

---

# **Software Requirements and Design Document**

**for**

# **Bank Management System**

**Prepared by Rafay Khattak, Waseem Akhtar and Hashir Saeed**

**Nexera Financial - Revolutionizing Financial Operations for Enhanced Efficiency**

**27<sup>TH</sup> November 2024**

# Table of Contents

<b>Table of Contents.....</b>	<b>ii</b>
<b>1. Introduction.....</b>	<b>3</b>
1.1 Purpose.....	3
1.2 Product Scope.....	3
1.3 Title.....	3
1.4 Objectives.....	3
1.5 Problem Statement.....	3
<b>2. Overall Description.....</b>	<b>4</b>
2.1 Product Perspective.....	4
2.2 Product Functions.....	4
2.3 List of Use Cases.....	4
2.4 Extended Use Cases.....	4
2.5 Use Case Diagram.....	20
<b>3. Other Nonfunctional Requirements.....</b>	<b>21</b>
3.1 Performance Requirements.....	21
3.2 Safety Requirements.....	22
3.3 Security Requirements.....	22
3.4 Software Quality Attributes.....	22
3.5 Business Rules.....	22
3.6 Operating Environment.....	23
3.7 User Interfaces.....	23
<b>4. Domain Model.....</b>	<b>23</b>
<b>5. System Sequence Diagram.....</b>	<b>24</b>
<b>6. Sequence Diagram.....</b>	<b>35</b>
<b>7. Class Diagram</b>	
<b>8. Package Diagram</b>	
<b>9. Deployment Diagram</b>	

# Introduction

## Purpose

The **Bank Management System (BMS)** is designed as a cross-platform application that ensures seamless operation across multiple operating systems using a unified codebase. Leveraging **Java**, **JavaFX**, and **MySQL**, the system provides robust performance, scalability, and security. Its architecture ensures compatibility and functionality across diverse environments, meeting the needs of administrators and customers alike.

## Product Scope

The Bank Management System is tailored to meet the needs of a **single bank**, providing essential features for both administrators and customers. The system is structured to streamline operations by offering the following:

- **Admin Functions:** Real-time management of customer loan requests, customer profiles, and services.
- **Customer Functions:** Application for a single loan at a time and access to multiple banking services such as account management and transaction processing.

By focusing on these roles, the BMS aims to enhance efficiency, reduce operational delays, and deliver an improved banking experience.

## Title

Bank management system.

## Objectives

The primary objectives of the Bank Management System are as follows:

1. **Intuitive User Interface:** Deliver a modern, interactive interface that is easy to navigate and understand for both administrators and customers.
2. **Efficient Data Handling:** Optimize database management by eliminating redundant file-based storage and instead using a centralized, secure database powered by MySQL Server Database.

3. **Fast Processing:** Ensure swift execution of banking operations, minimizing delays and enhancing user satisfaction.
4. **Cross-Platform Support:** Facilitate seamless functionality across multiple operating systems without altering the codebase, leveraging the cross-platform capabilities of Java and JavaFX.
5. **Simplified Banking Operations:** Streamline common banking tasks such as loan management and service requests, making the system highly accessible and user-friendly for both admins and customers.

## Problem Statement

Traditional banking systems heavily rely on maintaining hard copies of records, leading to significant challenges such as data loss, inefficiency, and vulnerability to mismanagement. The reliance on paper-based documentation not only increases operational costs but also poses security risks and makes data retrieval time-consuming.

This project aims to address these challenges by developing a **secure, reliable, and lossless digital banking solution**. By transitioning to a modernized software-based system, banks can eliminate the need for extensive physical document storage and adopt a streamlined approach to managing transactions and customer data. The proposed system will:

- **Prevent Data Loss:** Provide a robust and secure database that ensures all information is stored digitally, reducing the risk of misplacement or loss.
- **Efficient Transaction Management:** Store transactions in an organized and optimized format, facilitating faster and more accurate record-keeping.
- **Future-Proof Design:** Develop a system aligned with modern technology standards, ensuring it remains relevant and adaptable to future needs.

By implementing this solution, banks can significantly enhance their operational efficiency, security, and long-term sustainability, moving away from outdated practices and embracing the digital era.

## Overall Description

### Product Perspective

The **Bank Management System (BMS)** is a modern, self-contained software solution developed to revolutionize and optimize banking operations. Built with **Java**, **JavaFX**, and **MySQL**, the system addresses the limitations of legacy banking practices, such as reliance on physical records and manual data management.

The BMS is designed as a **flexible and scalable solution** that can seamlessly integrate with existing banking environments. Its architecture ensures compatibility with other financial systems, regulatory compliance tools, and customer relationship management (CRM) software, enabling a unified and efficient banking experience.

Key features of the system include:

- **User Roles and Interfaces:** Tailored dashboards and functionalities for both administrators and customers, allowing admins to manage operations such as account approvals, loan processing, and reports, while customers can access account details, apply for loans, and manage transactions.
- **Core Subsystems:** The system is divided into dedicated modules, including:
  - **User Authentication:** Secure login and identity verification for admins and customers.
  - **Loan Management:** Efficient handling of loan applications, approvals, and repayments.
  - **Account Management:** Centralized control over account creation, updates, and closures.
  - **Transaction Processing:** Real-time processing and tracking of deposits, withdrawals, and transfers.
  - **Database Management:** A robust MySQL-based database ensuring data integrity, security, and scalability.

The BMS is not only designed to **streamline banking operations** but also to ensure compliance with modern financial and security standards. Its scalability makes it suitable for deployment across multiple branches, enabling synchronized operations within a bank's network. By automating core banking processes, the BMS significantly reduces operational overhead while enhancing customer satisfaction and data accuracy.

### Product Functions

- Managing loan applications and applying for loans
- withdrawal of cash and deposit of cash
- viewing relative information
- documentation
- tabular displays where necessary
- authentication and validation of entry

## List of Use Cases

- Withdraw Cash
- Deposit Cash
- Validation
- Apply For Loans
- Loan Status
- Manage Accounts
- Manage Own Account
- Check Account Information
- Check Loan Information
- Generate All Transactions Report
- Manage Loan Applications
- Check For Current Balance
- View Transaction History

## Extended Use Cases

<b>Use Case Name:</b> Make Transaction	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Customer	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>- Customer Making the payment to acquire product/service of need/want</li> <li>- Acquiring Bank that will receive the payment capital</li> <li>- Customer's Bank which will be managing the transactions through bank system</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- Customer wants to make purchase</li> <li>- Customer has enough balance to make purchase</li> <li>- System is active</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- Customer has made payment and transaction is successful</li> <li>- Customer receives purchased item/service</li> <li>- System records updated</li> </ul>	
<b>Main Success Scenario:</b>	
Customer	System
1. Customer has items/service he wants to buy	
2. Customer Arrives at check-out with items	
3. Cashier scans items	
	4. System displays price of item
	5. System adds item to total
	6. System displays total
7. Cashier tells Customer to make payment	
8. Customer chooses means to make payment (cash or bank card)	
	9. Payment is made and system balance updated
	10. System generates receipt
11. Customer receives purchased items with receipt and leaves	
	12. System inventory updated

**Extensions:**

1. Customer has not acquired the items he wants to buy
2. Customer fails to arrive at check-out
- 3.



- a) Cashier is not present at check-out
- b) System is inactive
- c) System fails to scan item as scanner is damaged
- d) Item scan code is damaged
- 4.
  - a) System displays wrong price
  - b) Price of item not in system or is invalid
- 5. System fails to add to total
- 6. System fails to display total correctly
- 8. Customer does not have enough balance to make payment 9.
  - a) System is inactive
  - b) payment failed
  - c) bank card declined
  - d) cash is fake
  - e) cash is damaged
- 10. System fails to generate receipt
- 11. Customer does not receive purchased item/service/receipt
- 12. System inventory not updated

<b>Use Case Name:</b> Check Balance	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Customer	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>- Customer has satisfaction with having plenty of money or knowing how much more they need to earn</li> <li>- Bank with having more capital in the bank vault</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- System is active</li> <li>- Customer is logged in</li> <li>- Customer has account/s</li> <li>- Customer wants to check balance</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- Customer attains information about the balance state of his account and leaved</li> <li>- Customer is logged out of system</li> </ul>	
<b>Main Success Scenario:</b>	
Customer	System
1. Customer arrives at system	
2. Customer logs in to account	
	3. System displays menu
4. Customer chooses to check balance	
	5. System goes to balance page and displays customer balance



6. Customer acquires information regards the balance in his account	
7. Customer logs out of system and leaves	
	8. System becomes inactive when not in use
<b>Extensions:</b> <ol style="list-style-type: none"> <li>1. System is not active</li> <li>2. Login failure                             <ol style="list-style-type: none"> <li>a) Due to wrong password</li> <li>b) Due to poor internet connection</li> <li>c) Due to failure of system response</li> </ol> </li> <li>3. System fails to load menu</li> <li>4. Customer can not see checking balance in menu</li> <li>5. System fails to load balance page/ shows wrong balance</li> <li>6. Customer can not spot where balance is written on the page</li> <li>7. Customer does not log out of account</li> </ol>	

<b>Use Case Name:</b> Display Account Information/History	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Customer	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>- Customer has the interest to know if account details are correct</li> <li>- Customer/bank admin may want to view account history of use and logins</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- System is active</li> <li>- System has option to display account information/history</li> <li>- Customer wants to check information/history</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- Customer acquired information</li> </ul>	
<b>Main Success Scenario:</b>	
Customer	System
1. Customer arrives at system	
2. Customer logs into system	
	3. System displays menu
4. Customer chooses option to show information/history of account	
	5. System opens account information page
6. Customer manages/views account information	
	7. System updates records in case of changes

8. Customer logs out of system	
--------------------------------	--

	9. System becomes inactive when not in use
<b>Extensions:</b> <ol style="list-style-type: none"> <li>System is inactive so customer cannot use it</li> <li>Customer cannot login to system                             <ol style="list-style-type: none"> <li>Due to wrong password entry</li> <li>Due to poor internet connection</li> <li>Due to system failure</li> </ol> </li> <li>System fails to display menu</li> <li>System does not contain option to display account information/history</li> <li>System fails to open page</li> <li>Customer cannot view/manage account information</li> <li>System fails to update records</li> <li>Customer does not log out</li> </ol>	

<b>Use Case Name:</b> Transfer Amount	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Customer	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>Customer having interest in the amount of transfer</li> <li>Recipient having interest in receiving the amount</li> <li>Customer's bank having interest in the transfer of amount</li> <li>Recipient's bank having interest in handling the receiving of the amount</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>System is active</li> <li>Customer wants to transfer amount</li> <li>Customer can access the system</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>Transfer made successfully</li> <li>System is updated</li> </ul>	
<b>Main Success Scenario:</b>	
Customer	System
1. Customer arrives at system	
2. Customer logs into system	
	3. System displays menu
4. Customer chooses option to transfer amount	
	5. System opens page for transfer
6. Customer enters the details for transfer such as recipient account number and amount to transfer	
7. Customer presses enter	
	8. System manages the transfer and

	deducts amount from customer
--	------------------------------

9. Recipient receives amount	
10. Customer logs out of system	
	11. System becomes in active when not in use
<b>Extensions:</b> <ol style="list-style-type: none"> <li>1. Customer cannot access the system as it is inactive</li> <li>2. Customer fails to log into system                             <ol style="list-style-type: none"> <li>a) Due to wrong password</li> <li>b) Due to poor internet connection</li> <li>c) Due to failure of system</li> </ol> </li> <li>3. System fails to display menu</li> <li>4. Menu does not contain option to transfer amount</li> <li>5. System fails to open the page due to server failure or poor internet connection</li> <li>6. Customer cannot enter details into page</li> <li>7. Customer forgets to press enter</li> <li>8. System does not manage the transfer correctly</li> <li>9. Recipient does not receive amount</li> <li>10. Customer does not log out of system</li> </ol>	

<b>Use Case Name:</b> Make Withdrawal	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Customer	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>- Customer having interest in the amount of cash to withdraw</li> <li>- Bank having interest in the amount of cash is being taken from their vault/atm</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- System is active</li> <li>- Customer wants to make a withdrawal</li> <li>- System has enough cash for withdrawal</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- Customer leaves with correctly received cash</li> <li>- Customer account balance is updated</li> </ul>	
<b>Main Success Scenario:</b>	
Customer	System
1. Customer arrives at system	
2. Customer logs into account	
	3. System displays menu
4. Customer chooses option to withdraw cash	
	5. System displays information
6. Customer enter amount to withdraw	
	7. System asks for confirmation

8. Customer makes confirmation	
--------------------------------	--



	9. System manages withdrawal
	10. System updates customer balance
11. Customer receives cash of correct amount	
12. Customer logs out and leaves	
	13. System becomes inactive when not in use
<b>Extensions:</b> <ol style="list-style-type: none"> <li>Customer cannot access the system</li> <li>Customer fails to login</li> <li>System does not display menu</li> <li>There is no option to make withdrawal</li> <li>System fails to open the page for withdrawal</li> <li>Customer cannot enter amount</li> <li>System does not ask for confirmation and ends up withdrawing more/less cash than customer wanted</li> <li>Customer cancels confirmation</li> <li>System does not have enough cash to manage withdrawal or fails to withdraw with customer balance deducted</li> <li>System does not update customer balance</li> <li>Customer does not receive cash of correct amount</li> <li>Customer does not log out</li> </ol>	

<b>Use Case Name:</b> Track Loan Status/Progress	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Customer	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>Customer having interest in checking the amount of money he needs to pay the bank by a fixed date</li> <li>Bank having interest in receiving payments on time from customer</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>System is active</li> <li>Customer wants to track loan progress</li> <li>Customer has a loan in progress with the bank</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>Customer has identified how much is due and how much are paid already</li> <li>Bank has identified how much more money they will receive</li> </ul>	
<b>Main Success Scenario:</b>	
Customer	System
1. Customer arrives at system	
2. Customer logs into system	
	3. System displays menu

4. Customer chooses option to check loan progress	
---	--

	5. System open page to show loan progress information
6. Customer views information regard his loan and its dues	
7. Customer logs out	
	8. System is inactive when not in use
<b>Extensions:</b> <ol style="list-style-type: none"> <li>1. Customer fails to access system</li> <li>2. Customer fails to login to system</li> <li>3. System fails to load the menu</li> <li>4. There is no option to track loan progress</li> <li>5. System fails to load page of loan information</li> <li>6. Customer cannot see necessary information</li> <li>7. Customer does not log out</li> </ol>	

<b>Use Case Name:</b> Login of Customer/Admin	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Customer & Admin	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>- Customer having interest to be able to access the system</li> <li>- Admin having interest to be able to access the system</li> <li>- Bank having interest to be able to track the number of users they have</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- System is active</li> <li>- User has an account</li> <li>- Internet connection is good</li> <li>- User wants to use the system</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- User successfully accesses the system</li> </ul>	
<b>Main Success Scenario:</b>	
User	System
1. User arrives at the system	
2. User enters login information and presses enter	
	3. System validates login details against database of login records
	4. System logs in user into its system
5. User can access system options relevant to him/her	
<b>Extensions:</b> <ol style="list-style-type: none"> <li>1. User cannot access the system</li> <li>2. User cannot enter details</li> <li>3. User enters invalid details</li> </ol>	

4. System does not log user into its system

## 5. User cannot access his relevant options

**Use Case Name:** Manage Account

**Scope:** Bank Management System

**Level:** Usergroup

**Primary Actor:** Customer & Admin

**Stakeholders & Interests:**

- Customer having interest in managing his/her checking/saving account
- Admin having interest in managing accounts of all users
- Bank having interest in the number of accounts held by them

**Precondition:**

- System is active
- User has account/s
- User wants to manage account/s

**Postcondition:**

- Successfully manages accounts

**Main Success Scenario:**

User	System
1. User arrives at system	
2. User logs into system	
	3. System displays options menu
4. User chooses option to manage account	
	5. System opens account management page
6. User manages his/her account/s	
	7. System updates records according to user changes
8. User logs out of system when done using	
	9. System is inactive when not in use

**Extensions:**

1. User cannot access system
2. User fails to login to system
3. System fails to load page after login
4. There is no option to choose account management
5. System does not open page due to server or network issue
6. Customer cannot make changes to his/her account/s
7. System does not update records
8. Customer forgets to logout

