| Post-condition | Admin has logged-in  |
| Basic flow     | 1. Open ATM system.<br>2. Enter name<br>3. Enter password<br>4. Click on log-in button   |
| Alternate path | Not available  |
| Exception      | 1. Don't have account<br><b>Solution:</b><br>a. Admin should make an account.  |
| Qualities      | <b>Performance:</b> Admin must open ATM system in 2 sec.<br><b>Correctness:</b> ATM system must be open after logged-in<br><b>Security:</b> Account must be secure |

#### Use case 2:

|                |   |
|----------------|---|
| Name           | Add User Details  |
| Actor          | Admin   |
| Goal           | As a admin ,I want to add user details                                    |
| Trigger        | When Admin click Add user button  |
| Pre-condition  | 1. Must have account on ATM system<br>2. Must be logged-in to ATM system. |
| Post-condition | Admin add user successfully   |

|                |  |
|----------------|--|
| Basic flow     | <ol style="list-style-type: none"> <li>1. Click Add user detail button.</li> <li>2. Add user name, Account number and password of user.</li> <li>3. Also add gender, age and upload picture of user.</li> <li>4. Click add user button.</li> </ol>   |
| Alternate path | Not available  |
| Exception      | <ol style="list-style-type: none"> <li>1. Field is not filled.<br/><b>Solution:</b> <ol style="list-style-type: none"> <li>a. Fill all the fields.</li> </ol> </li> <li>2. Don't have account<br/><b>Solution:</b> <ol style="list-style-type: none"> <li>a. Admin should make an account</li> </ol> </li> <li>3. Don't signed-in to account<br/><b>Solution:</b> <ol style="list-style-type: none"> <li>a. Admin should log-in</li> </ol> </li> </ol> |
| Qualities      | <p><b>Performance:</b> Admin must open ATMS in 2 sec.</p> <p><b>Correctness:</b> User must be added after clicking on Add user button</p> <p><b>Security:</b> account must be secure</p>   |

### Use case 3:

|                |   |
|----------------|---|
| Name           | View User Details   |
| Actor          | Admin   |
| Goal           | As admin, I want to view User Details   |
| Trigger        | When user click on view User Details  |
| Pre-condition  | <ol style="list-style-type: none"> <li>1. Must be sign-in to account</li> </ol>   |
| Post-condition | Admin can view User Details   |
| Basic flow     | <ol style="list-style-type: none"> <li>1. Sign-in to account</li> <li>2. Click on View User Details</li> </ol>  |
| Alternate path | Not available   |
| Exception      | <ol style="list-style-type: none"> <li>1. Don't have account<br/><b>Solution:</b> <ol style="list-style-type: none"> <li>a. Admin should make an account.</li> </ol> </li> <li>2. Don't signed-in to account<br/><b>Solution:</b> <ol style="list-style-type: none"> <li>a. Admin should sign-in</li> </ol> </li> </ol> |



|           |  |
|-----------|--|
| Qualities | <b>Performance:</b> Admin must open ATMS in 2 sec.<br><b>Correctness:</b> User details must be view after clicking on view User detail button<br><b>Security:</b> Account must be secure |
|-----------|--|

**Use case 4:**

|                |   |
|----------------|---|
| Name           | Update User Details   |
| Actor          | Admin   |
| Goal           | As a admin, I want to update user details in their account.   |
| Trigger        | When admin clicks on Update button.   |
| Pre-condition  | 1. Admin is logged in to the system.<br>2. Access rights to manage users accounts are granted.<br>3. Selected customer exists in the system.  |
| Post-condition | Specified customer's information is updated   |
| Basic flow     | 1. Administrator click on view details button.<br>2. Selects the relevant student from the list.<br>3. Modifies relevant fields with new information.<br>4. Click "Update" button.                                    |
| Alternate path | <b>Not available</b>  |
| Exception      | 1. Don't have account<br><b>Solution:</b><br>a. Admin should make an account.   |
| Qualities      | <b>Performance:</b> Admin must open ATMS in 2 sec.<br><b>Correctness:</b> Updated information must be accurate and displayed consistently.<br><b>Security:</b> Only authorized admin can update user account details. |

**Use case 5:**

|                |  |
|----------------|--|
| Name           | Delete User Details  |
| Actor          | Admin  |
| Goal           | As an administrator, I want to delete user information   |
| Trigger        | When the administrator selects a student and chooses "Delete" option   |
| Pre-condition  | <ol style="list-style-type: none"><li>1. Administrator is logged in to the system.</li><li>2. Access rights to manage users accounts are granted.</li><li>3. Selected student exists in the system.</li></ol>  |
| Post-condition | Specified user information is deleted  |
| Basic flow     | <ol style="list-style-type: none"><li>1. Administrator click on view details button.</li><li>2. Selects the relevant student from the list.</li><li>3. Click "Delete" button.</li></ol>  |
| Alternate path | Not available  |
| Exception      | <p>1. Don't log-in to account<br/><b>Solution:</b><br/>a. Log-in to account</p> <p>2. Admin don't have right to manage student account<br/><b>Solution:</b><br/>a. Get access right to manage student account</p> <p>3. User don't exist<br/><b>Solution :</b><br/>a. Make sure that the specific student exists</p> |
| Qualities      | <p><b>Performance:</b> Delete operation should be completed within 10 seconds.</p> <p><b>Correctness:</b> Account must be delete after clicking delete button..</p> <p><b>Security:</b> Only authorized administrators can delete student account details.</p>   |

**User:****Use case 1:**

|                |   |
|----------------|---|
| Name           | LOG-in to account   |
| Actor          | USER  |
| Goal           | As a user ,I want to log-in to ATM Management System  |
| Trigger        | When a user clicks log-in button  |
| Pre-condition  | 1. Must have an account   |
| Post-condition | User has logged-in  |
| Basic flow     | 1. Open System<br>2. Enter account no<br>3. Enter password<br>4. Click on log-in button   |
| Alternate path | Not available   |
| Exception      | 1. Don't have account<br><b>Solution:</b><br>a. User should talk to admin   |
| Qualities      | <b>Performance:</b> user must open System in 2 sec.<br><b>Correctness:</b> System must be open after logged-in<br><b>Security:</b> account must be secure |

**Use case 2:**

|                |   |
|----------------|---|
| Name           | Withdraw Money  |
| Actor          | User  |
| Goal           | As a user ,I want to withdraw money   |
| Trigger        | When user click withdraw button   |
| Pre-condition  | 1. Must have account<br>2. Must be logged-in to system<br>3. Balance must be sufficient |
| Post-condition | User has withdrawn money  |
| Basic flow     | 1. Click on withdraw money option<br>2. Enter amount<br>3. Click withdraw button        |

|                |   |
|----------------|---|
| Alternate path | Not available   |
| Exception      | 1.Don't have account<br><b>Solution:</b><br>a. User should talk to admin<br>2. Don't logged-in to account<br><b>Solution:</b><br>a. User should log-in                        |
| Qualities      | <b>Performance:</b> user must open System in 2 sec.<br><b>Correctness:</b> money must be deposit after clicking on withdraw button<br><b>Security:</b> account must be secure |

### Use case 3:

|                |  |
|----------------|--|
| Name           | Deposit Money  |
| Actor          | User   |
| Goal           | As a user ,I want to deposit money   |
| Trigger        | When user click deposit button   |
| Pre-condition  | 4. Must have account<br>5. Must be logged-in to system   |
| Post-condition | User has withdrawn money   |
| Basic flow     | 4. Click on deposit option<br>5. Enter amount<br>6. Click deposit button   |
| Alternate path | Not available  |
| Exception      | 1.Don't have account<br><b>Solution:</b><br>a. User should talk to admin<br>2. Don't logged-in to account<br><b>Solution:</b><br>a. User should log-in<br>3.Don't have sufficient balance<br><b>Solution:</b><br>a.Make sure the balance sufficiency |

|           |   |
|-----------|---|
| Qualities | <b>Performance:</b> user must open System in 2 sec.<br><b>Correctness:</b> money must be deposit after clicking on withdraw button<br><b>Security:</b> account must be secure |
|-----------|---|