**Use Case Name:** Apply for Loan

**Scope:** Bank Management System

**Level:** Usergroup

<b>Primary Actor:</b> Customer & Admin
<b>Stakeholders &amp; Interests:</b>

<ul style="list-style-type: none"> <li>- Customer having interest in wanting a loan</li> <li>- Admin having interest in providing a loan which will profit the bank</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- System is active</li> <li>- Customer wants a loan</li> <li>- Customer has good financial status for loan</li> <li>- Admin can hand such loan profitably</li> <li>- Bank has enough capital to give the loan</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- Customer receives money on loan</li> <li>- System is updated</li> </ul>	
<b>Main Success Scenario:</b>	
User	System
1. Customer arrives at system	
2. Customer logs into system	
	3. System displays menu
4. Customer chooses option for loan	
	5. System open page for loan application
6. Customer fills form to apply for loan	
7. Customer presses enter to file for loan	
	8. System add record to filed loans
9. Admins logs into system	
10. Admin accesses customer loan application	
11. Admin check customer financial background to see if he will be able to repay the loan	
12. Admins check if bank can give out the amount on loan	
13. Admin chooses relevant interest rate and duration of loan	
14. Admin enters loan information for the customer into the system	
15. Admin logs out of system	
	16. System updated records for the user
	17. System adds loan amount into customer balance and deduct's amount from banks own balance





18. Customer receives message of approval with the balance updated with the loan amount	
19. Customer logs out of system	
	20. System is inactive when not in use
<p><b>Extensions:</b></p> <ol style="list-style-type: none"> <li>1. Customer fails to access system</li> <li>2. Customer cannot login to system</li> <li>3. System does not display menu page</li> <li>4. Customer cannot see loan applications option</li> <li>5. System fails to open loan application page</li> <li>6. Customer cannot enter information in loan application form</li> <li>7. Customer does not press send</li> <li>8. System does not add file to loan applications records</li> <li>9. Admin does not arrive at system</li> <li>10. Admin does not access customer's loan application</li> <li>11. Admin does not check customer's financial background</li> <li>12.                         <ol style="list-style-type: none"> <li>a) Admin reject loan application</li> <li>b) Admin enters wrong information for loan</li> <li>c) Admin fails to press save</li> <li>d) Admin loses internet connection</li> <li>e) Admin accidentally deletes customer's loan application</li> </ol> </li> <li>13.                         <ol style="list-style-type: none"> <li>a) Admin reject loan application</li> <li>b) Admin enters wrong information for loan</li> <li>c) Admin fails to press save</li> <li>d) Admin loses internet connection</li> <li>e) Admin accidentally deletes customer's loan application</li> </ol> </li> <li>14. Admin does not enter further information for loan</li> <li>15. Admin does not logout of system</li> <li>16. System does not update records</li> <li>17. System does not add loan amount to customer's balance and/or deduct amount from bank's own balance</li> <li>18. Customer does not receive message of approval and/or with the balance updated</li> <li>19. Customer does not log out of system</li> </ol>	

<b>Use Case Name:</b> Manage Transaction History
<b>Scope:</b> Bank Management System
<b>Level:</b> Usergroup
<b>Primary Actor:</b> Admin
<p><b>Stakeholders &amp; Interests:</b></p> <ul style="list-style-type: none"> <li>- Customer having interest in viewing their transaction history and pointing out errors</li> </ul>

- Admin having interest in managing transaction history by fixing updates and generating bank statements for customers
- Bank having interest to see if customer is operation transactions legally to protect the bank's image

**Precondition:**

- System is active
- Customer has an account
- Customer has transaction history
- Customer/admin/bank want to access transaction history

**Postcondition:**

- Successfully viewed transaction history
- System records updated in case of changes
- System is inactive

**Main Success Scenario:**

User	System
1. User arrives at system	
2. User logs into system	
	3. System displays menu
4. User chooses option to view transaction history of an account	
	5. System open transaction history page
6. User enters account number for which to show history	
	7. System use transaction history relevant to the account number
8. User views (and manages if bank/admin) transaction history to check if everything is correct	
9. User logs out of system	
	10. System records updated in case any changes were made
	11. System is inactive when not in use

**Extensions:**

1. User fails to access the system
  2. User fails to login to system
  3. System fails to display menu or load the page
  4. User cannot see option to access transaction history
  5. System fails to open
- page 6.
- a) User enters wrong account number
  - b) User does not enter account number
  - c) User enters wrong data
7. System fails to display transaction history of the relevant account

8. User cannot view (manage too if admin/bank) the transaction history correctly
9. User does not log out of system
10. System fails to update records in case of changes

<b>Use Case Name:</b> Manage System Records	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Admin	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>- Admin has interest in fixing and maintaining system records to avoid errors and malfunctions</li> <li>- Customer has interest to has the errors in their bank accounts fixed</li> <li>- Bank has interest in making sure their system is running smoothly</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- System is active</li> <li>- Admin needs/wants to manage records</li> <li>- Admin can access system</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- System records update</li> <li>- Changes to records made successfully</li> </ul>	
<b>Main Success Scenario:</b>	
Admin	System
1. Admin arrives at system	
2. Admin logs into system	
	3. System displays menu
4. Admin chooses option to manage system records	
	5. System open page of system records
	6. System asks admin which records he wants to manage
7. Admin enter name of records he wants to manage	
	8. System display table of relevant records
9. Admin views and manages records	
	10. System updates records if any changes are made
11. Admin logs out of system	
	12. System is inactive when not in use

**Extensions:**

1. Admin fails to access at system
2. Admins fails to log into system
3. System fails to display menu or load main page
4. Admin cannot see option to manage system records

5. System does not open page of system records
6. System does not ask admin for the records he wants to manage
7. Admin does not enter option for the records he wants to manage
8. System does not output relevant records
9. Admin cannot view and/or manage records
10. System does not update records after changes are make
11. Admin does not log out of system

<b>Use Case Name:</b> Generate Reports	
<b>Scope:</b> Bank Management System	
<b>Level:</b> Usergroup	
<b>Primary Actor:</b> Admin	
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>- Admin having interest to generate reports</li> <li>- Manager having interest in receiving reports of the bank's profits, financial status, etc.</li> <li>- Customer having interest in receiving their balance statement report</li> </ul>	
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- System is active</li> <li>- Admins wants/needs to generate reports</li> <li>- Admin can access the system</li> <li>- There is enough data to generate reports</li> </ul>	
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- Report/s generated successfully</li> </ul>	
<b>Main Success Scenario:</b>	
Admin	System
1. Admin arrives at the system	
2. Admin logs into the system	
	3. System displays main menu
4. Admin chooses option to generate reports	
	5. System opens generate reports page
	6. System asks admin for which report he wants to generate and relevant information needed to find data for the report
7. Admin chooses report type and the necessary data for query	
	8. System receives information from admin and generates report after querying relevant data
	9. System displays report to admin with option to print/save



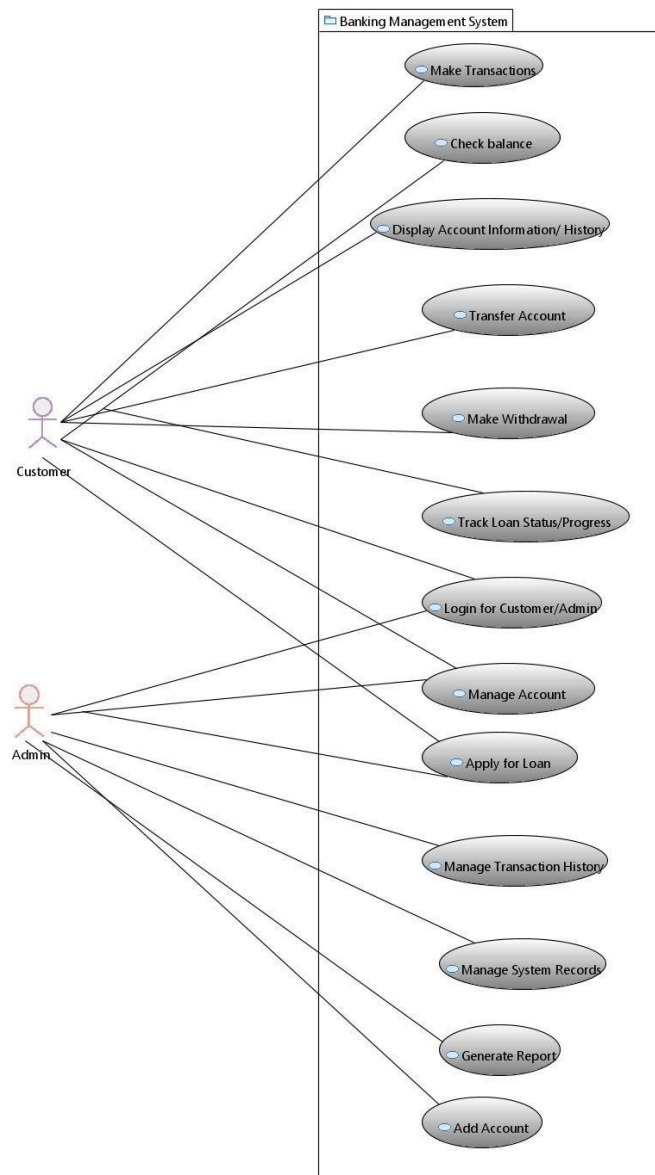
10. Admin views and (saves/prints) report	
	11. System downloads report if admin chooses download or prints report of admin chooses print
12. Admin logs out of system	
	13. System is inactive when not in use
<b>Extensions:</b> <ol style="list-style-type: none"> <li>Admin fails to access system</li> <li>Admin fails to log into system</li> <li>System fails to load main page or display menu</li> <li>Admin cannot see/access option to generate reports</li> <li>System fails to load page of generate reports</li> <li>System does not ask admin for report type and relevant information</li> <li>Admin enters wrong type and information for the report</li> <li>System fails to generate report <ol style="list-style-type: none"> <li>Due to poor internet connect</li> <li>Due to invalid query conditions</li> <li>Due to data and type mismatch</li> </ol> </li> <li> <ol style="list-style-type: none"> <li>System fails to display report in screen</li> <li>System fails to provide option to print/save</li> </ol> </li> <li>Admin does not choose to print/save</li> <li>System fails to print/download report</li> <li>Admin does not log out of system</li> </ol>	

<b>Use Case Name:</b> Add Account
<b>Scope:</b> Bank Management System
<b>Level:</b> Usergroup
<b>Primary Actor:</b> Admin
<b>Stakeholders &amp; Interests:</b> <ul style="list-style-type: none"> <li>- Admin having interest in gaining customer</li> <li>- Customer having interest in holding an account in the bank</li> <li>- Customer's occupation company having interest in which bank and bank account to send the customer's salary</li> <li>- Bank having interest in gaining number of accounts being used</li> </ul>
<b>Precondition:</b> <ul style="list-style-type: none"> <li>- System is active</li> <li>- Customer wants to open an account</li> <li>- Admin can access the system</li> <li>- Admin is present</li> </ul>
<b>Postcondition:</b> <ul style="list-style-type: none"> <li>- Account created successfully</li> <li>- User has his account login information</li> <li>- System is updated</li> </ul>



Main Success Scenario:	
User	System
1. Customer arrives at registration counter	
2. Admin accesses the system	
3. Admin logs into system	
	4. System displays main menu
5. Admin chooses registration option	
	6. System opens registration page
7. Admin asks customer for details	
8. Customer gives his/her details to admin	
9. Admin enters details into system till all required entries are filled	
10. Admin presses 'create account' option	
	11. System creates new records of new user with the data
12. Admin gives customer details of his login and password	
13. Customer acquired login and password to access system and leaves	
14. Admin logs out of system	
	15. System in inactive when not in use
<b>Extensions:</b> <ol style="list-style-type: none"> <li>Customer fails to arrive at registration point</li> <li>Admin fails to access the system</li> <li>Admin fails to login to system</li> <li>System fails to load main menu</li> <li>Admin cannot find the registration option</li> <li>System fails to load registration page</li> <li>Admin does not ask all necessary details from customer</li> <li>Customer does not give correct details</li> <li>Admin cannot enter details into system or fails to enter all details</li> <li>Admin does not press 'create account'</li> <li>System does not create new account with/without user data</li> <li>Admin does not give customer his/her username and password</li> <li>Customer does not receive login and password to access the system</li> <li>Admin does not log out of system</li> </ol>	

## Use Case Diagram



## Other Nonfunctional Requirements

### Performance Requirements

To ensure optimal performance and compatibility, the **Bank Management System (BMS)** is designed to operate with the following technical stack and configurations:

#### 1. Database Requirements:

- **MySQL Database:** Required for robust and secure data storage, handling large volumes of customer and transaction records efficiently.
- The database schema is provided below for setup. It includes all necessary tables for managing customers, administrators, accounts, transactions, loans, and loan applications.

#### CODE for MYSQL:

```
-- Drop existing tables if
necessary
drop table
BankLoanApplication;
drop table BankTransaction;
drop table BankLoan;
drop table BankAccount;
drop table BankAdmin;
drop table BankCustomer;

-- Create Customer Table
create table BankCustomer (
    custID varchar2(10) not
null primary key,
    name varchar2(30) not
null,
    phone# varchar2(13),
    email varchar2(30),
    username varchar2(20)
not null unique,
    password varchar2(20) not
null
);

-- Insert sample data for
Customers
insert into BankCustomer
values ('1', 'Rafay Khattak',
'+923333333333',
'rafay@gmail.com',
'rafaycust', 'rafay');
insert into BankCustomer
values ('2', 'Waseem Akhtar',
'+923333333334',
```

```
'waseem@gmail.com',
'waseemcust', 'waseem');
insert into BankCustomer
values ('3', 'Hashir',
'+923333333335',
'hashir@gmail.com',
'hashircust', 'hashir');

-- Create Admin Table
create table BankAdmin (
  adminID varchar2(10) not
null primary key,
  name varchar2(30) not
null,
  phone# varchar2(13),
  email varchar2(30),
  username varchar2(20)
not null unique,
  password varchar2(20) not
null
);

-- Insert sample data for
Admins
insert into BankAdmin values
('1', 'Rafay Khattak',
'+923333333336',
'rafay1312@gmail.com',
'rafayadm', 'rafay');

-- Create Account Table
create table BankAccount (
  acc# varchar2(12) not null
primary key,
  balance number not null,
  creationDate date not null,
  custID varchar2(10) not
null,
  foreign key (custID)
references
BankCustomer(custID),
  status varchar2(10) not
null
);

-- Insert sample data for
Accounts
insert into BankAccount
values ('1', 300000,
TO_DATE(sysdate,
'yyyy/mm/dd hh24:mi:ss'),
'1', 'Active');
insert into BankAccount
values ('2', 300000,
TO_DATE(sysdate,
'yyyy/mm/dd hh24:mi:ss'),
'2', 'Active');
insert into BankAccount
values ('3', 300000,
```

```
TO_DATE(sysdate,
'yyyy/mm/dd hh24:mi:ss'),
'3', 'Active');

-- Create Transaction Table
create table BankTransaction
(
    transactionID varchar2(15)
not null primary key,
    acc# varchar2(12) not null,
    foreign key (acc#)
references
BankAccount(acc#),
    transDate date not null,
    amount number not null,
    dueDate date,
    reason varchar2(50) not
null
);

-- Create Loan Table
create table BankLoan (
    loanID varchar2(15) not
null primary key,
    acc# varchar2(12) not null,
    foreign key (acc#)
references
BankAccount(acc#),
    amount number not null,
    interestRate number not
null,
    acceptDate date not null,
    endDate date not null,
    status varchar2(10) not
null
);

-- Create Loan Application
Table
create table
BankLoanApplication (
    loanID varchar2(15) not
null primary key,
    acc# varchar2(12) not null,
    foreign key (acc#)
references
BankAccount(acc#),
    amount number not null,
    applyDate date not null,
    status varchar2(10) not
null
);
```

## Safety Requirements

The Bank Management System (BMS) ensures the highest level of data integrity and safety by leveraging secure storage mechanisms. All critical data is stored in a centralized database (MySQL), which is backed up regularly to prevent any potential loss due to system failures or unforeseen incidents. The system implements robust file-handling processes to minimize risks associated with corruption or accidental deletion.

Additionally, the platform incorporates safeguards such as transaction logging and audit trails to track changes, ensuring data recovery in case of unexpected disruptions. The architecture adheres to best practices for data redundancy and error checking, making it highly resistant to data loss or corruption. This approach guarantees that sensitive banking information remains secure, recoverable, and protected from potential hazards.

## Security Requirements

The Bank Management System (BMS) prioritizes data security by utilizing MySQL, a trusted and highly secure database management system. Access to the database is strictly controlled through a robust authentication mechanism, ensuring that only authorized users—admins or customers—can interact with the system. User credentials are securely stored, and passwords are encrypted to prevent unauthorized access.

It is crucial to ensure that the installed SQL system is a licensed and official version to avoid vulnerabilities associated with cracked or unverified installations. Unauthorized versions may contain exploitable code segments, posing significant security risks. Additionally, the system is designed to safeguard against SQL injection attacks and employs parameterized queries to enhance database security. Regular security patches and updates should be applied to the software to maintain its integrity and protect against emerging threats.

By following these measures, the system guarantees a secure environment for handling sensitive banking data, mitigating risks of breaches or unauthorized intrusions.

## Software Quality Attributes

The Bank Management System (BMS) is designed with key software quality attributes to ensure reliability, adaptability, and maintainability.

- **Reusability:** The system's modular design, with functionalities encapsulated in separate classes and controllers, allows components to be reused across different modules. This reduces redundancy and simplifies the development of future features.
- **Ease of Debugging:** Each module is developed independently, making it easier to

identify and resolve issues. Clear error-handling mechanisms and detailed logs are integrated to assist in debugging and ensure system stability.

- **Validation Checks:** Robust validation mechanisms are implemented at every critical point, such as user input, transaction processing, and loan applications, ensuring the system remains accurate and secure.
- **Updatability:** The system's architecture supports seamless updates and enhancements. New features or improvements can be added without significant downtime, as components can be independently modified or upgraded.
- **Maintainability:** With a well-organized codebase and adherence to clean coding principles, the system is easy to maintain and extend. Comprehensive documentation further simplifies troubleshooting and future development.
- **Scalability:** Designed to handle increasing workloads, the system can support a growing number of users, accounts, and transactions without performance degradation.
- **Interoperability:** The use of Java and MySQL ensures compatibility with a wide range of platforms and third-party tools, making the system versatile and easily integrable into existing infrastructure.

## Business Rules

The Bank Management System (BMS) operates under a defined set of business rules to ensure smooth, secure, and efficient operation:

- **Administrative Privileges:**
  - Only the Admin has access to advanced functionalities, including viewing and managing all customer accounts and transactions.
  - The Admin can generate detailed reports, approve or reject loan applications, and oversee the overall system operations.
- **Customer Privileges:**
  - Customers are limited to accessing their own accounts and performing basic banking functions, such as viewing account balances, initiating transactions, and applying for loans.
  - Customers are restricted to one active loan application at a time to simplify loan management and ensure fair processing.
- **Data Access Control:**
  - Admin accounts and customer accounts are securely segregated, ensuring that no unauthorized access to sensitive data occurs.
- **Transaction Handling:**
  - All transactions must pass validation checks before being processed to avoid errors or inconsistencies.
  - Transaction limits and restrictions may apply based on account status or specific banking policies.
- **Loan Management:**
  - Loan approval is exclusively handled by the Admin, based on customer eligibility and predefined criteria.

- Loan statuses are updated in real time, and customers are notified of decisions promptly.
- **Compliance and Reporting:**
  - The system ensures compliance with banking regulations, and audit logs are maintained for all significant operations.



## Operating Environment

The Bank Management System (BMS) is designed to operate seamlessly across various platforms, provided the following conditions are met:

- **Software Requirements:**
  - The system requires the installation of Java SDK 20, JavaFX SDK 21, and MySQL (or higher) for database management.
  - All software components must be authentic and up-to-date to ensure compatibility and security.
- **Hardware Compatibility:**
  - BMS is optimized to run efficiently on standard hardware configurations, including personal computers, servers, and laptops.
  - Sufficient storage space and processing power are necessary to handle the database operations and user interactions without performance issues.
- **Operating System Support:**
  - The program is platform-independent and can operate on any operating system (e.g., Windows, macOS, Linux) capable of running the required Java and database environments.
- **Networking and Connectivity:**
  - For multi-user environments, a stable network connection is required to ensure real-time data synchronization between the database and the application.
- **Environment Maintenance:**
  - Regular updates to the underlying software and hardware environment are recommended to maintain system performance and security.
  - The system is tested and validated to perform under standard environmental conditions without requiring specialized configurations.

## User Interfaces

The user interface (UI) of the Bank Management System (BMS) is designed to be intuitive, user-friendly, and minimalist, ensuring an easy and efficient user experience. The key features of the UI are as follows:

- **Simple and Clean Design:**
  - The interface is designed to be minimalistic, with clearly labeled fields and buttons for easy navigation. It aims to reduce clutter and focus on essential functions, enhancing usability for both admins and customers.
- **User Guidance:**
  - Proper validation and prompt messages are integrated into the system to guide users in case of invalid entries or incorrect input formats. If a user enters data incorrectly, the system provides clear instructions or error messages, helping them to correct the issue and proceed smoothly.

- **Role-based Access Control:**

- The system restricts access to features based on user roles. Administrators have full access to account management, loan decisions, transaction records, and report generation. Customers, on the other hand, can only access their own data, perform basic transactions, and apply for loans. This ensures that users only have access to functionalities relevant to their roles.

- **Data Security and Integrity:**

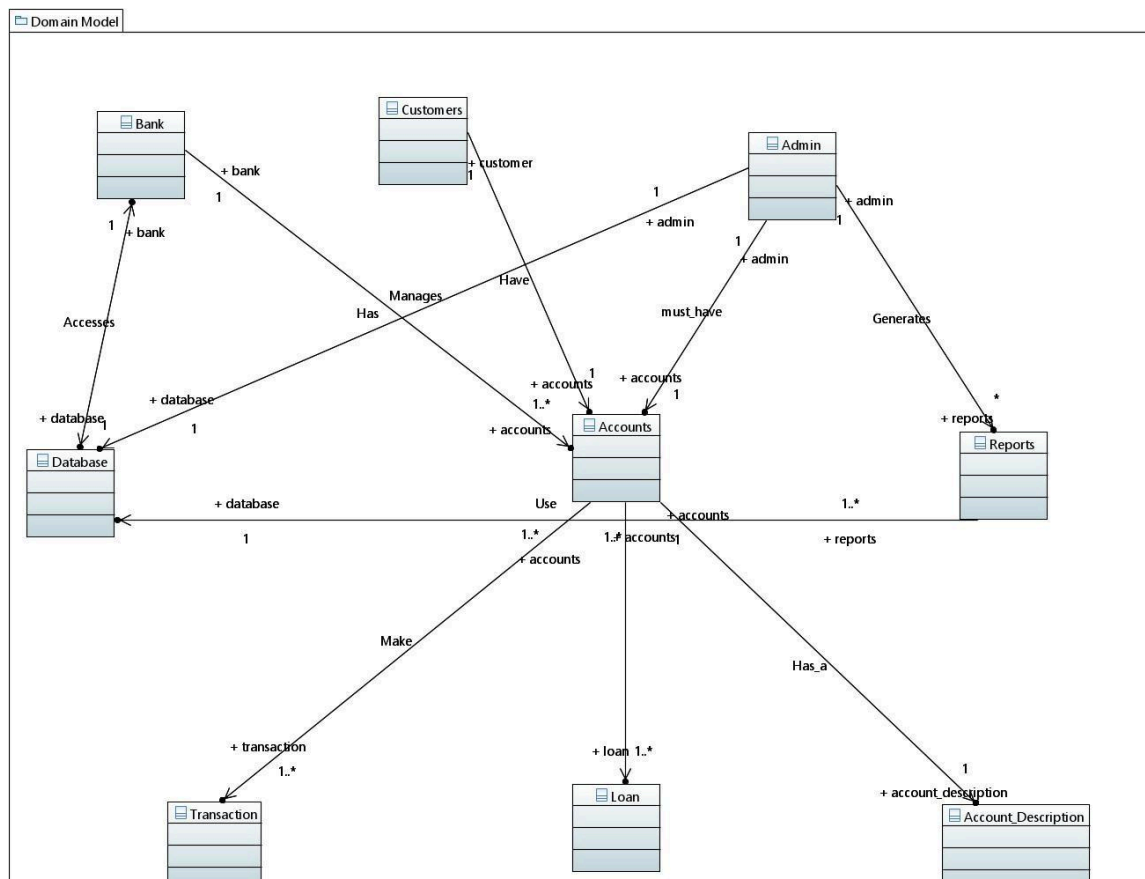
- User data is securely stored in an MySQL database, encrypted and protected to prevent unauthorized access or loss. Each user's data is kept private, with access granted only to the individual user or the admin, as per their permissions.
- The system ensures that no data is lost or corrupted, providing a reliable environment for users to view, edit, and manage their data without fear of accidental deletion.

- The system ensures that no data is lost or corrupted, providing a reliable environment for users to view, edit, and manage their data without fear of accidental deletion.

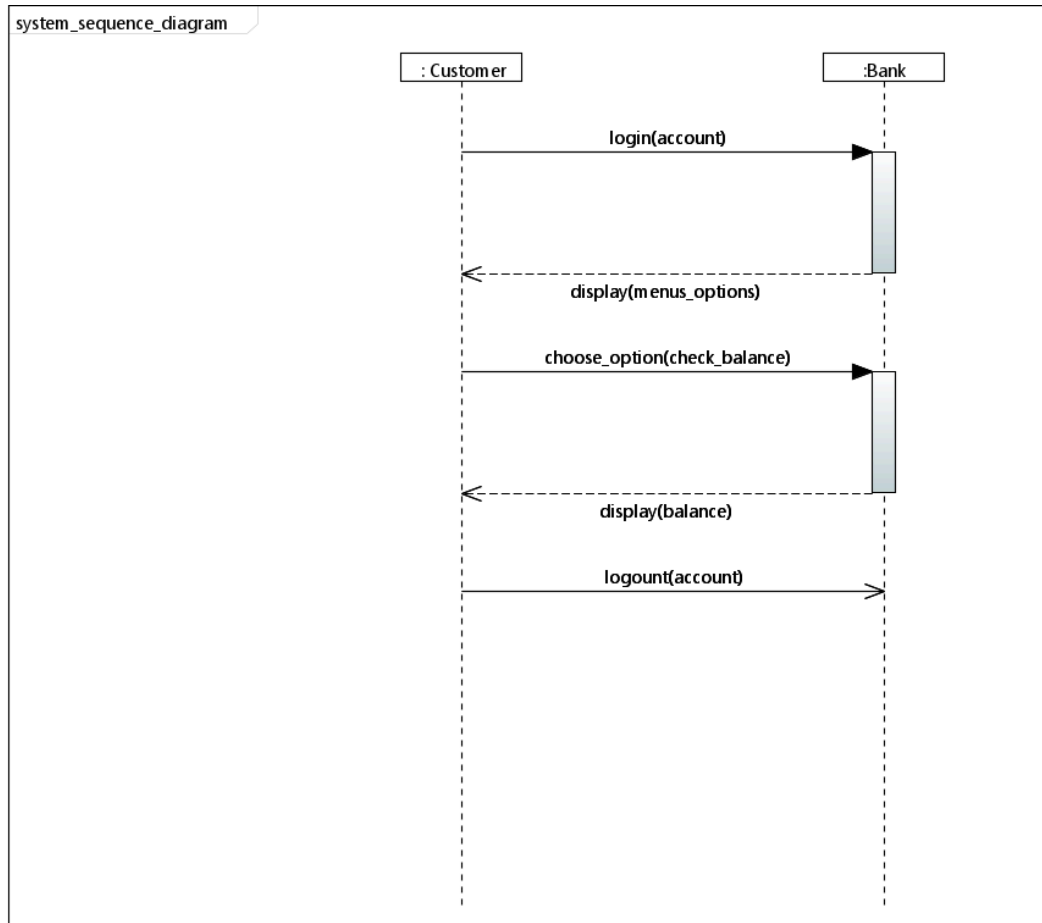
- **Responsive and Adaptive Design:**

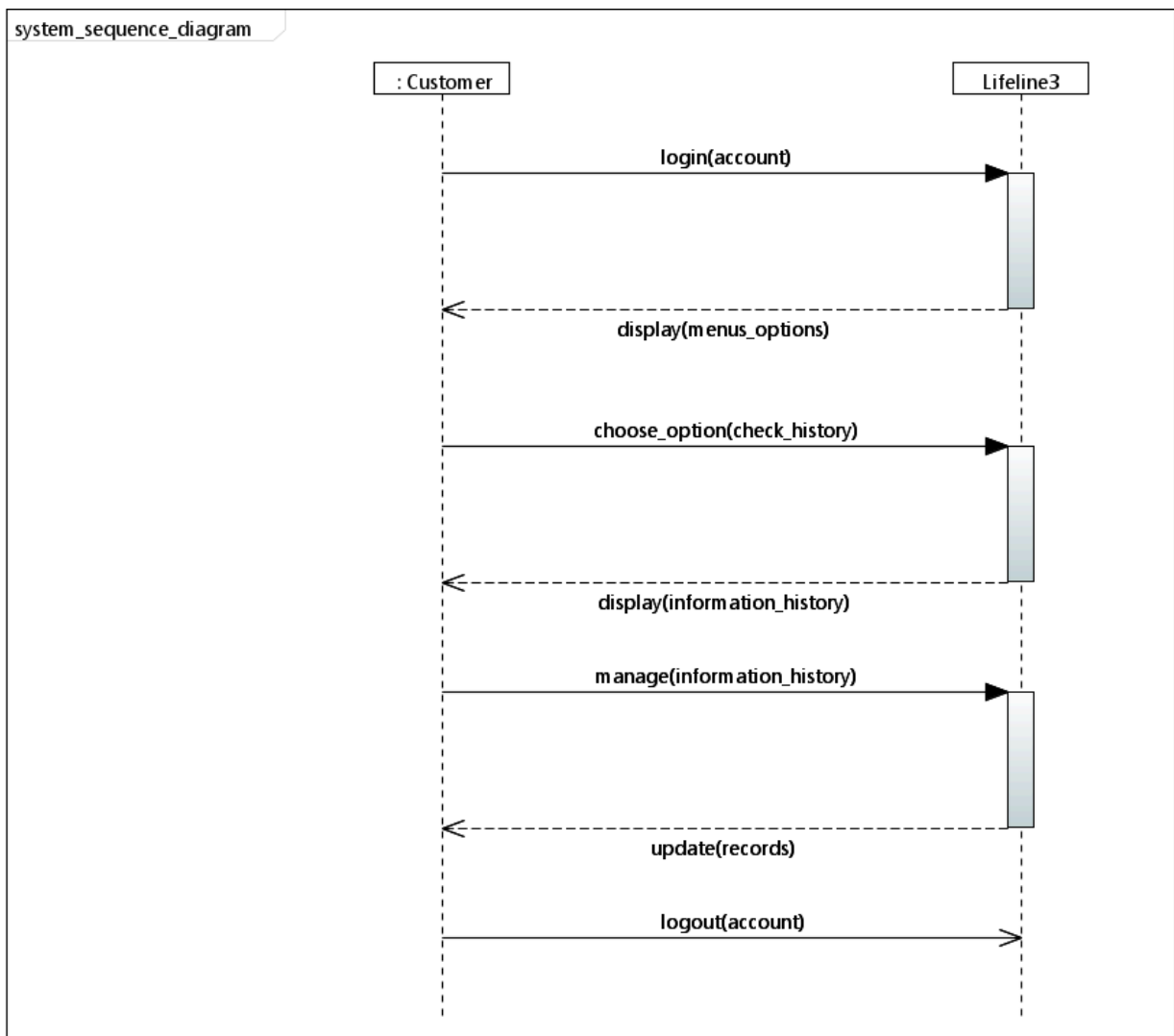
- The UI is designed to be responsive and adapt to various screen sizes, ensuring a consistent experience on desktops, tablets, and smartphones. This flexibility makes it convenient for users to interact with the system across multiple devices.