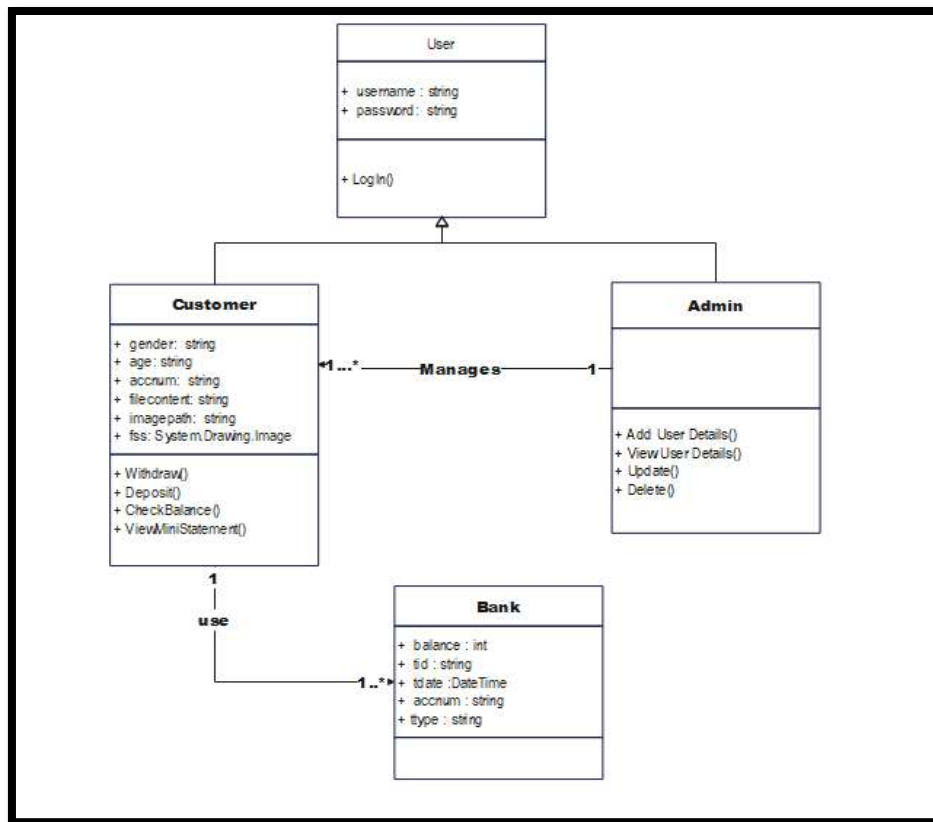
**Use case 4:**

|                |   |
|----------------|---|
| Name           | Check balance   |
| Actor          | User  |
| Goal           | As a user ,I want to check balance  |
| Trigger        | When user click deposit button  |
| Pre-condition  | 6. Must have account<br>7. Must be logged-in to system  |
| Post-condition | User has withdrawn money  |
| Basic flow     | 1.Click on balance option   |
| Alternate path | Not available   |
| Exception      | 1.Don't have account<br><b>Solution:</b><br>a. User should talk to admin<br>2. Don't logged-in to account<br><b>Solution:</b><br>a.User should log-in                         |
| Qualities      | <b>Performance:</b> user must open System in 2 sec.<br><b>Correctness:</b> balanced must be shown after clicking on balance option<br><b>Security:</b> account must be secure |

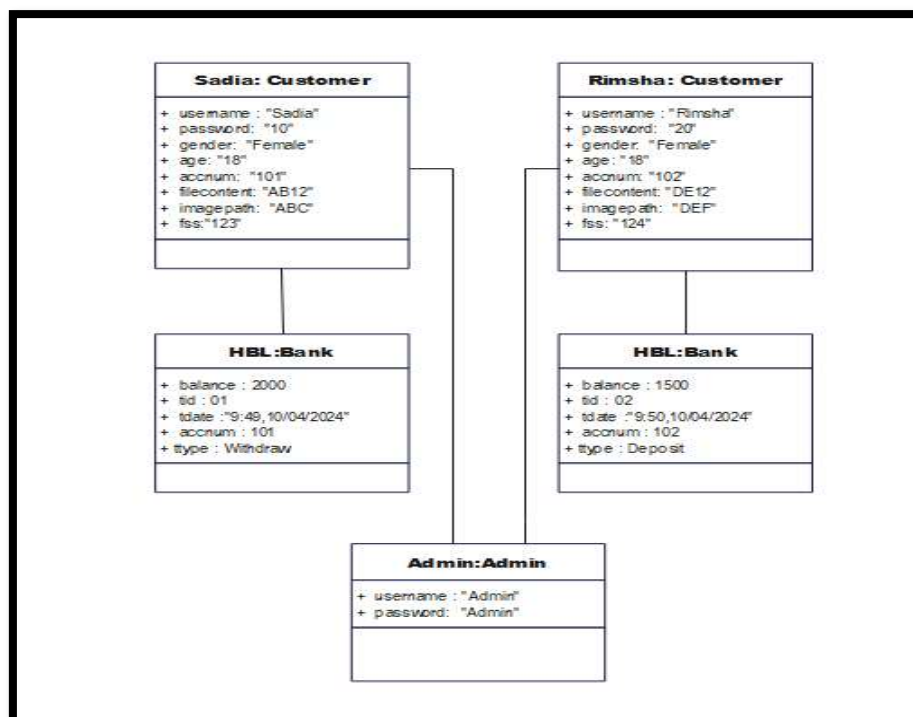
**Use case 5:**

|                |   |
|----------------|---|
| Name           | View statement  |
| Actor          | User  |
| Goal           | As a user ,I want to view statement   |
| Trigger        | When user click statement button  |
| Pre-condition  | 1.Must have account<br>2.Must be logged-in to system  |
| Post-condition | User has viewed statement   |
| Basic flow     | 1.Click on statement option   |
| Alternate path | Not available   |
| Exception      | 1.Don't have account<br><b>Solution:</b><br>a. User should talk to admin<br>2. Don't logged-in to account<br><b>Solution:</b><br>2. User should log-in                        |
| Qualities      | <b>Performance:</b> user must open System in 2 sec.<br><b>Correctness:</b> money must be deposit after clicking on withdraw button<br><b>Security:</b> account must be secure |

## Class Diagram:



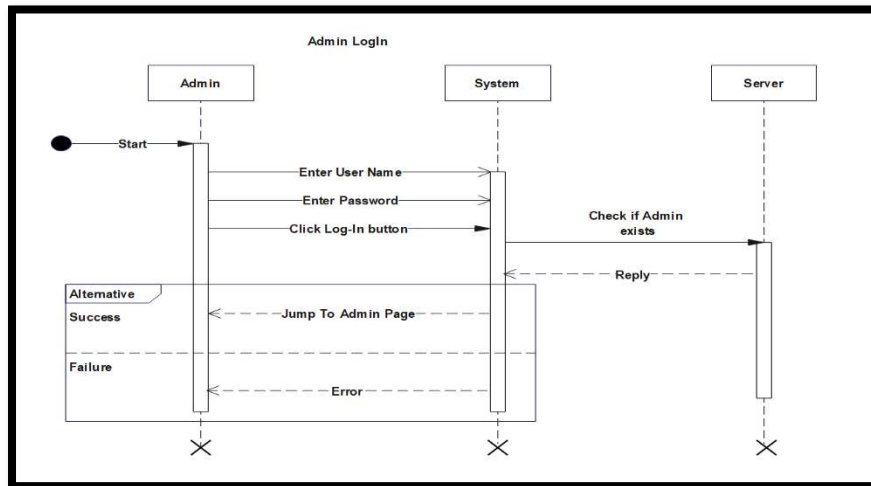
## Object Diagram:



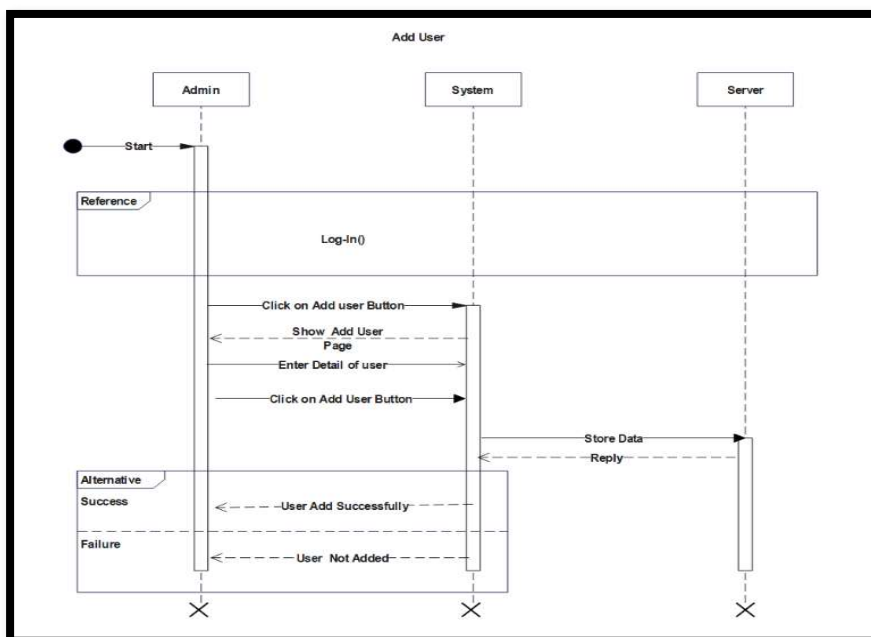
## Sequence Diagram:

### Admin Functions:

#### Log-in:

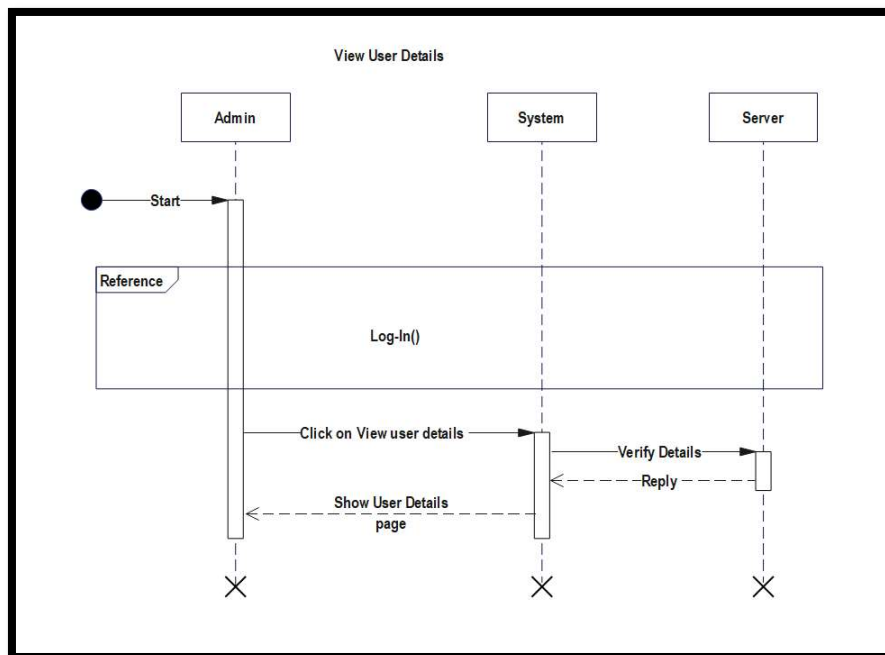


#### Add users:

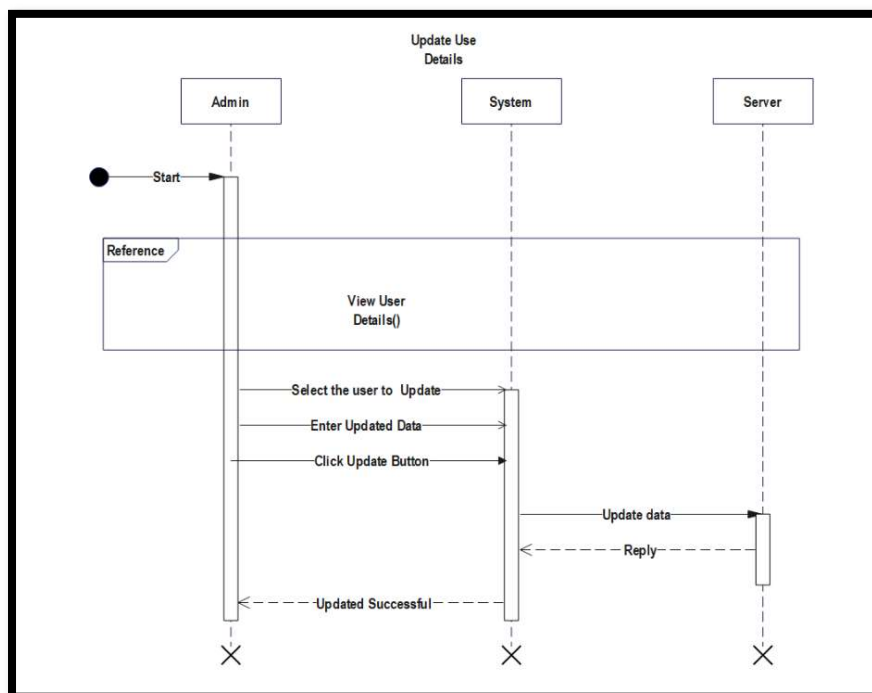




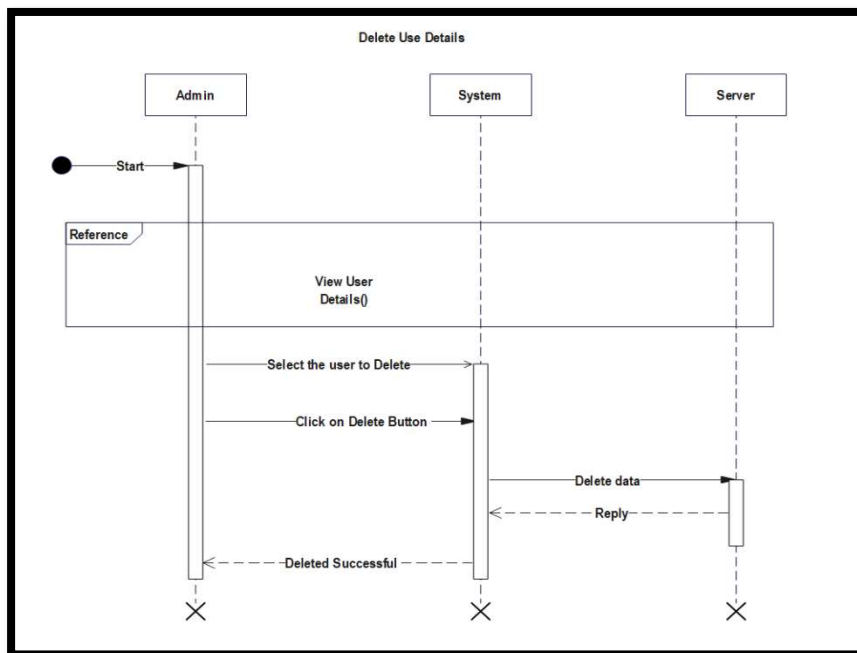
## View User Detail:



## Update User Detail:

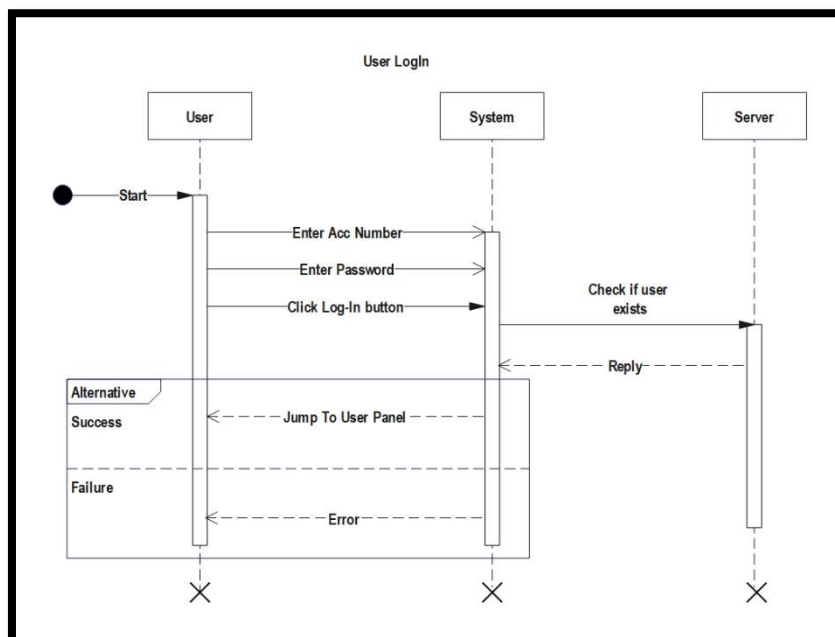


## Delete User Detail:

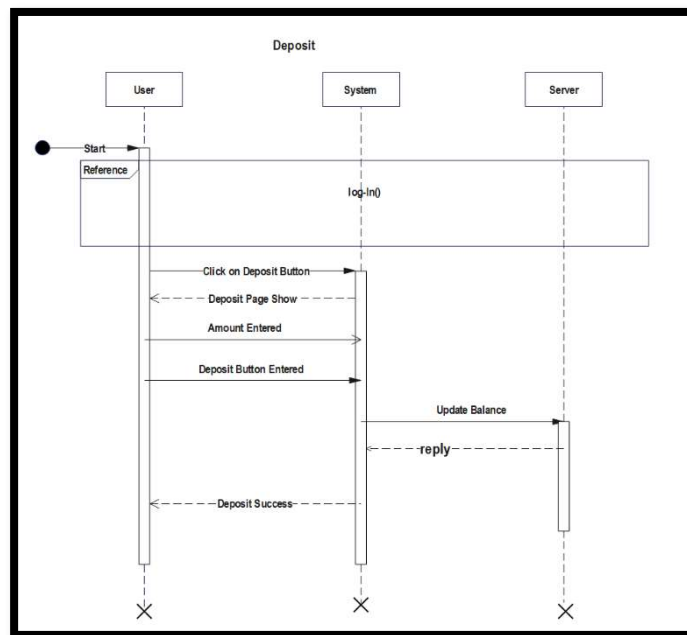


## User Functions:

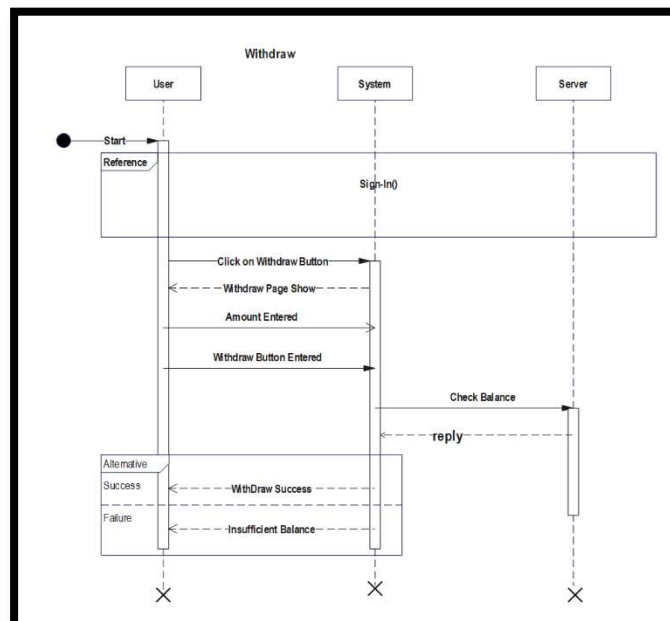
### Log-in:



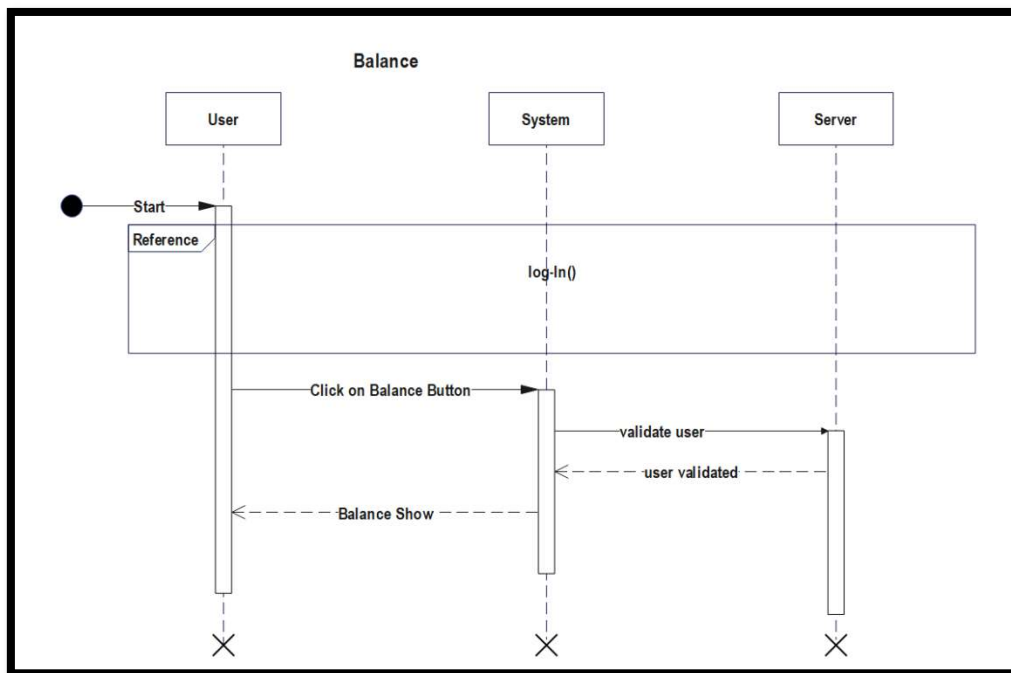
## User deposit money:



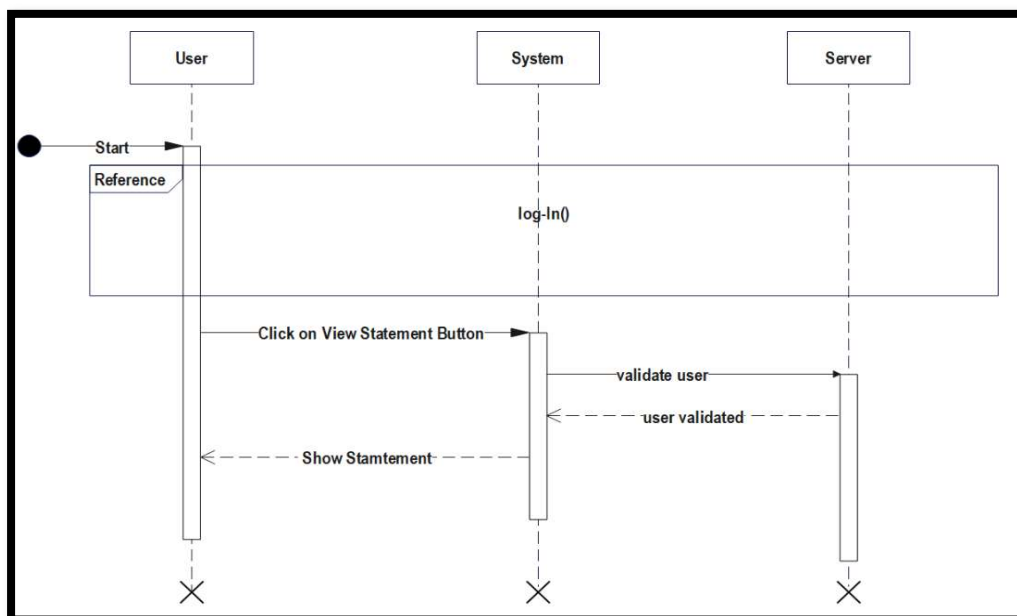
## User withdraw money:



### Check balance:



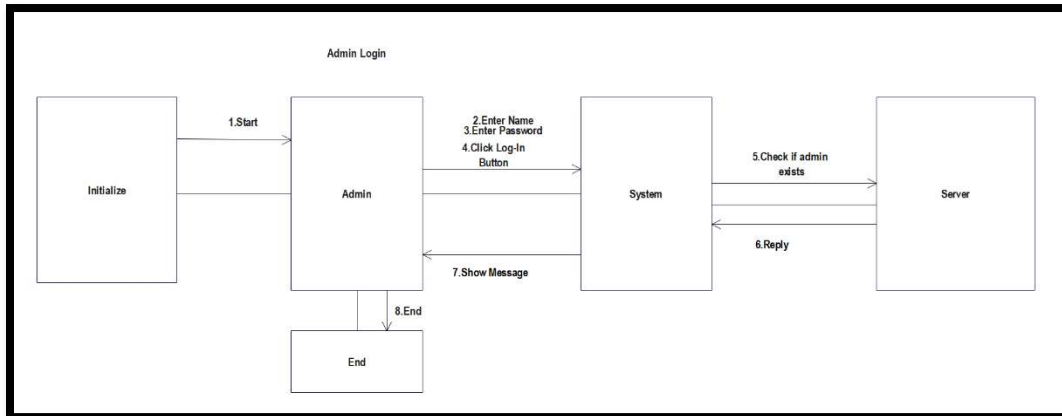
### View statement:



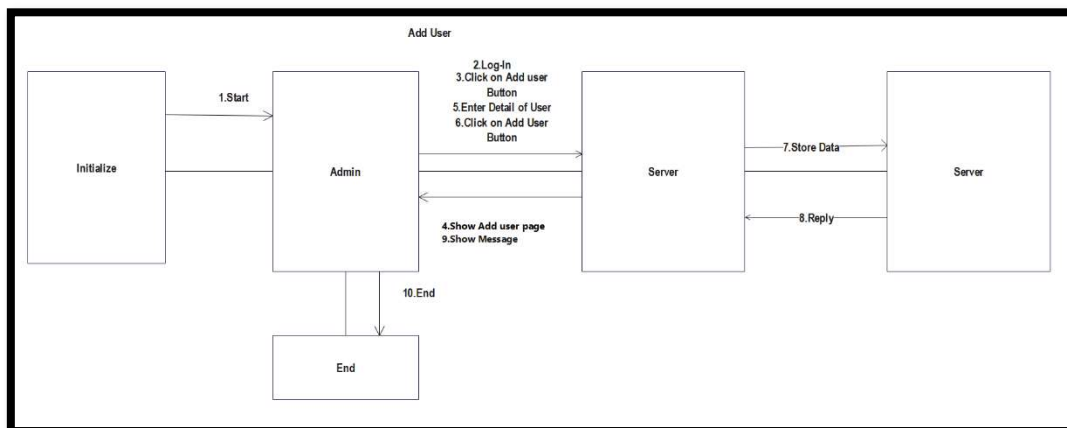
## Collaboration Diagram:

### Admin Functions:

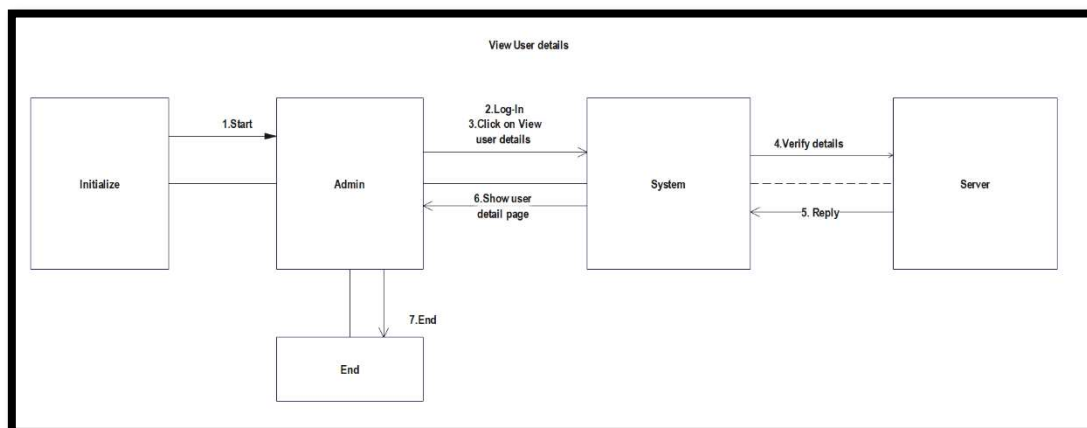
#### Log-in:



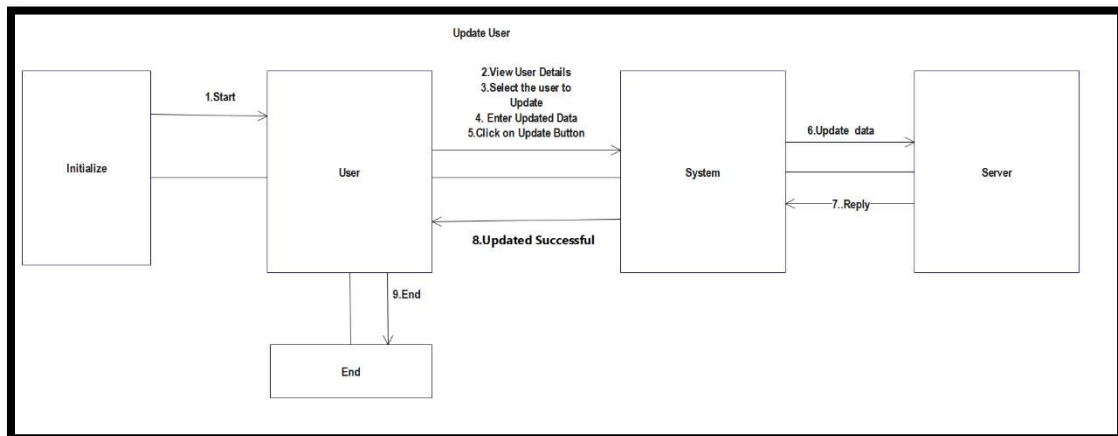
#### Add users:



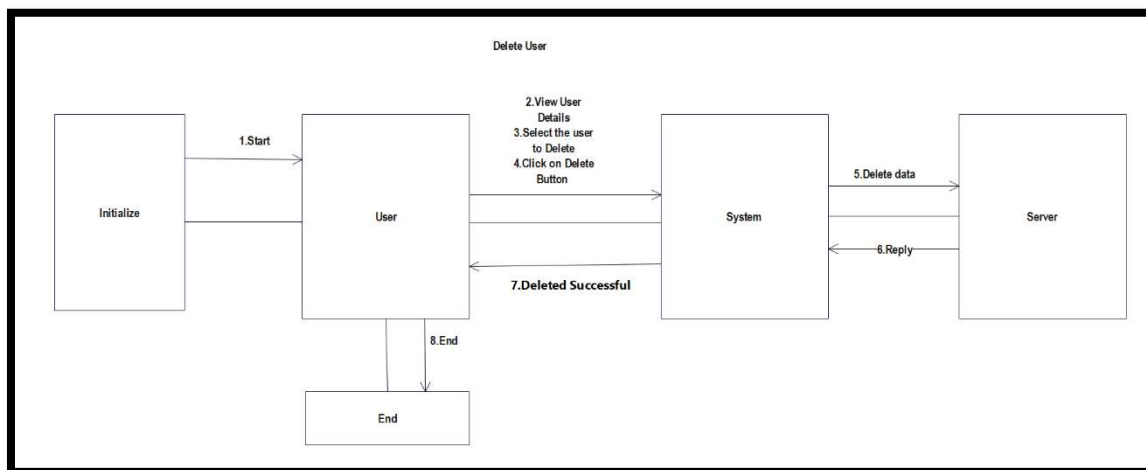
#### View User Detail:



## Update User Detail:

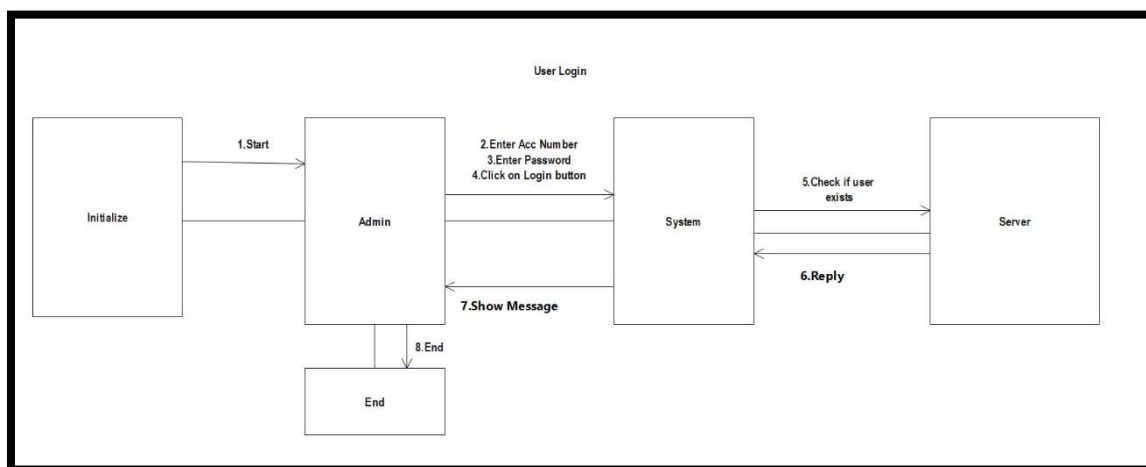


## Delete User Detail:

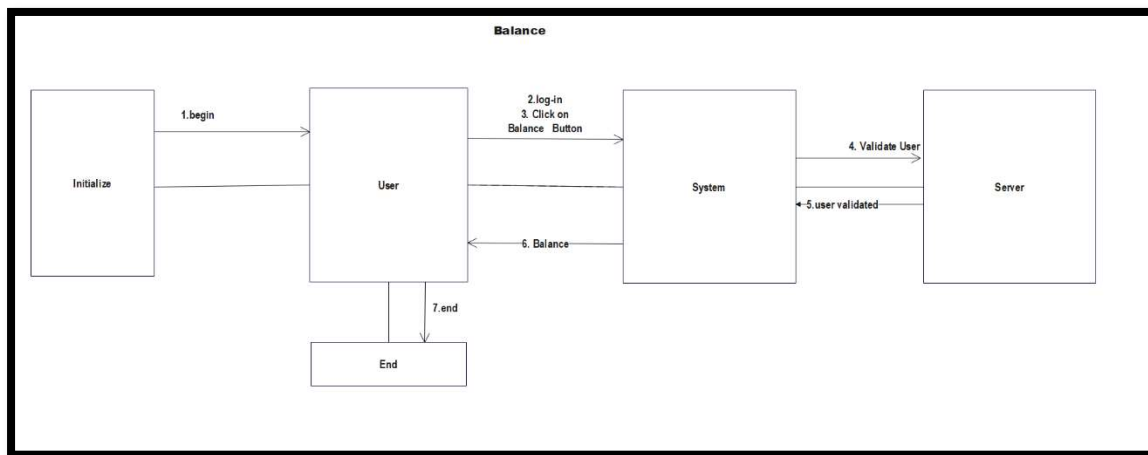


## User Functions:

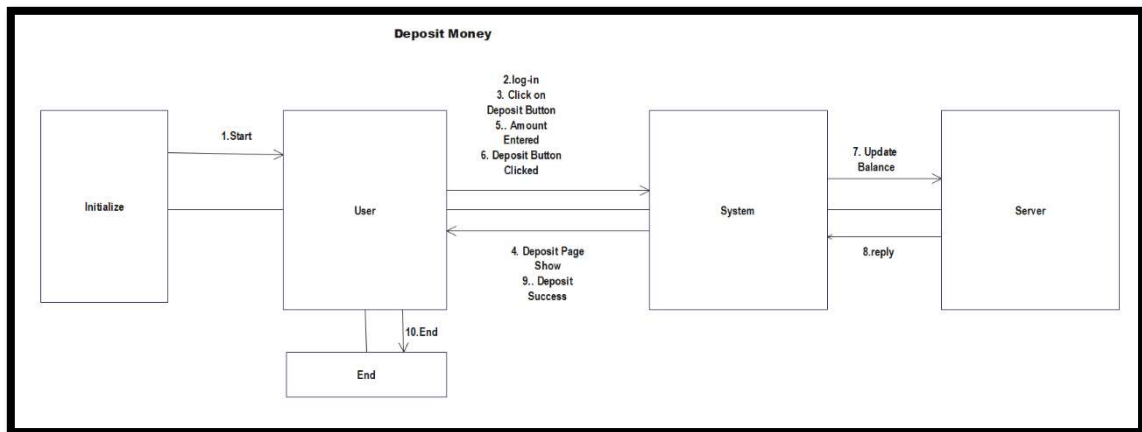
### Log-in:



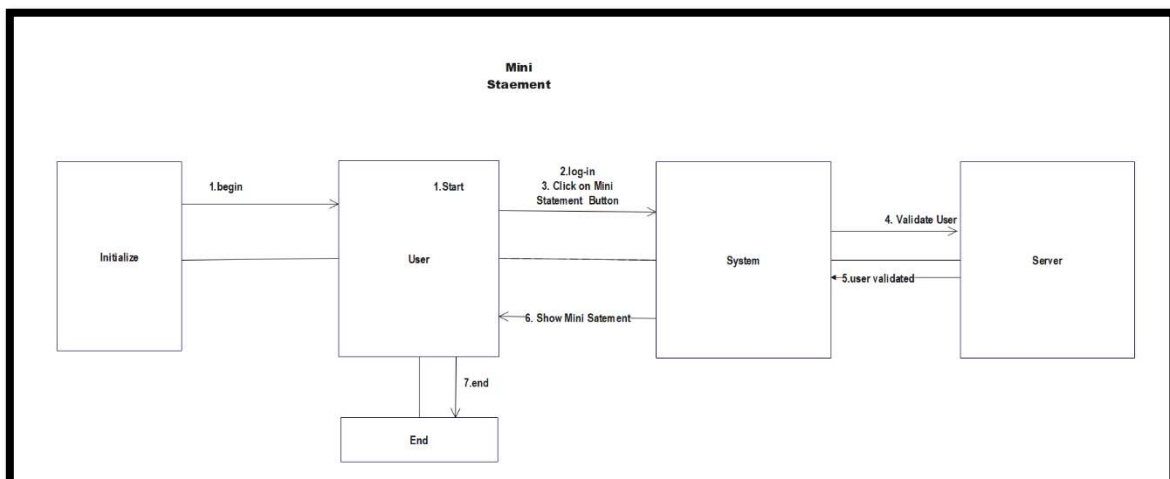
### Check balance:



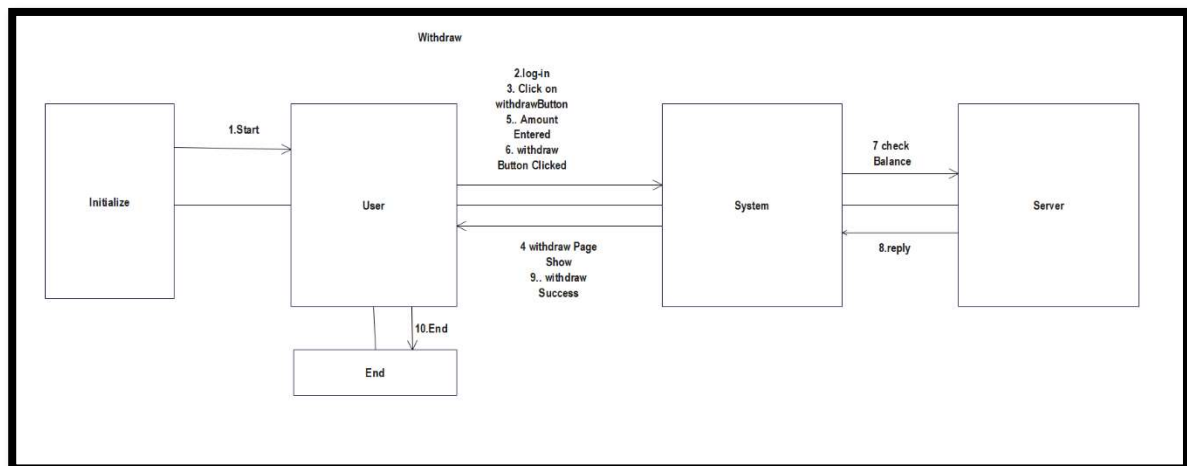
### User deposit money:



### View statement:

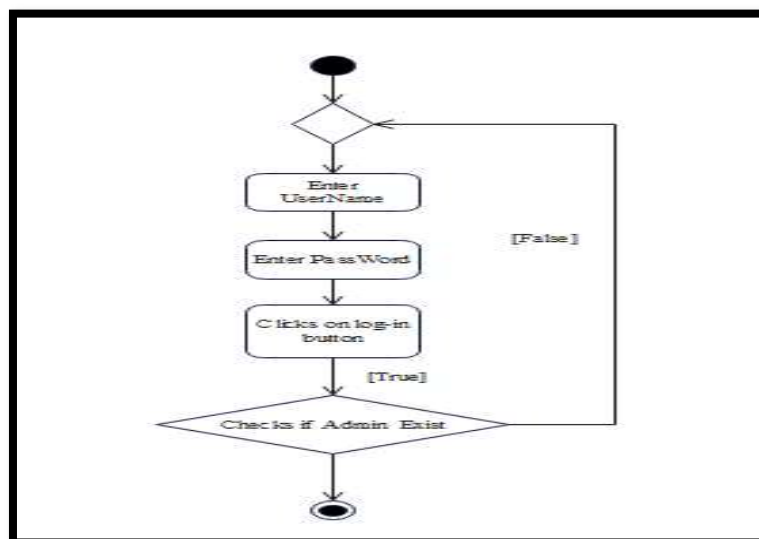


## User withdraw money:

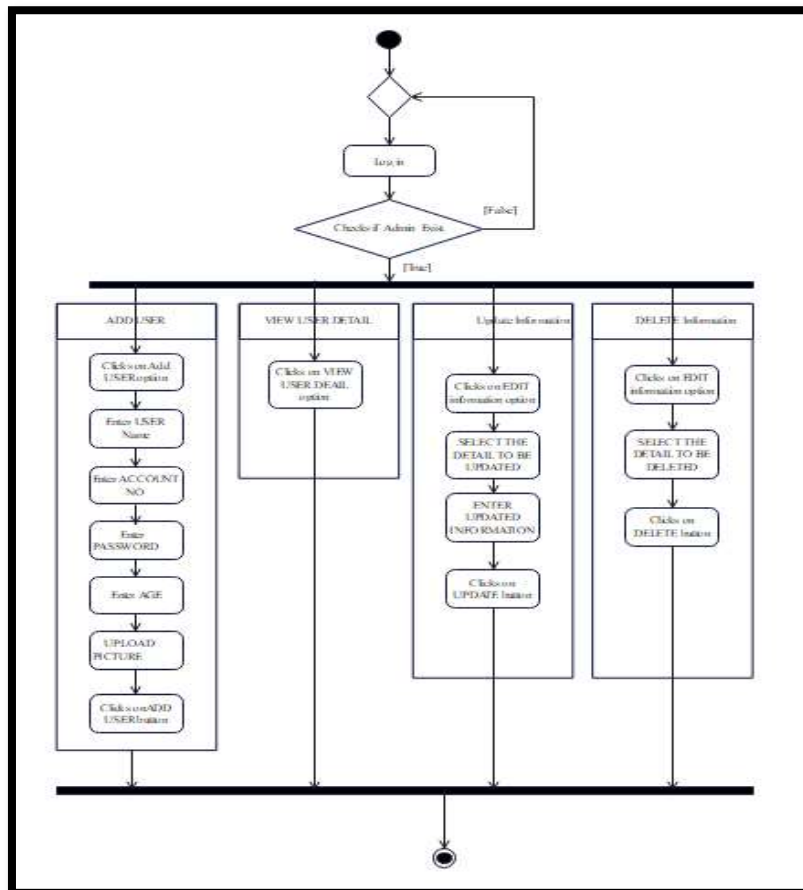


## Activity Diagram:

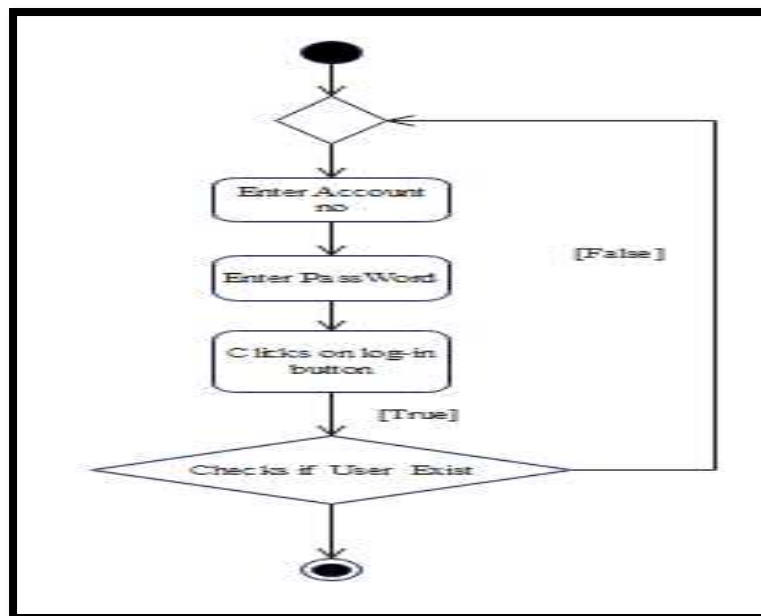
### Admin Functions:

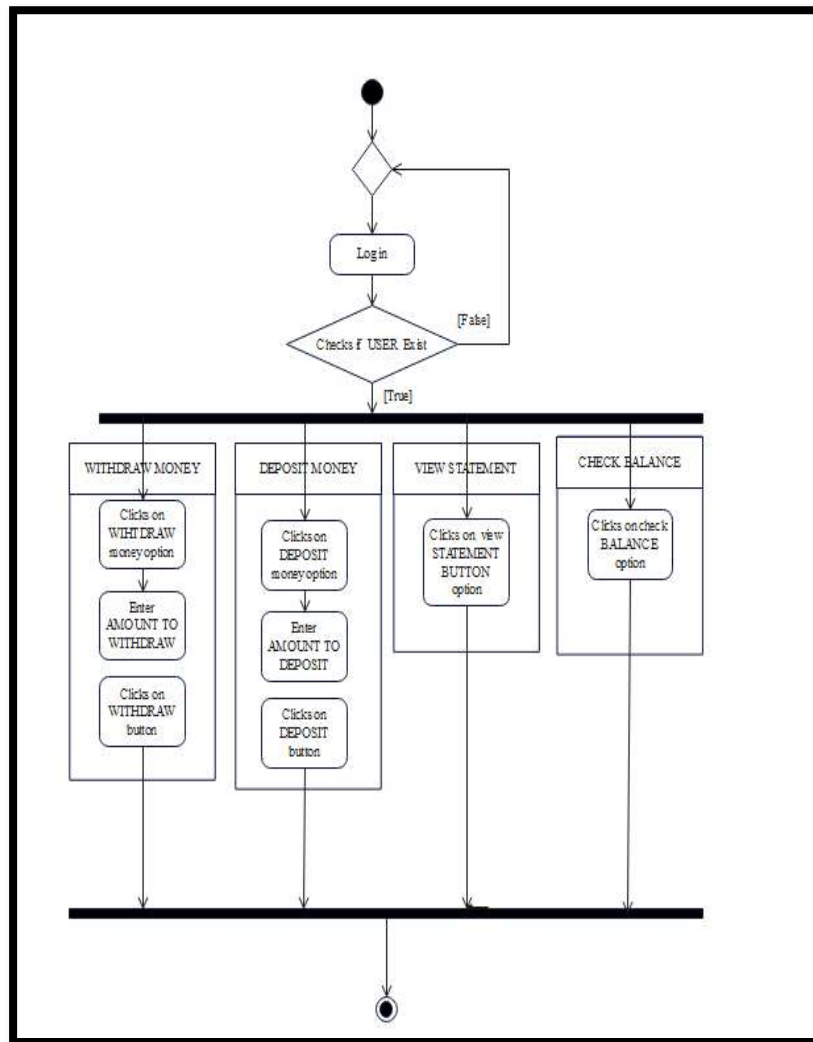






## User Functions:

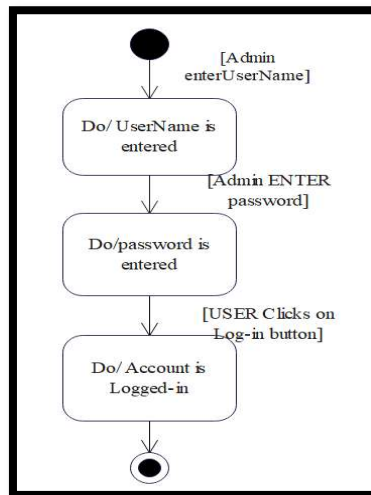




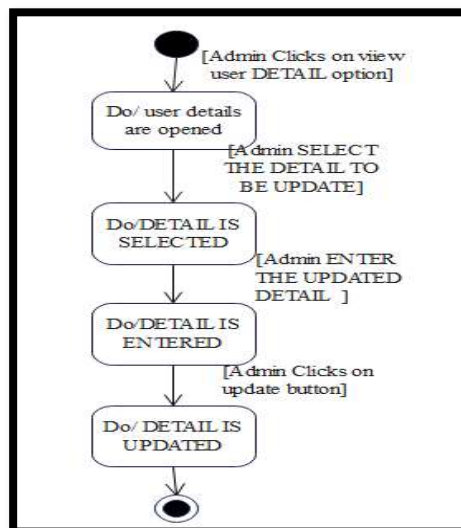
**State Machine Diagram:**

**Admin Functions:**

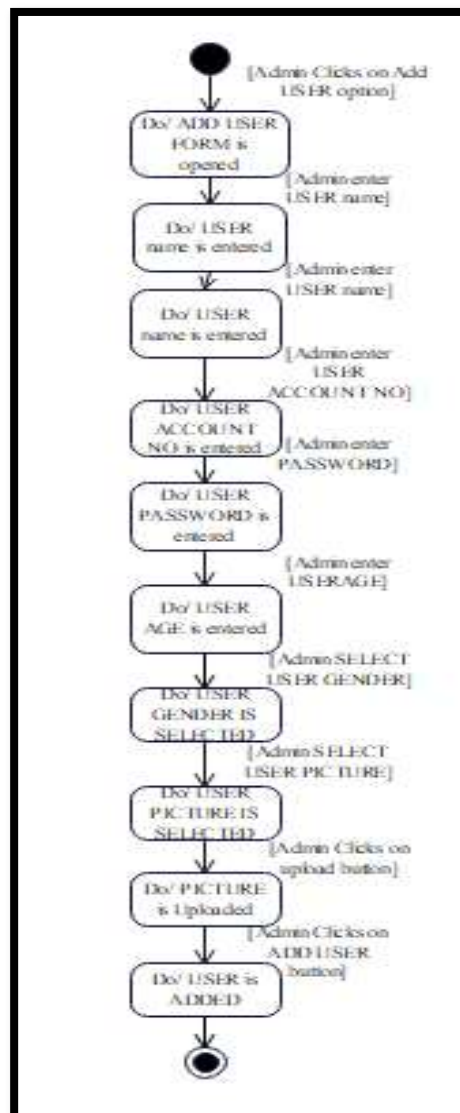
**Log-in:**



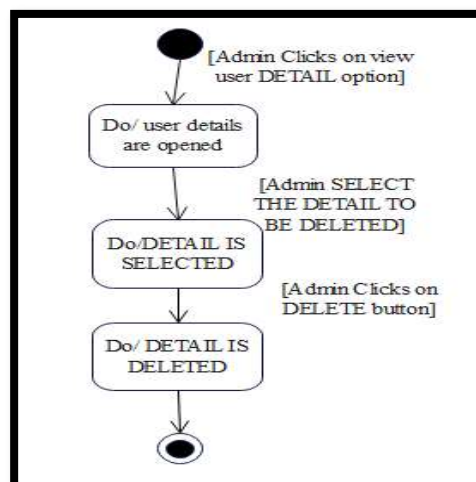
### Update User Detail:



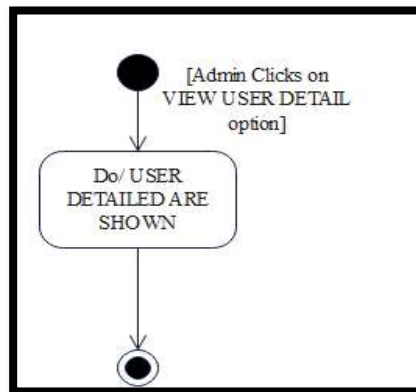
## Add users:



## Delete User Detail:

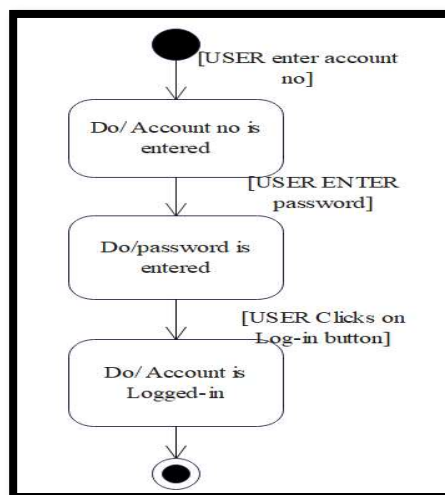


### View User Detail:

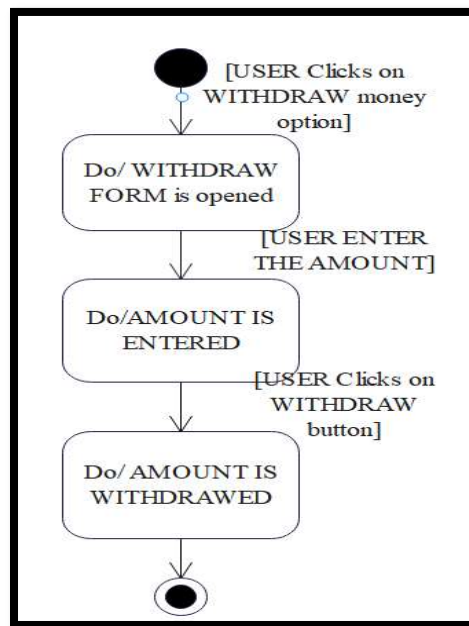


### User Functions:

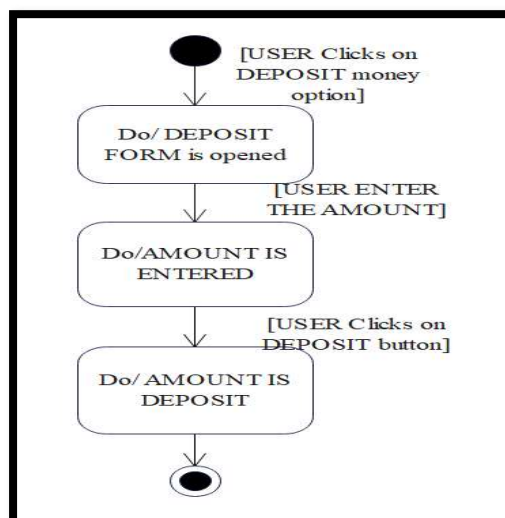
#### Log-in:



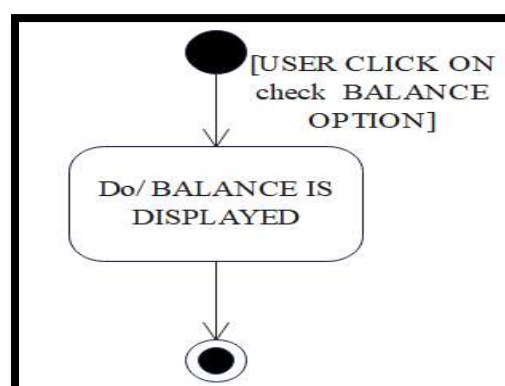
### User withdraw money:



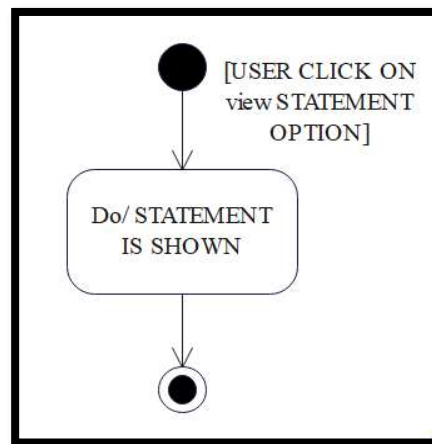
### User deposit money:



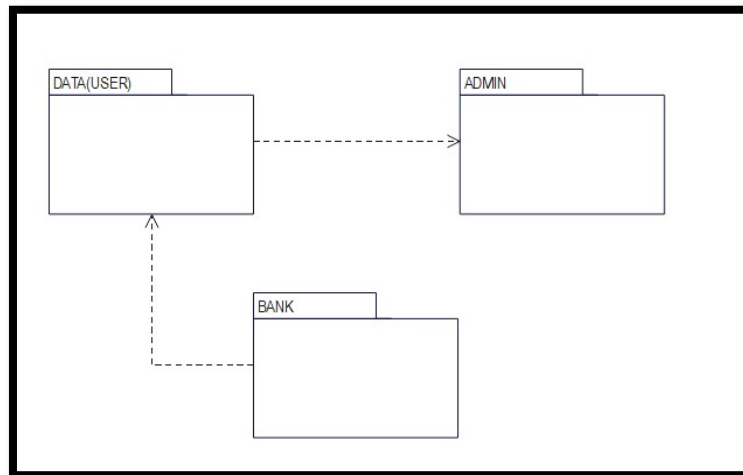
### Check balance:



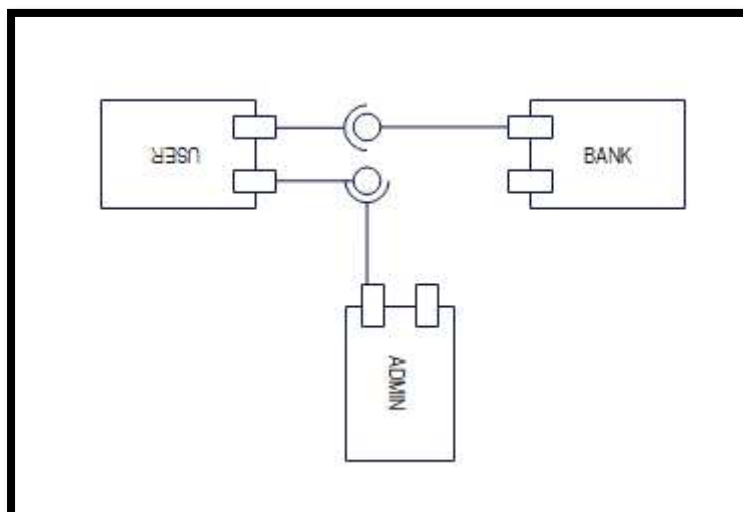
### View statement:



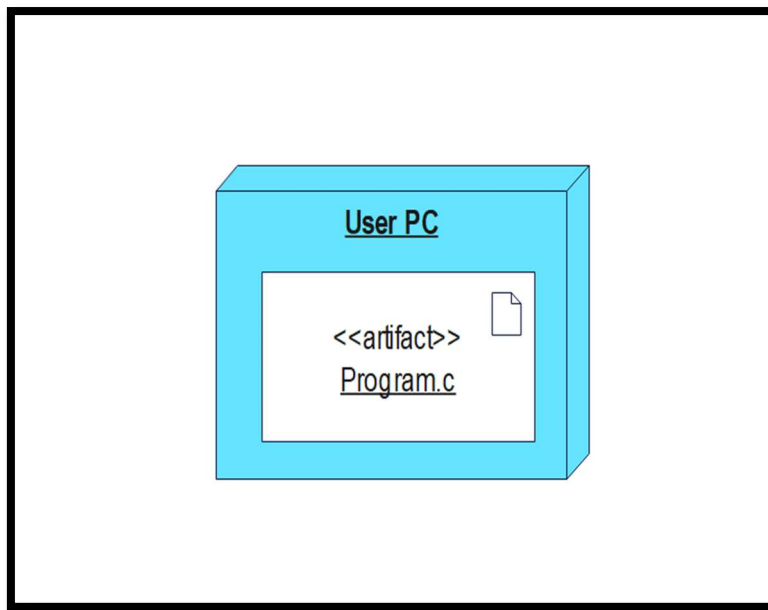
### Packages Diagram:



### Component Diagram:



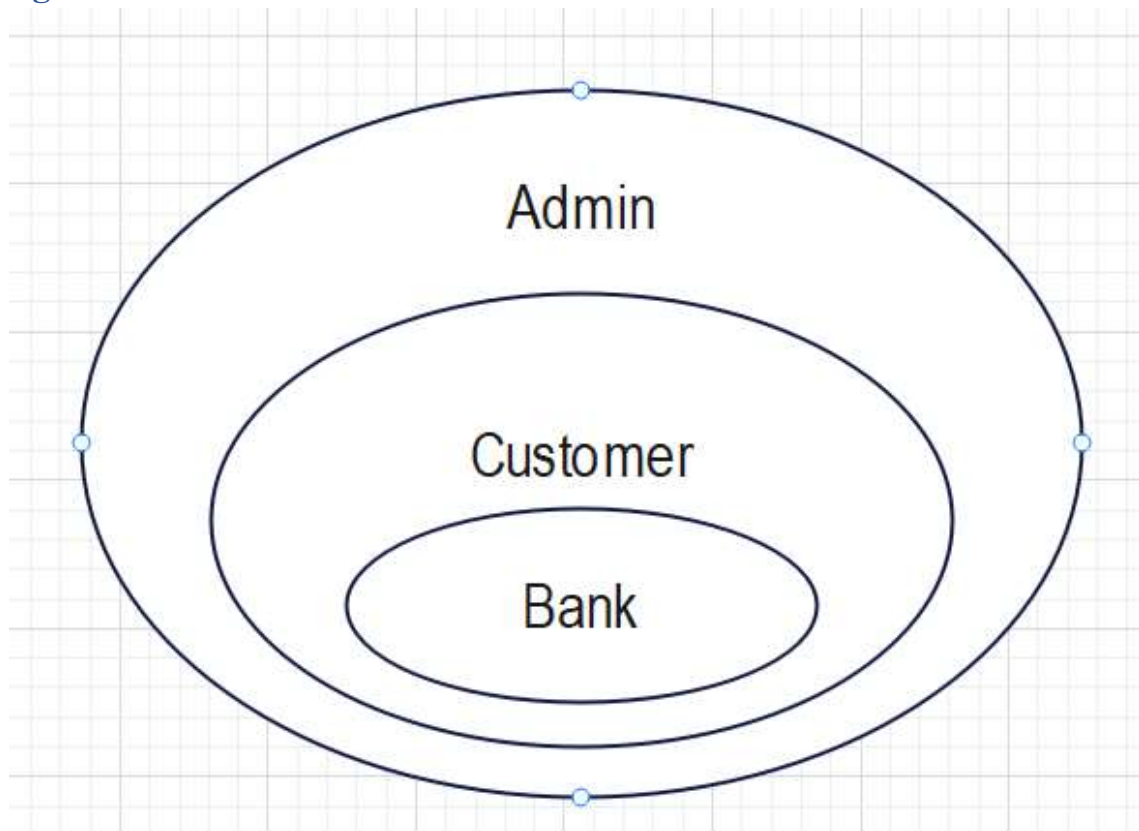
### Deployment Diagram:



### SYSTEM ARCHITECTURE:

We used layered architecture in our system.

### Diagram:





## Advantages:

- Layers are highly cohesive and promote information hiding.
- Layers are not strongly coupled to layers above them, reducing overall coupling.
- Layers help decompose programs, reducing complexity.
- Layers are easy to alter or fix by replacing entire layers, and easy to enhance by adding functionality to a layer.
- Layers are usually easy to reuse.

## Drawbacks:

- Debugging through multiple layers can be difficult.
- Getting the layers right can be difficult.
- Layer constraints may have to be violated to achieve unforeseen functionality

=====