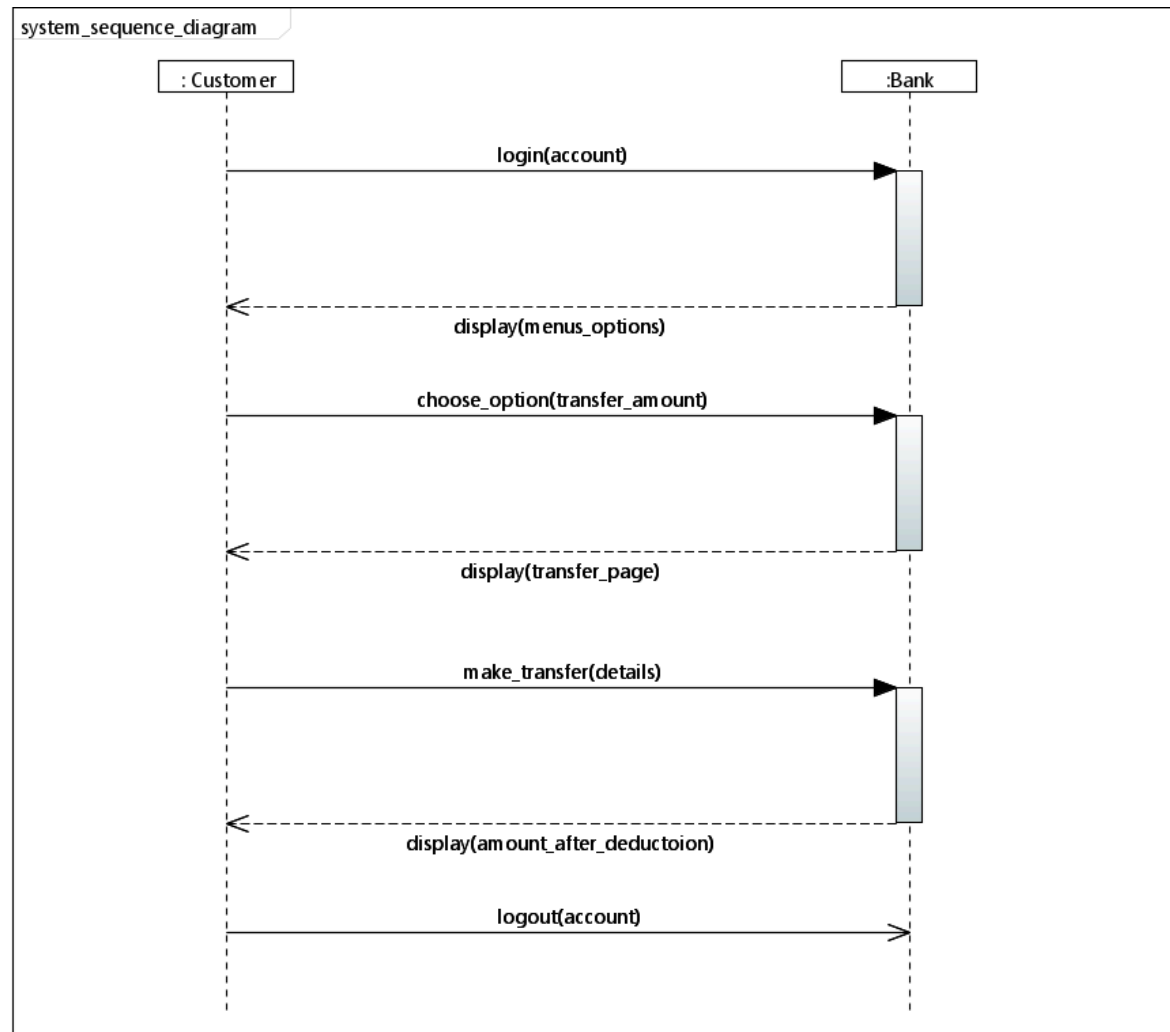
## 4 Domain Model

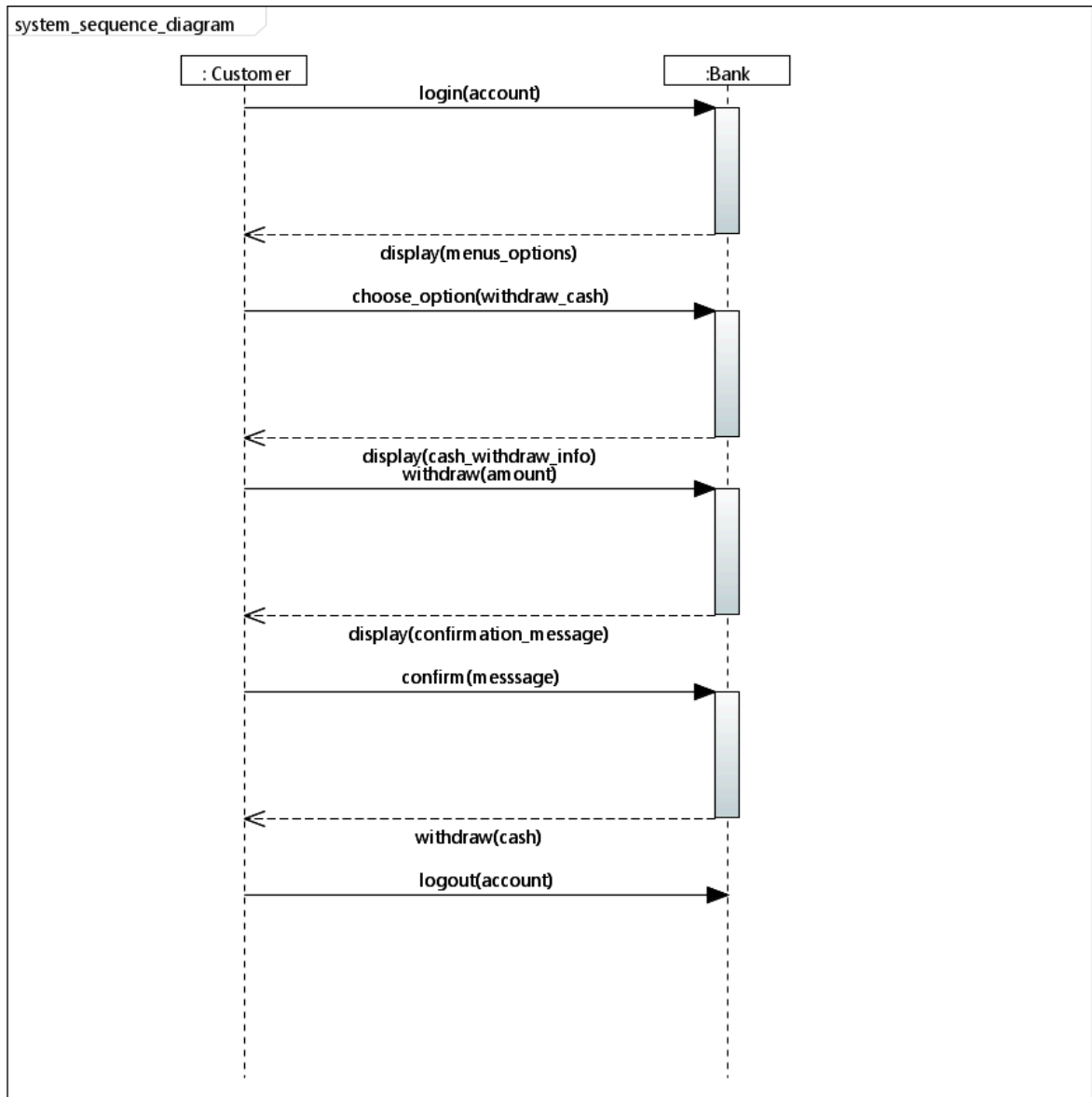


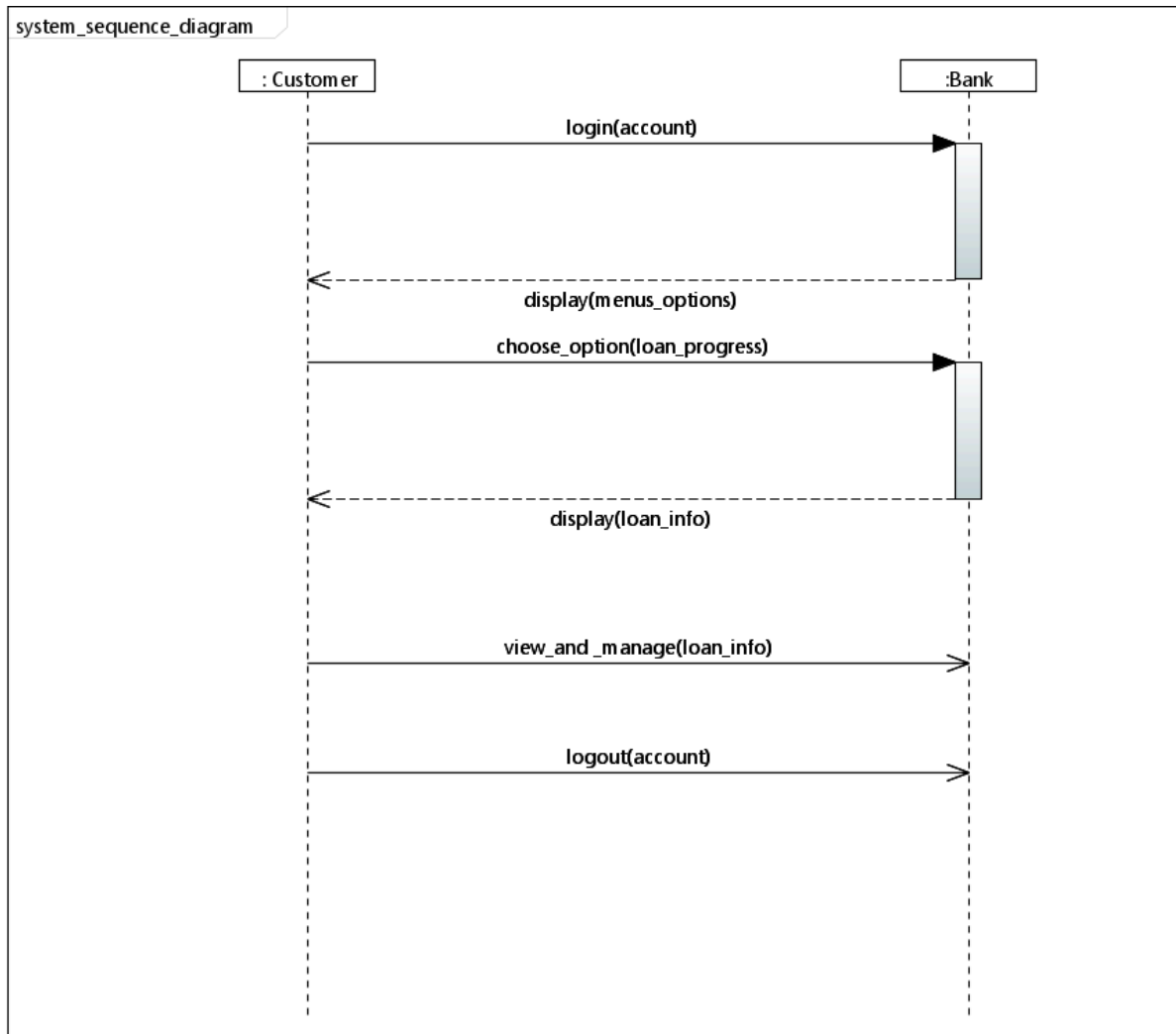
## 5 System Sequence Diagram

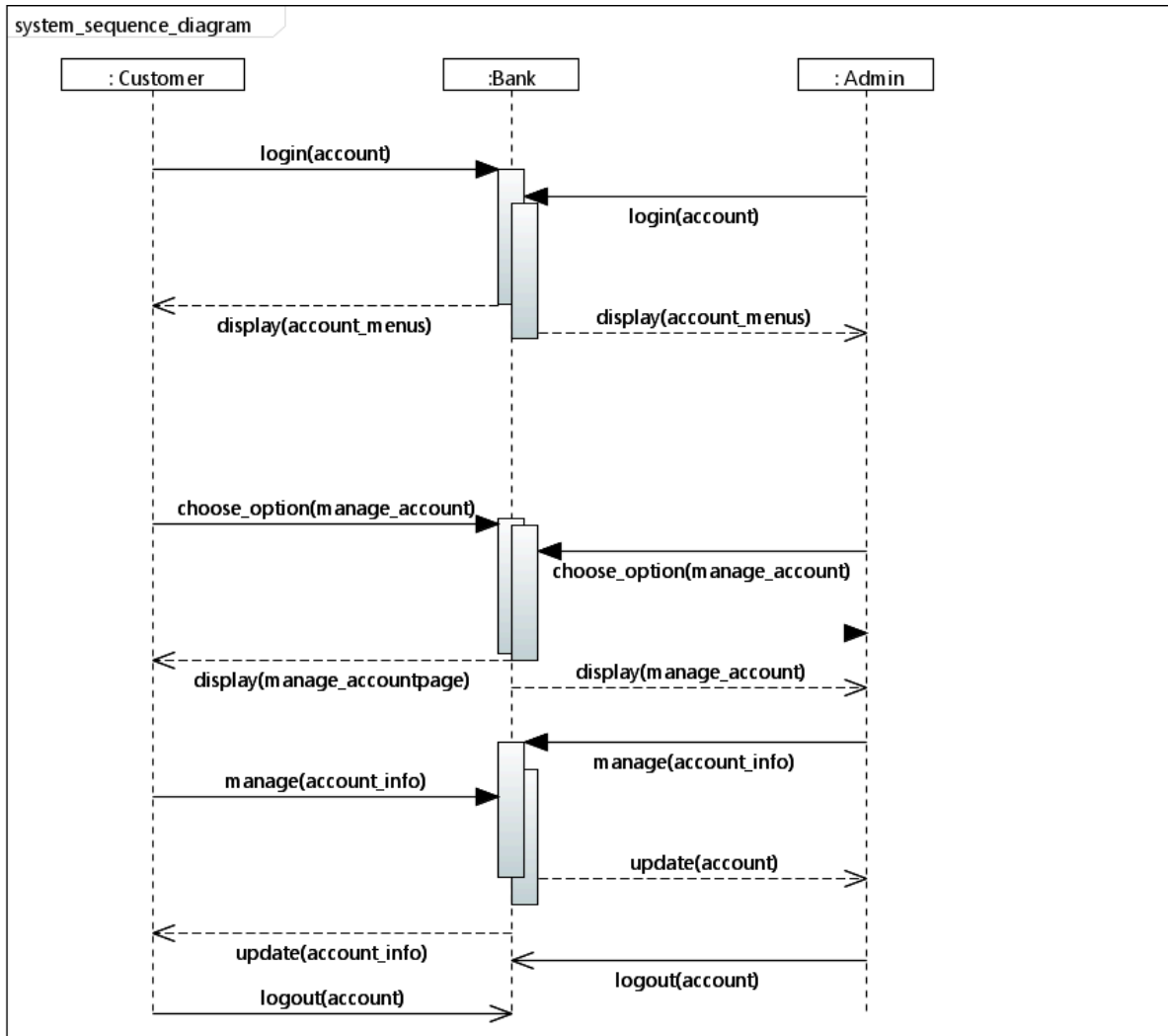




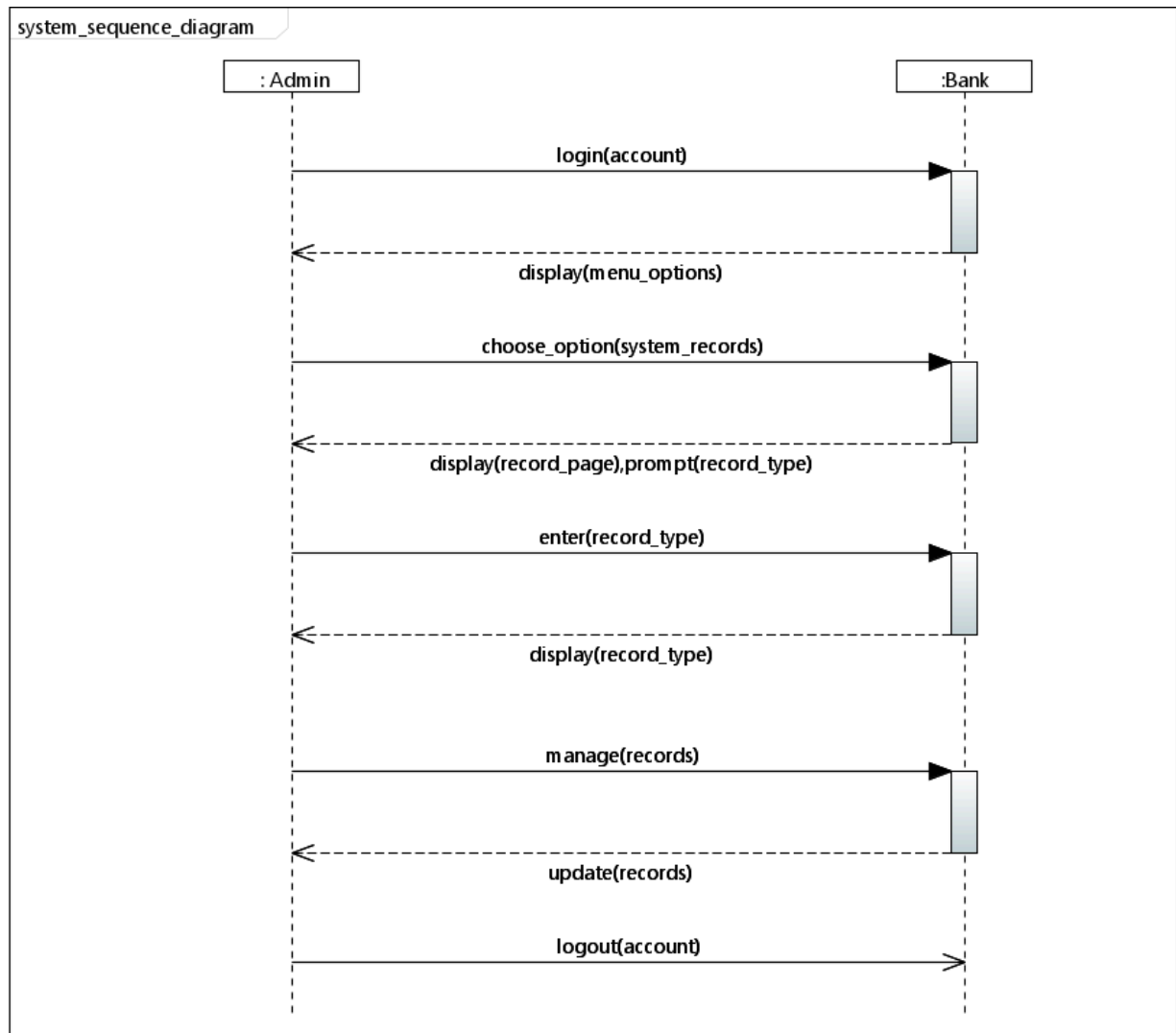


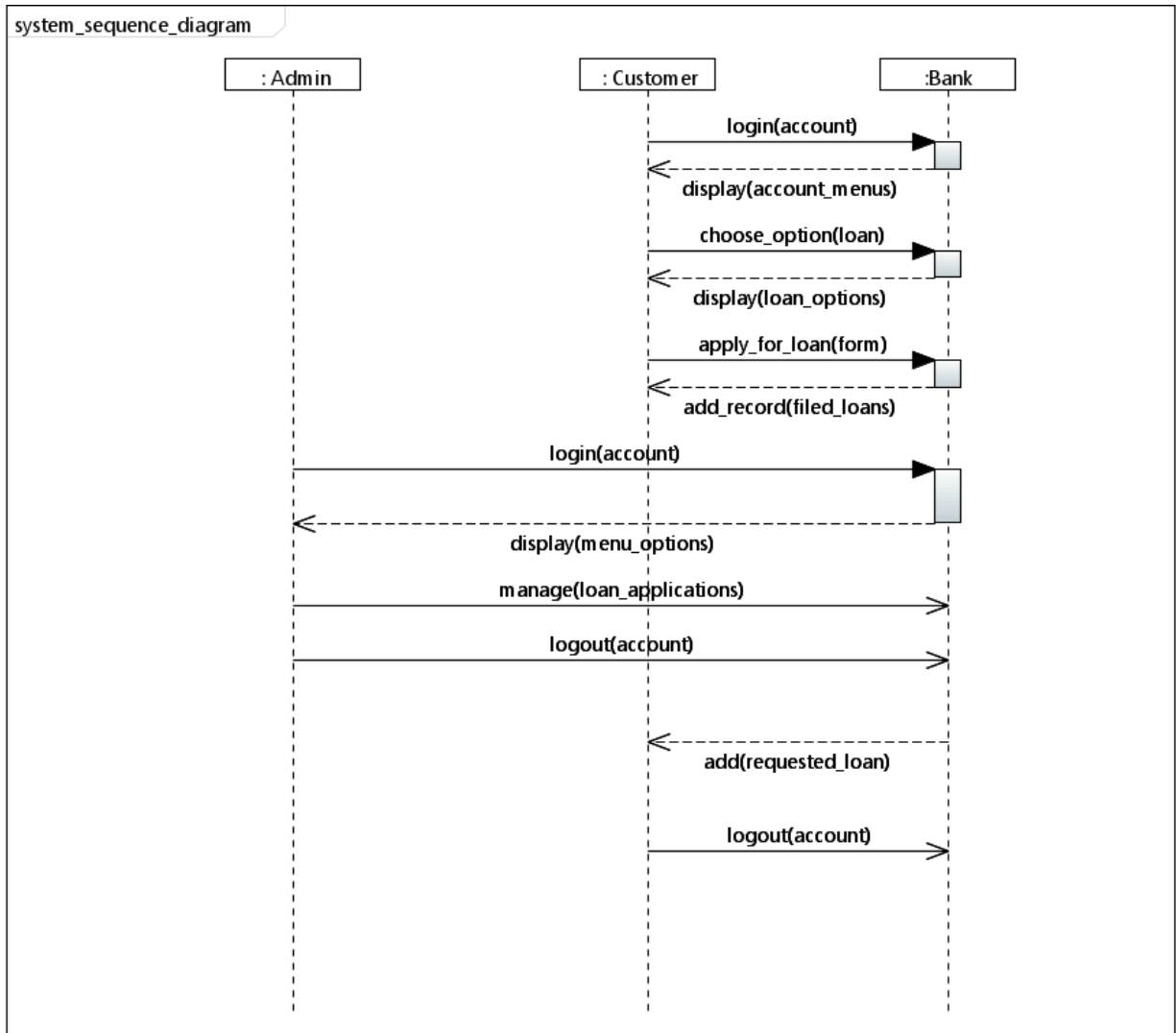


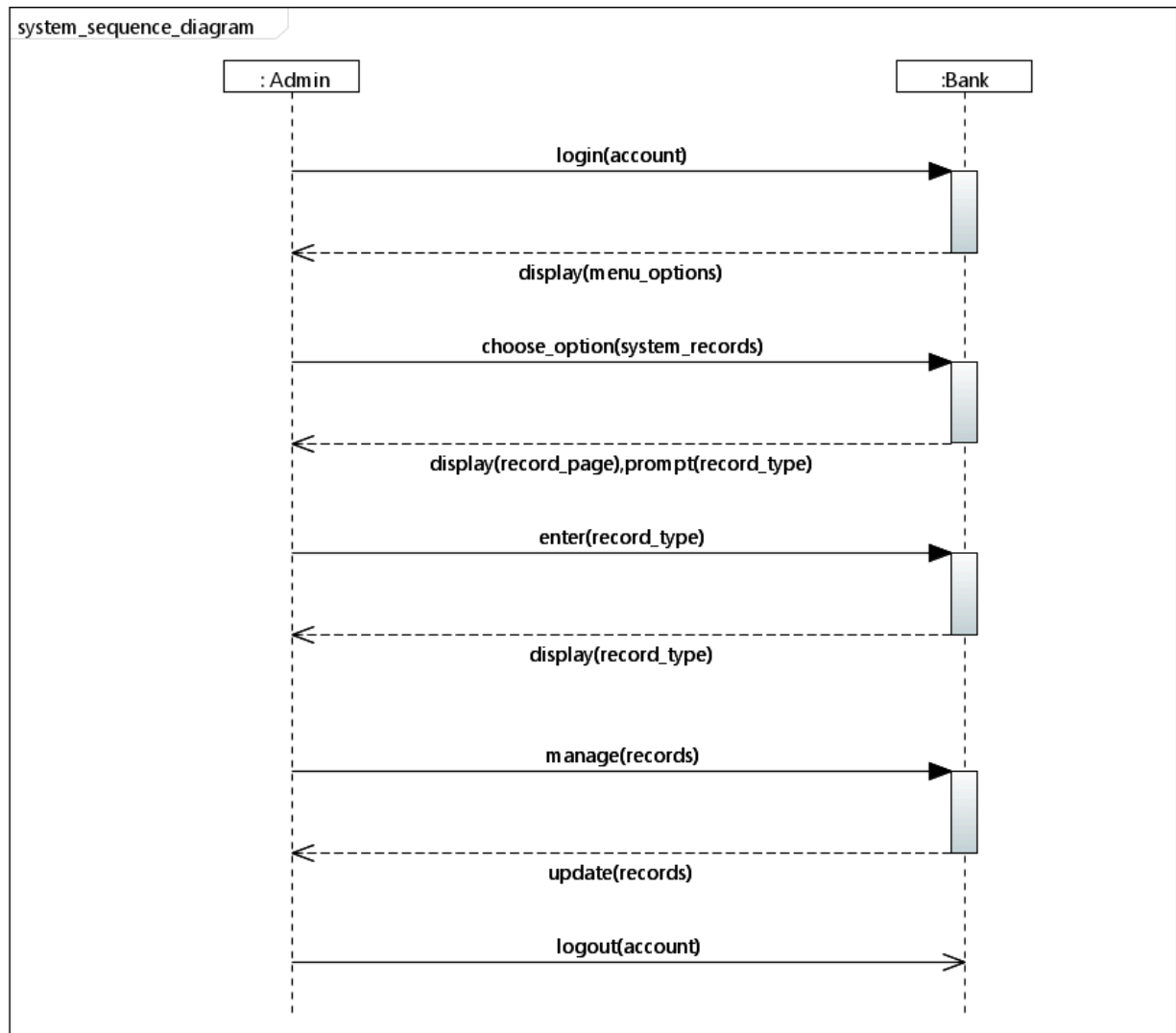


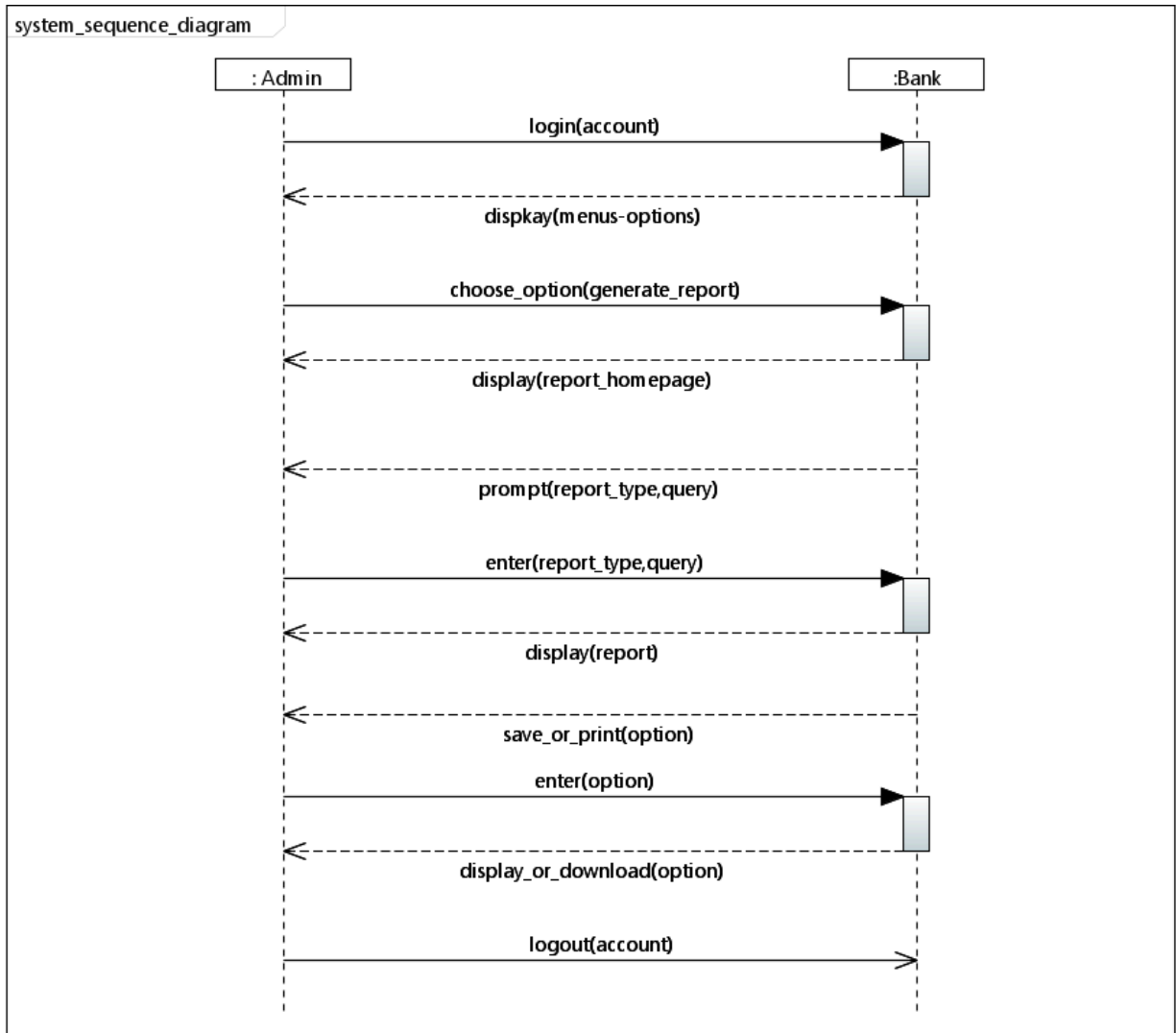


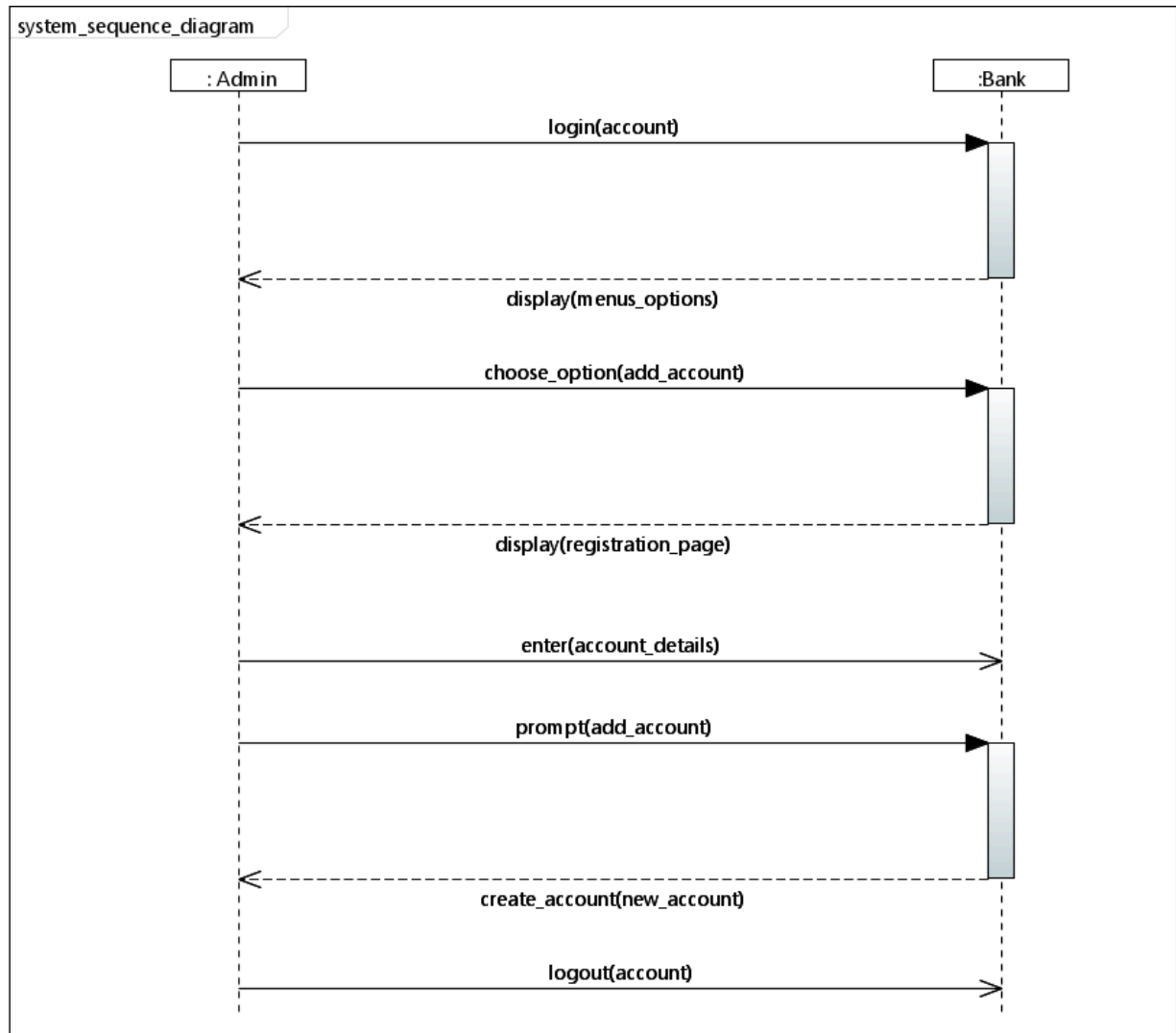




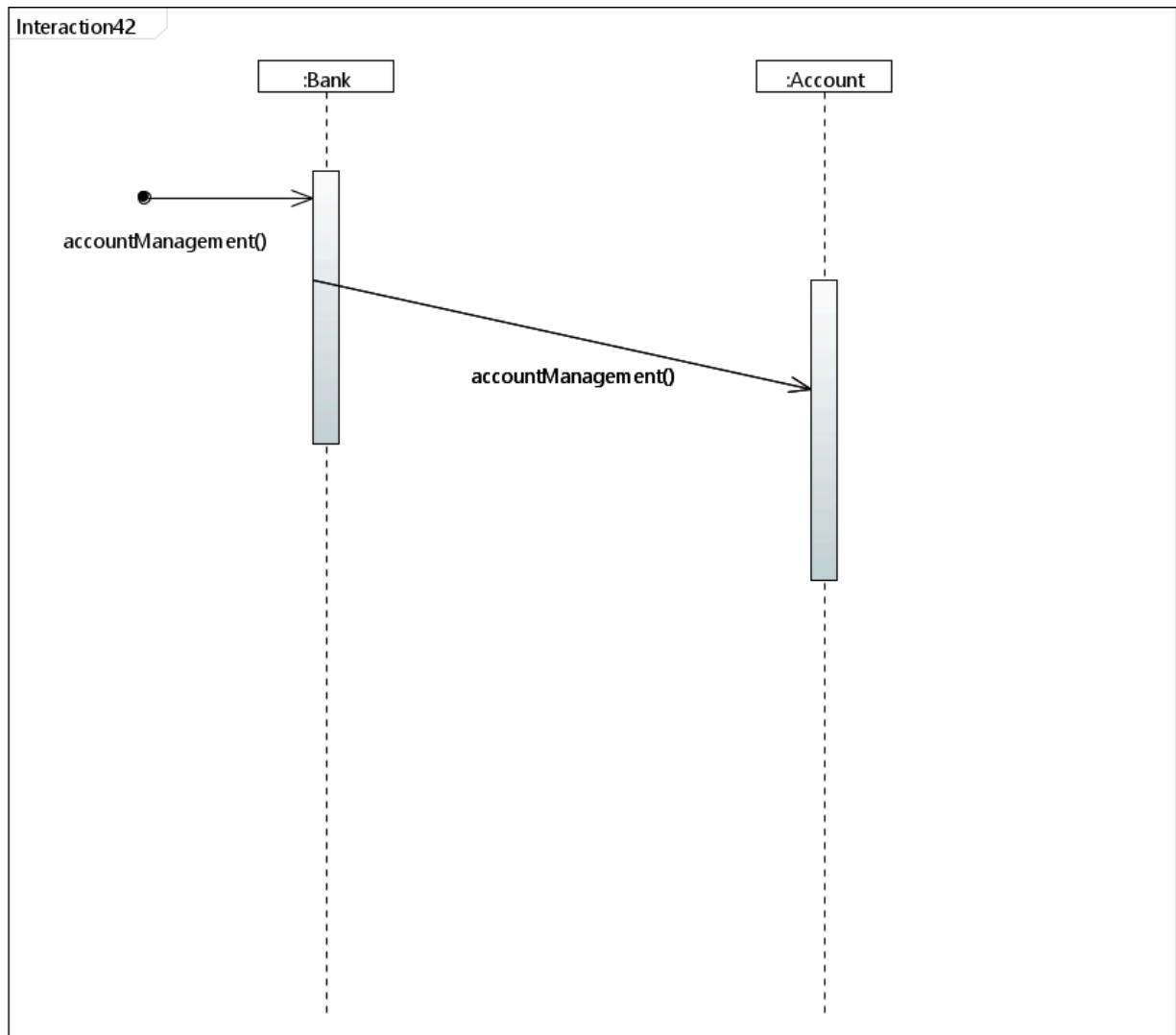


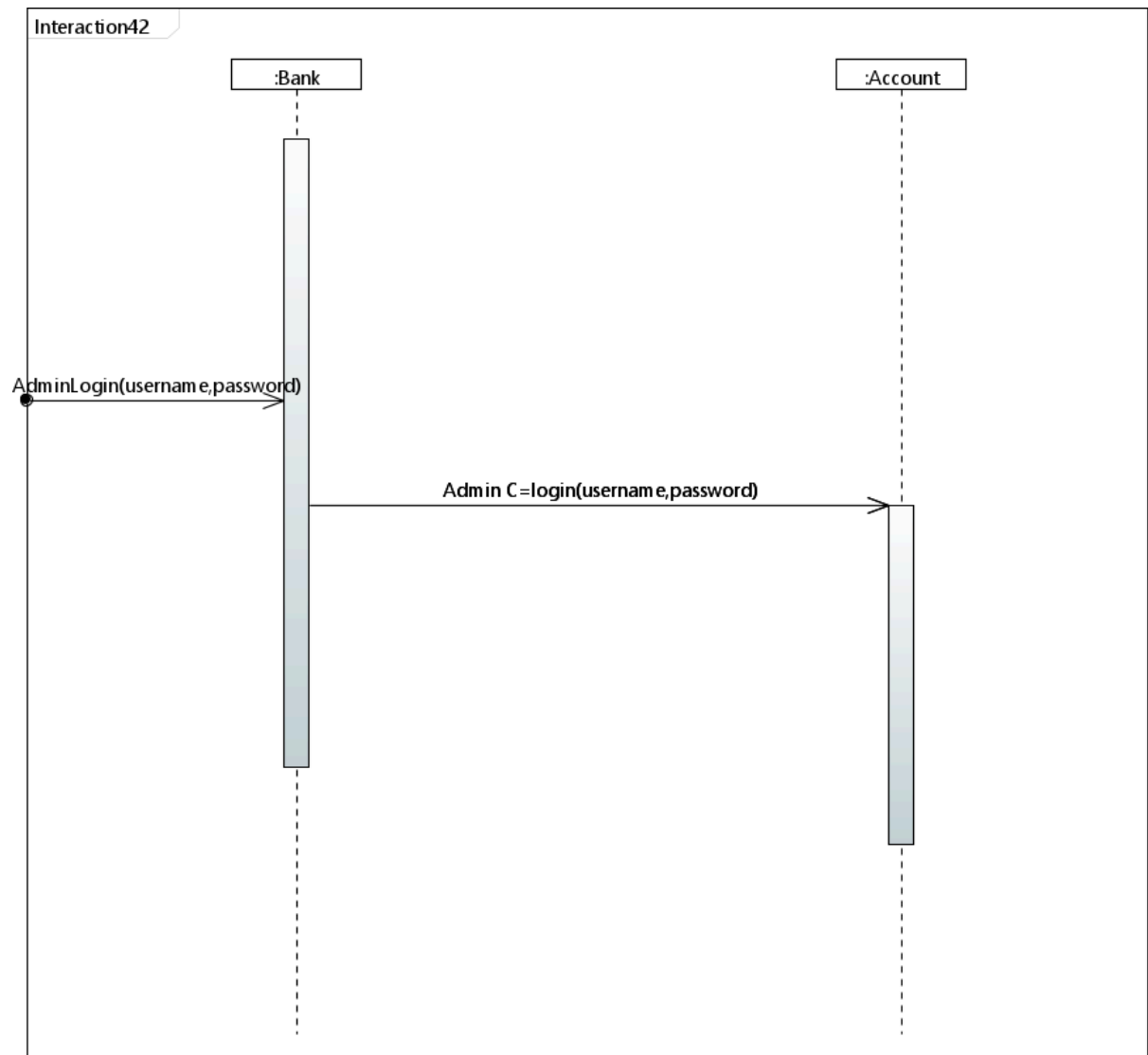


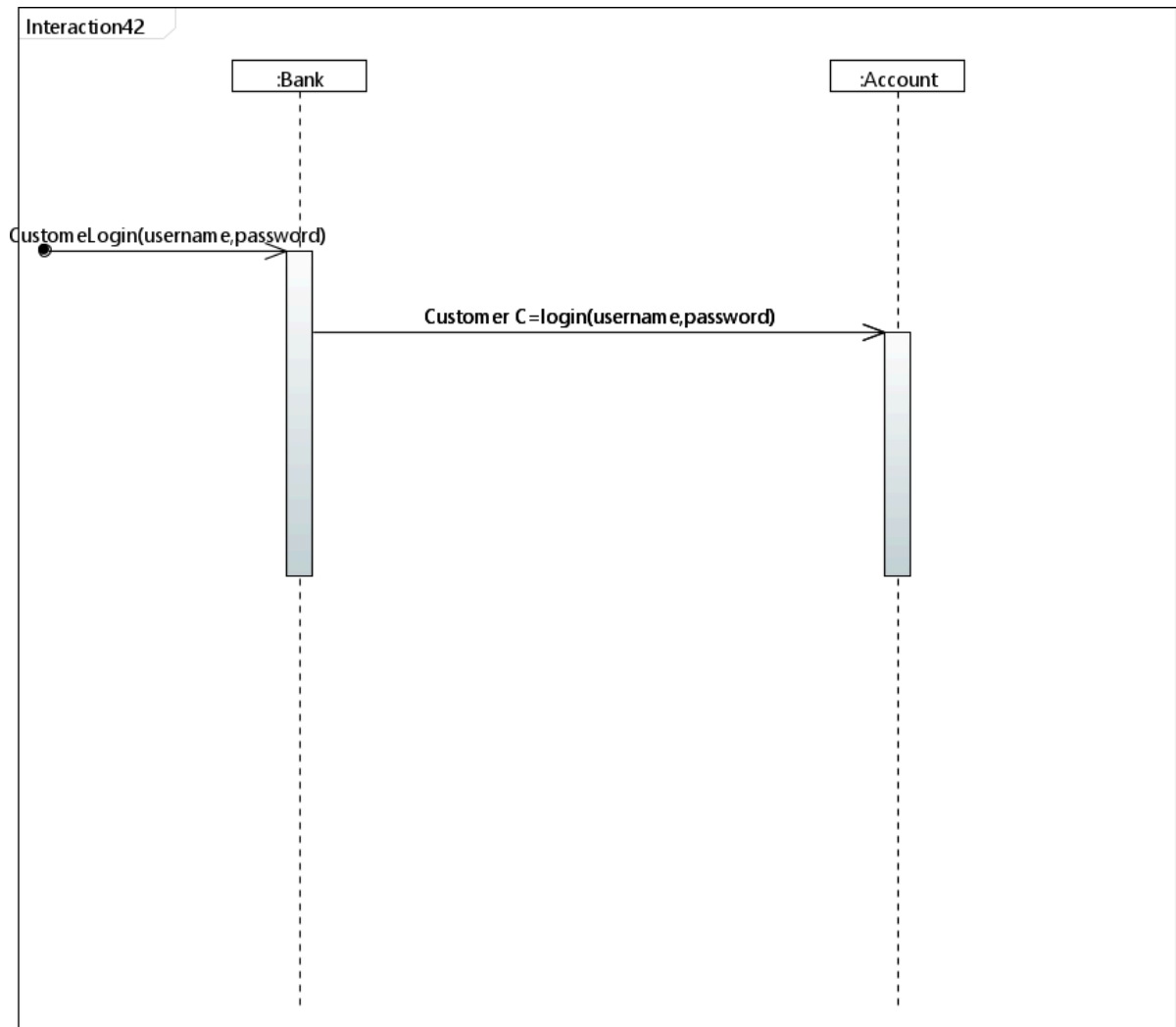




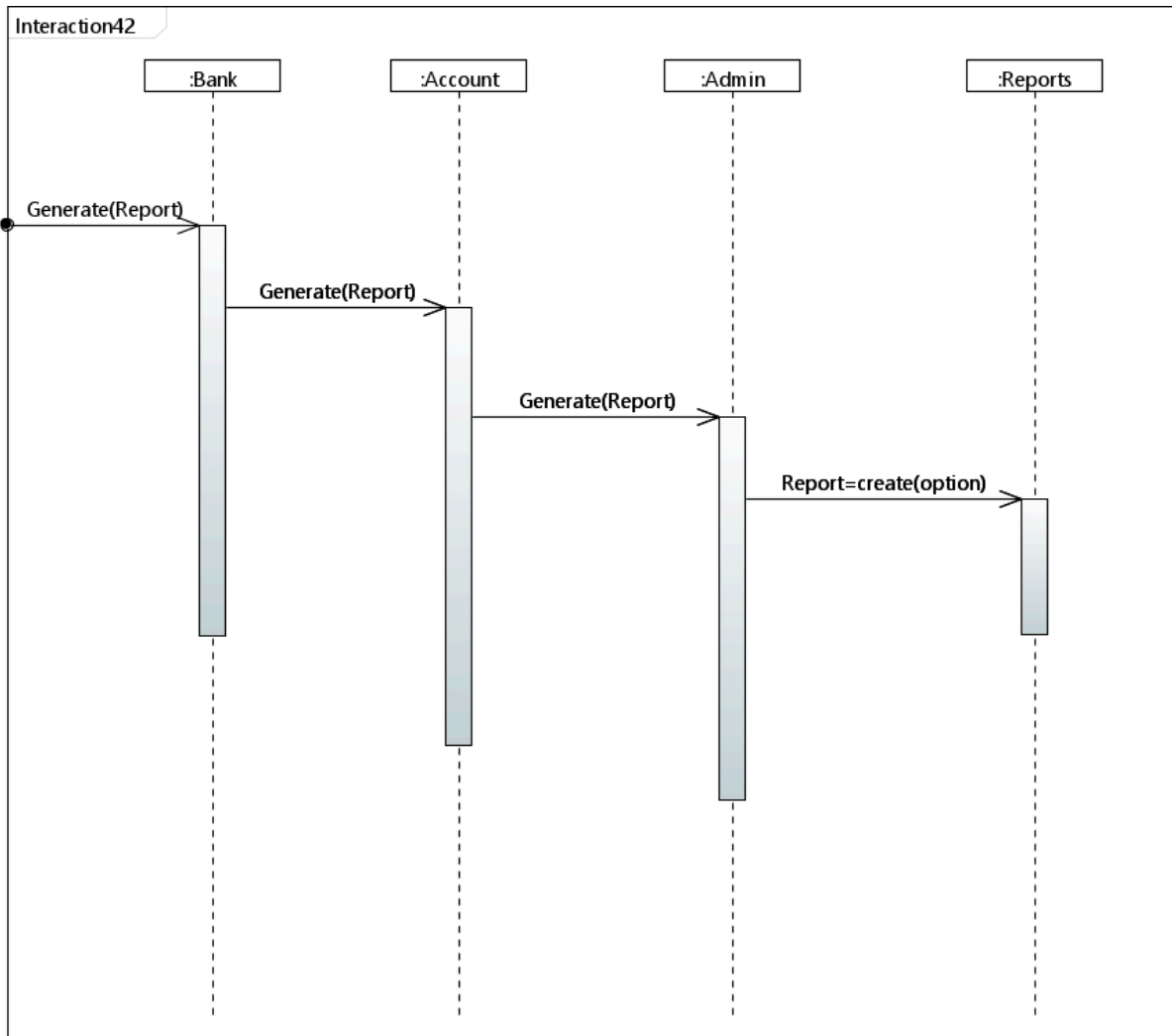
## 6 Sequence Diagram

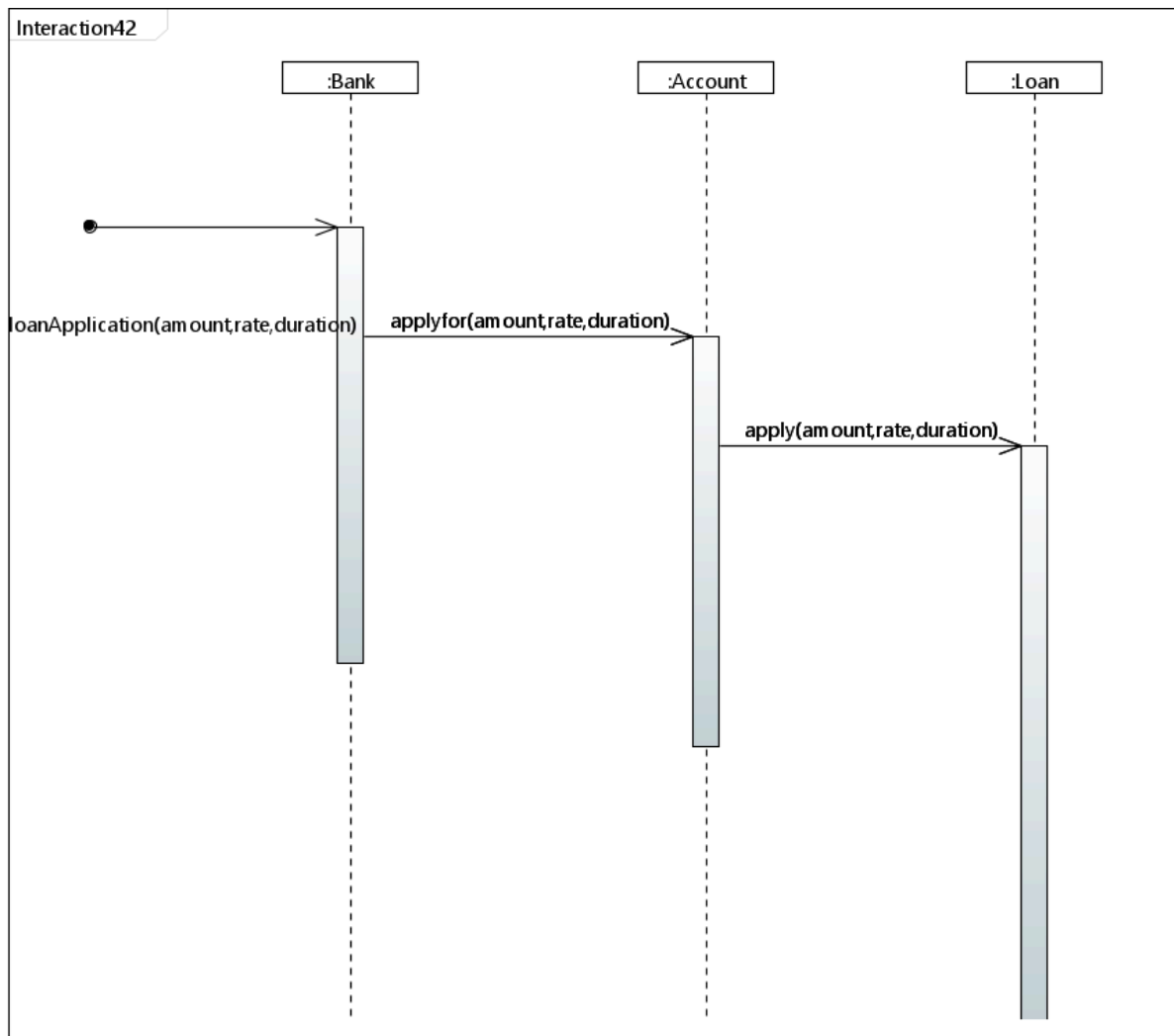




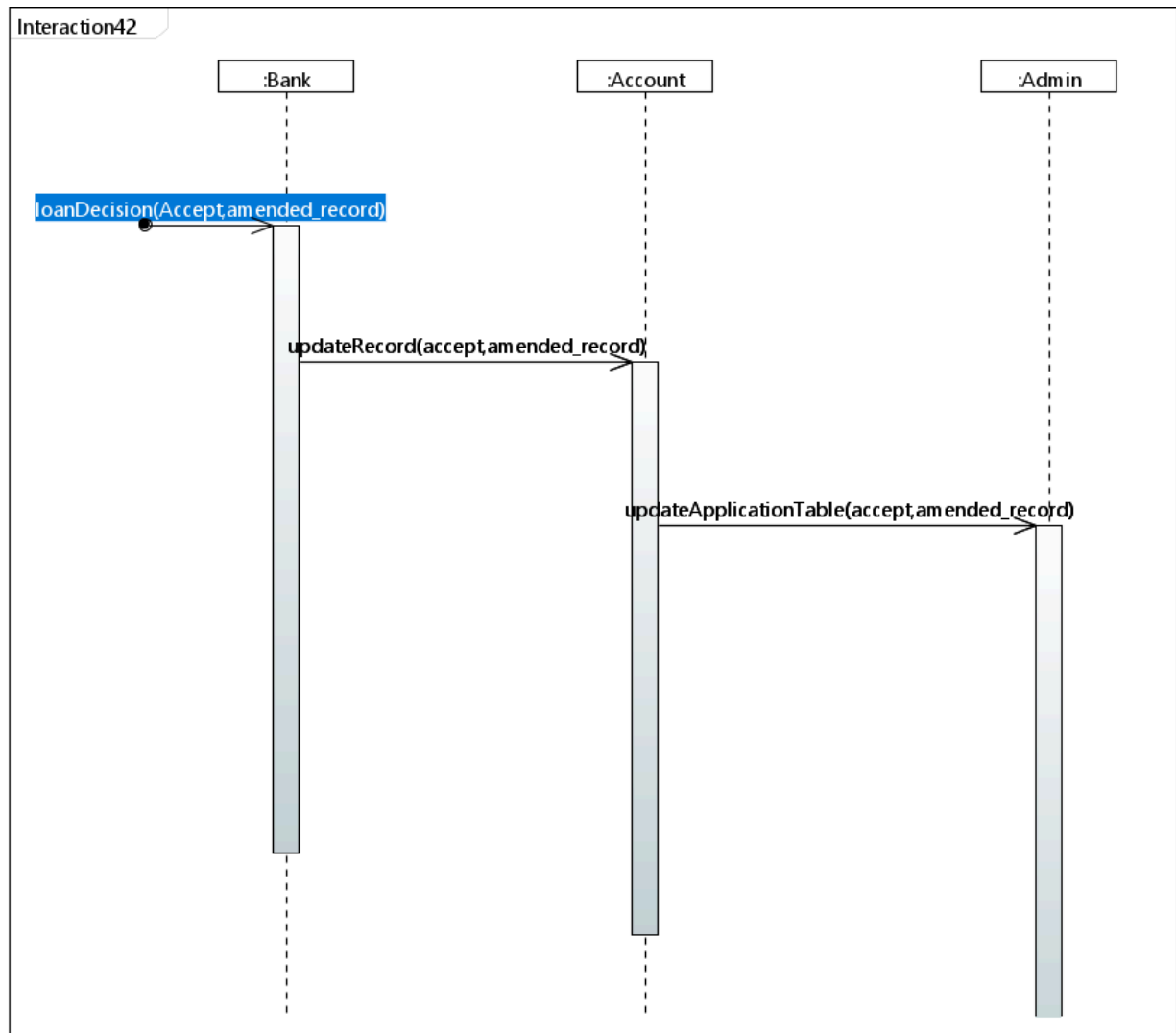


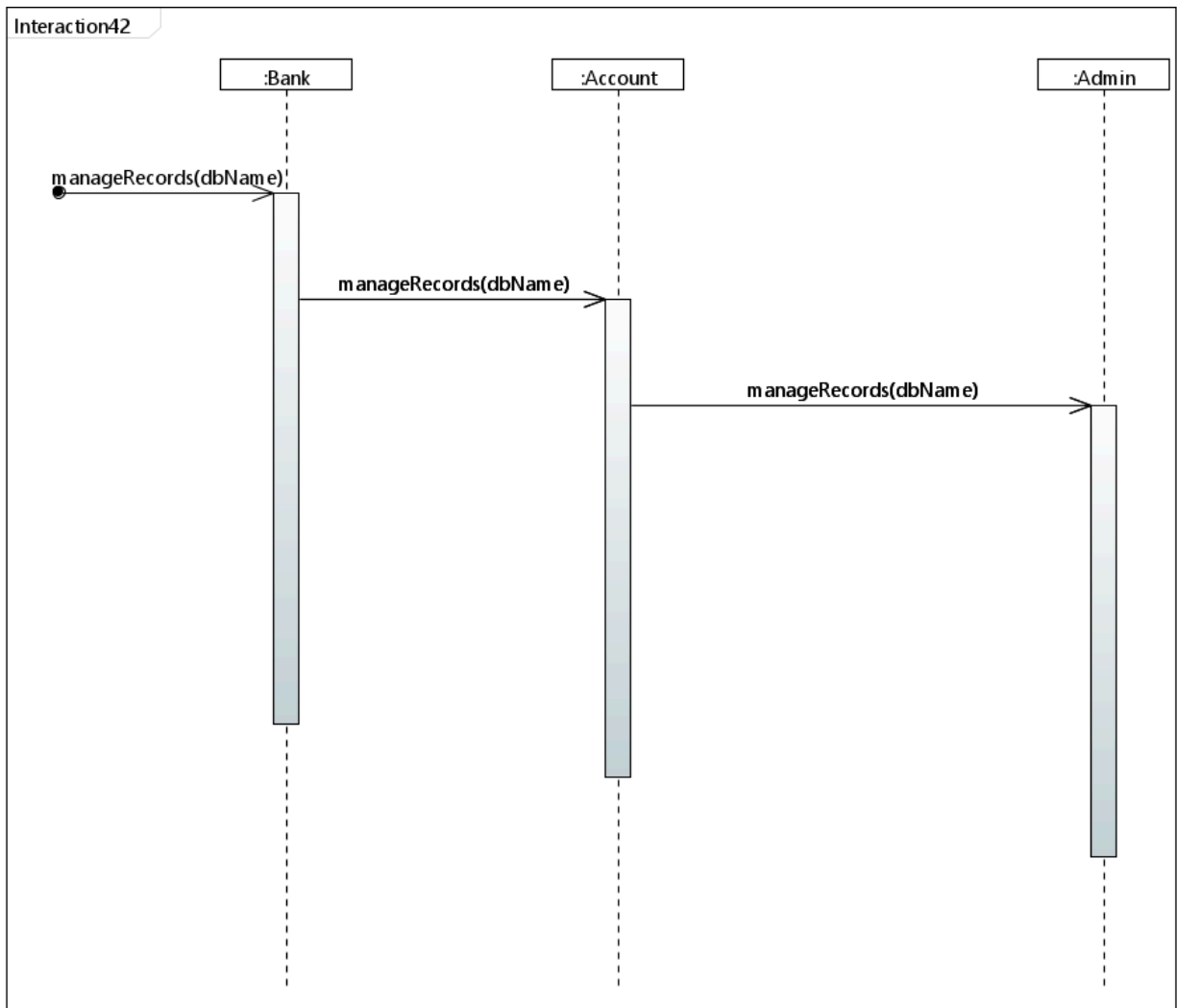


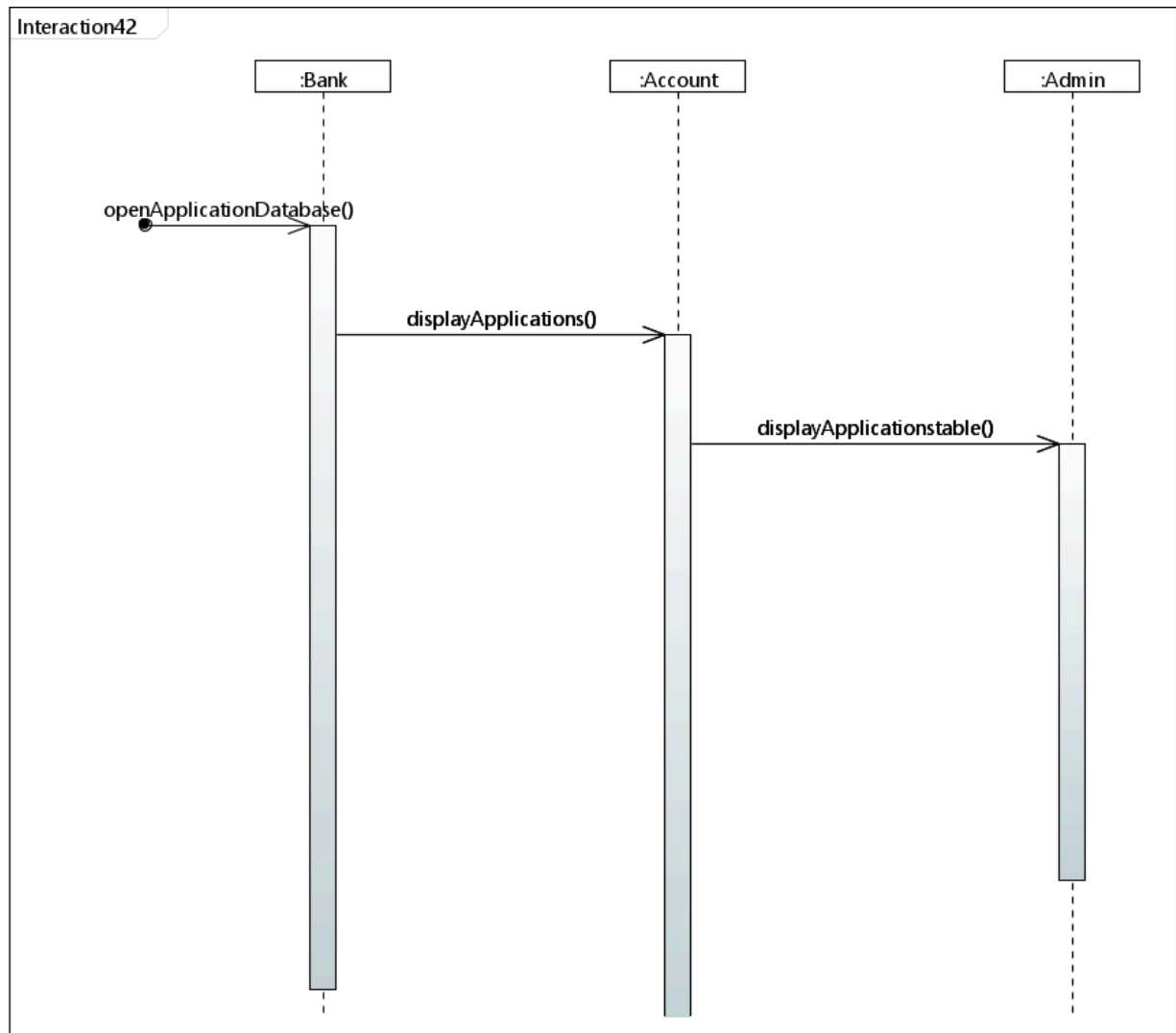


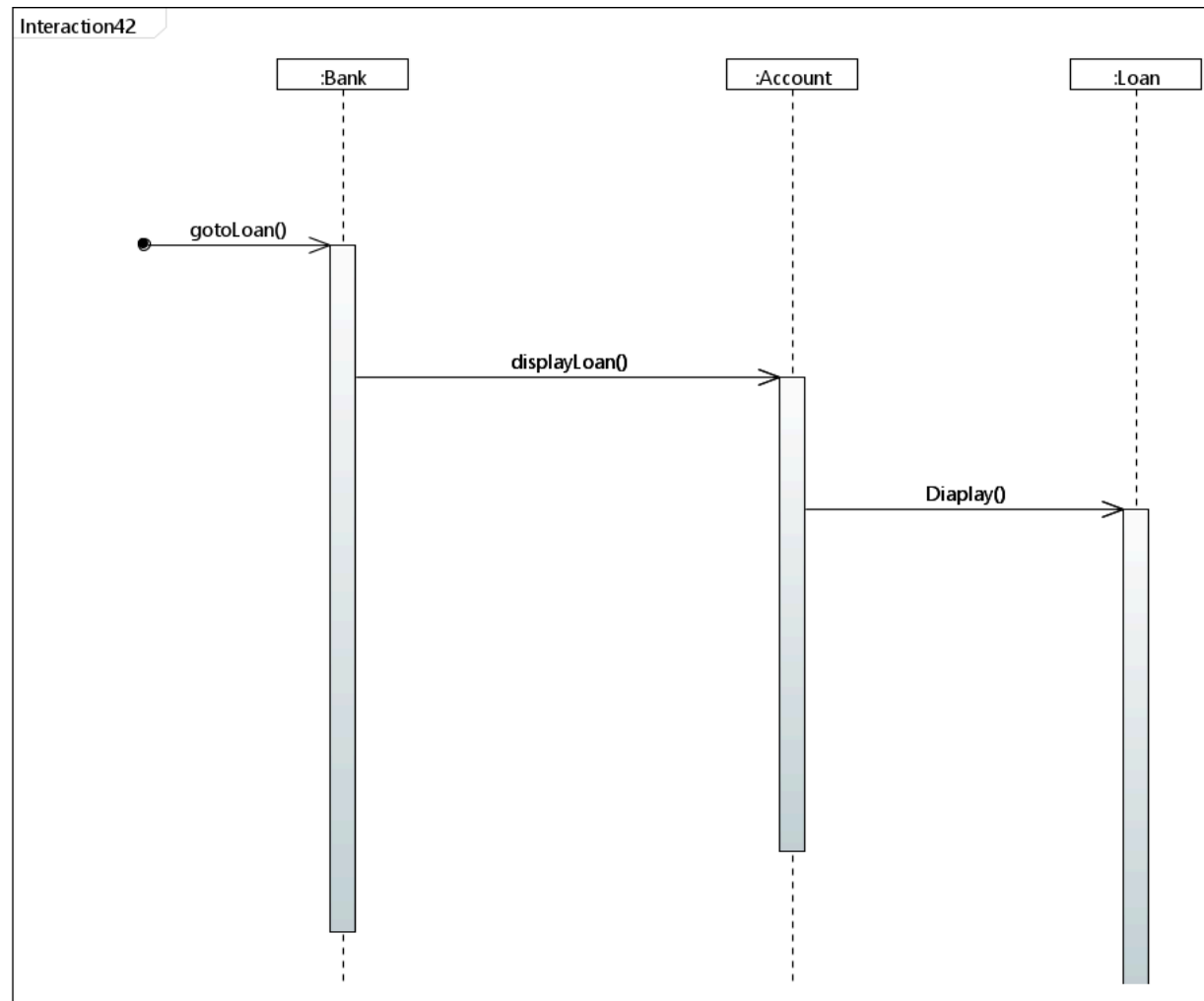


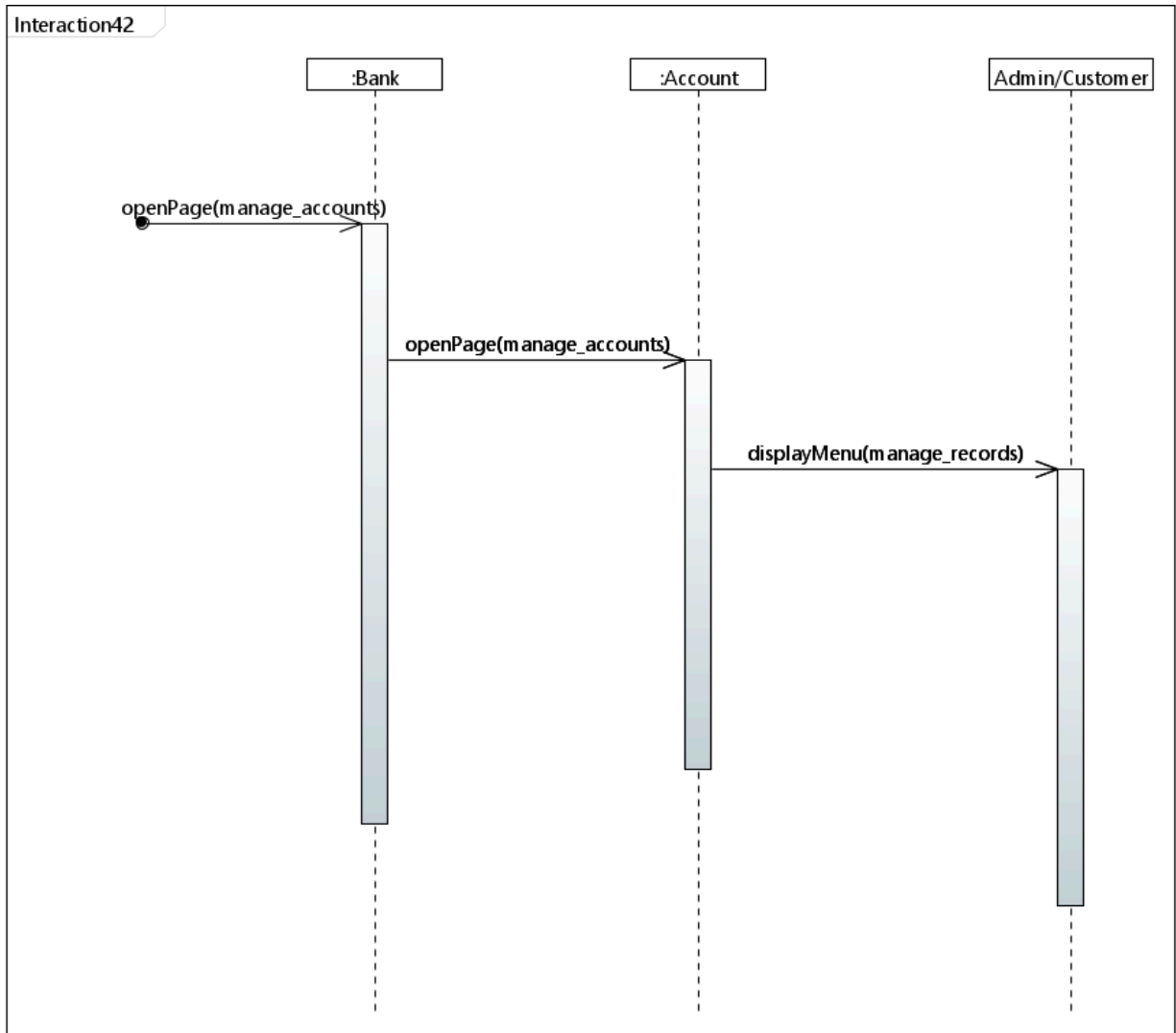




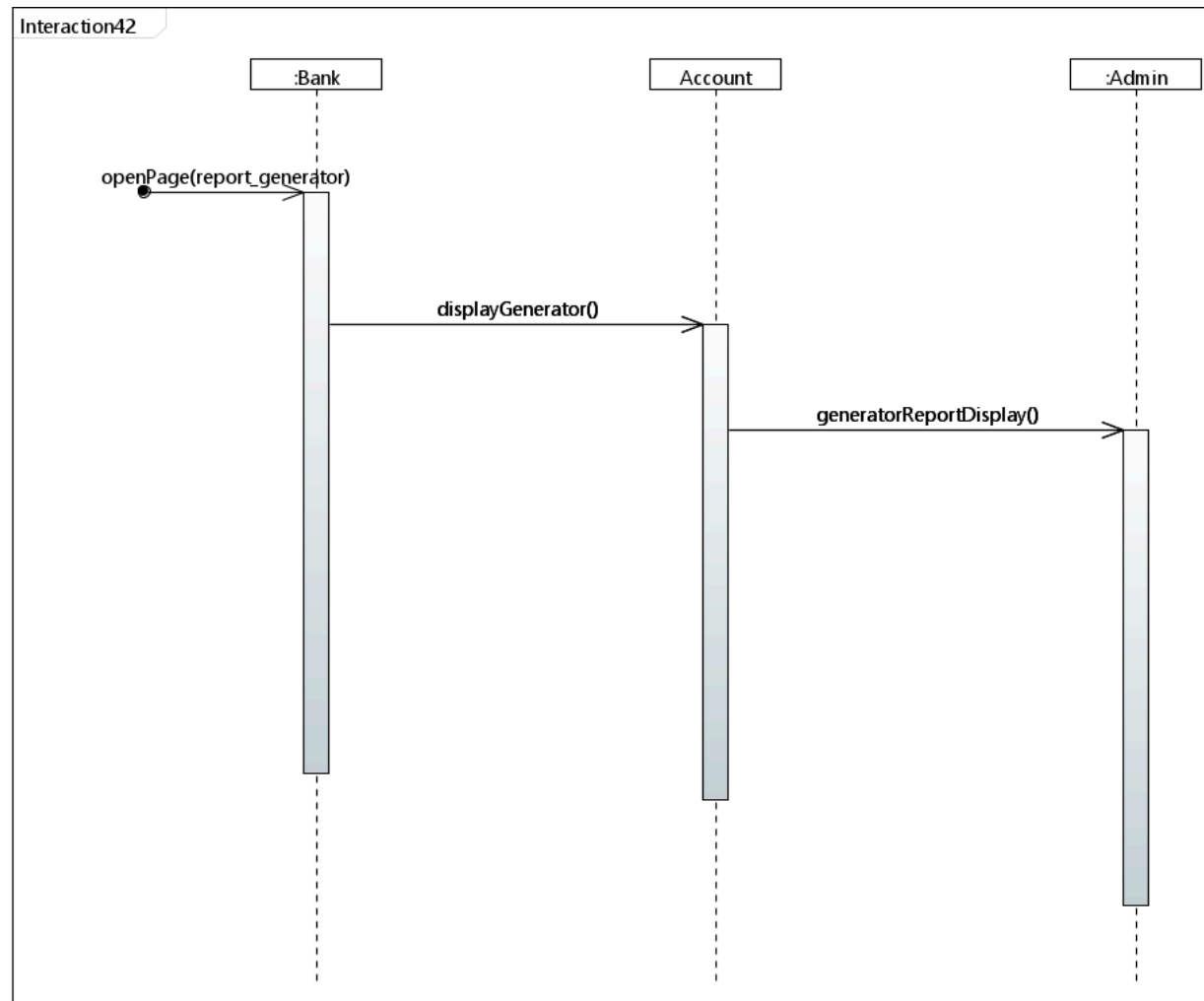


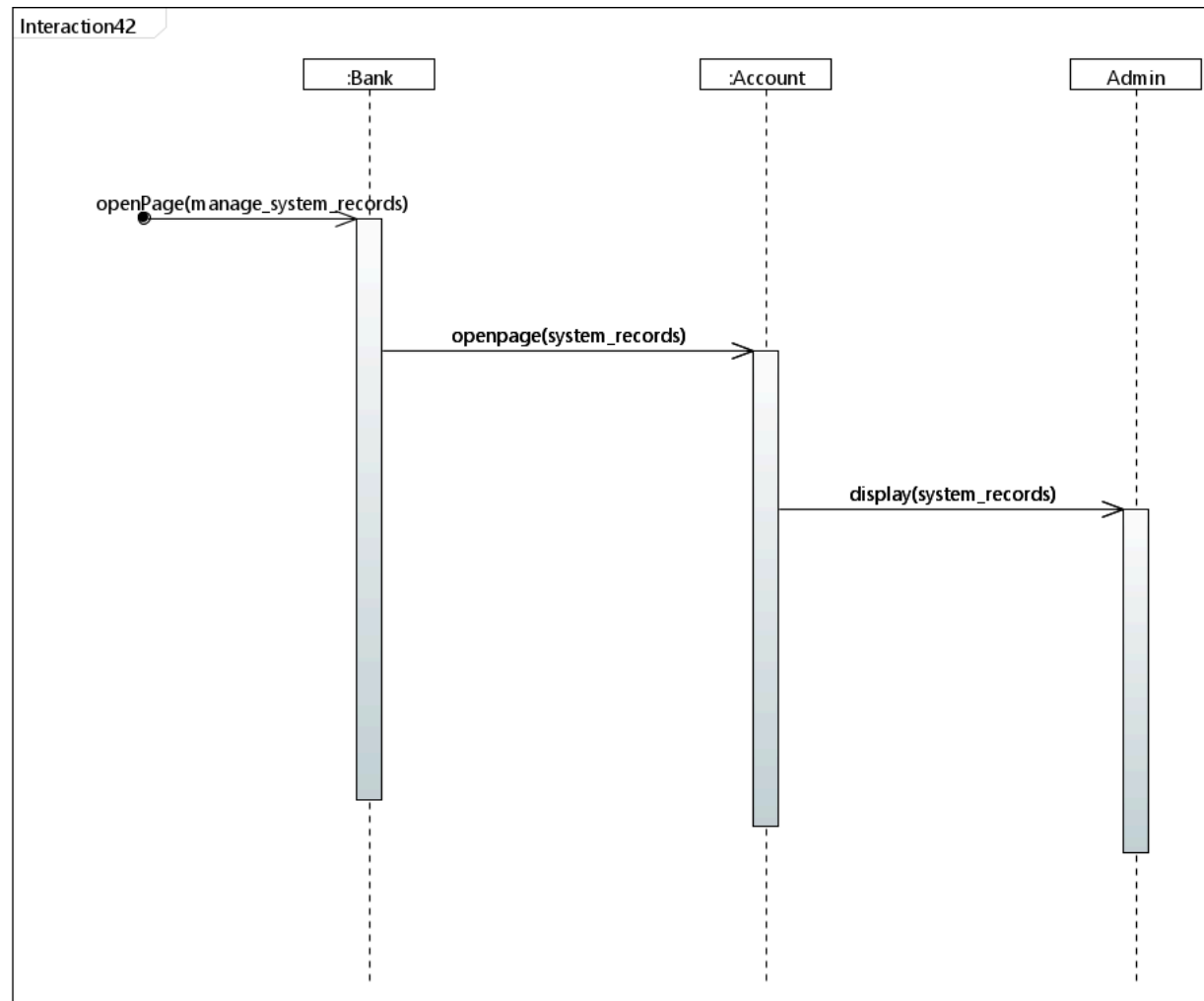


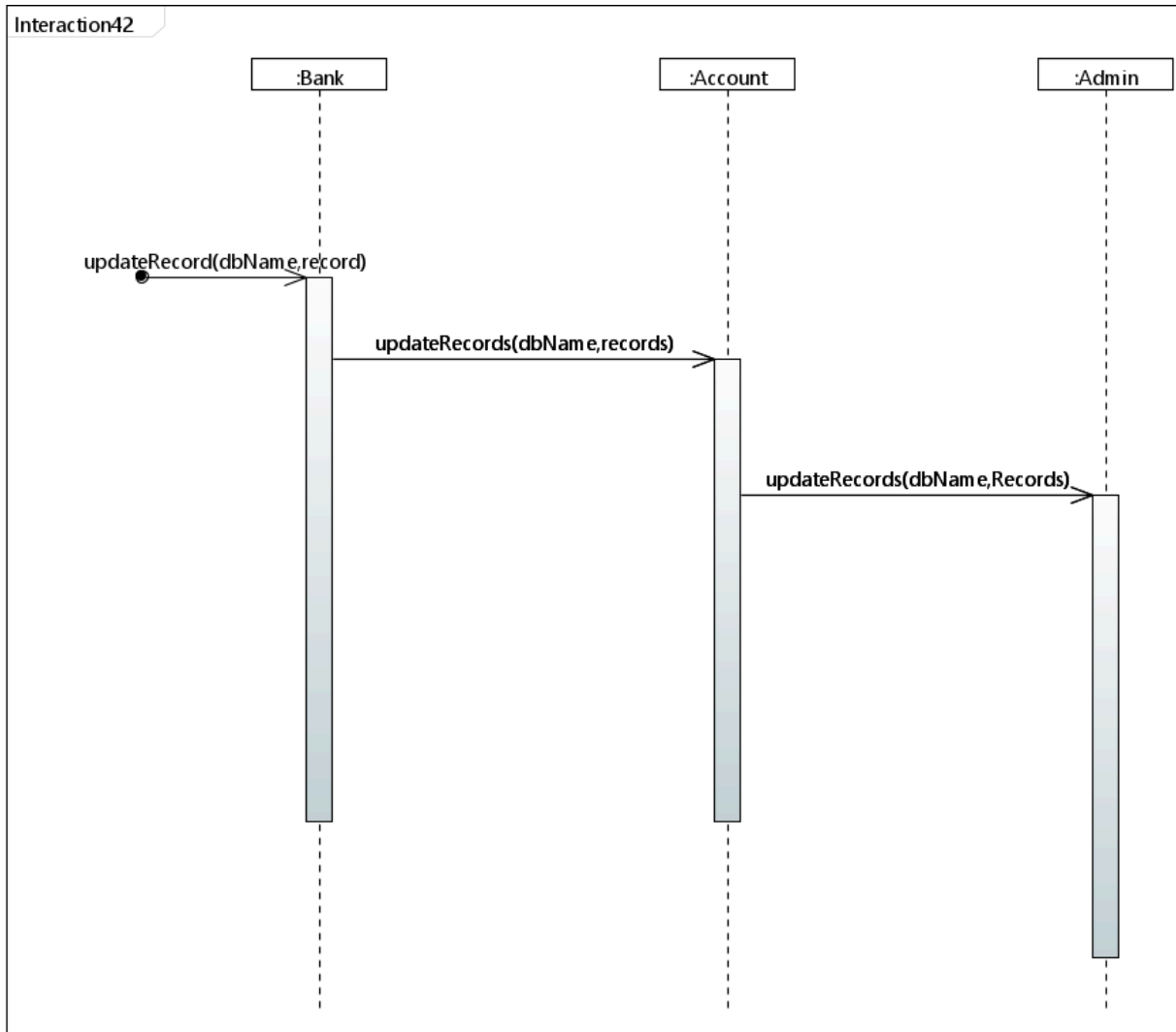




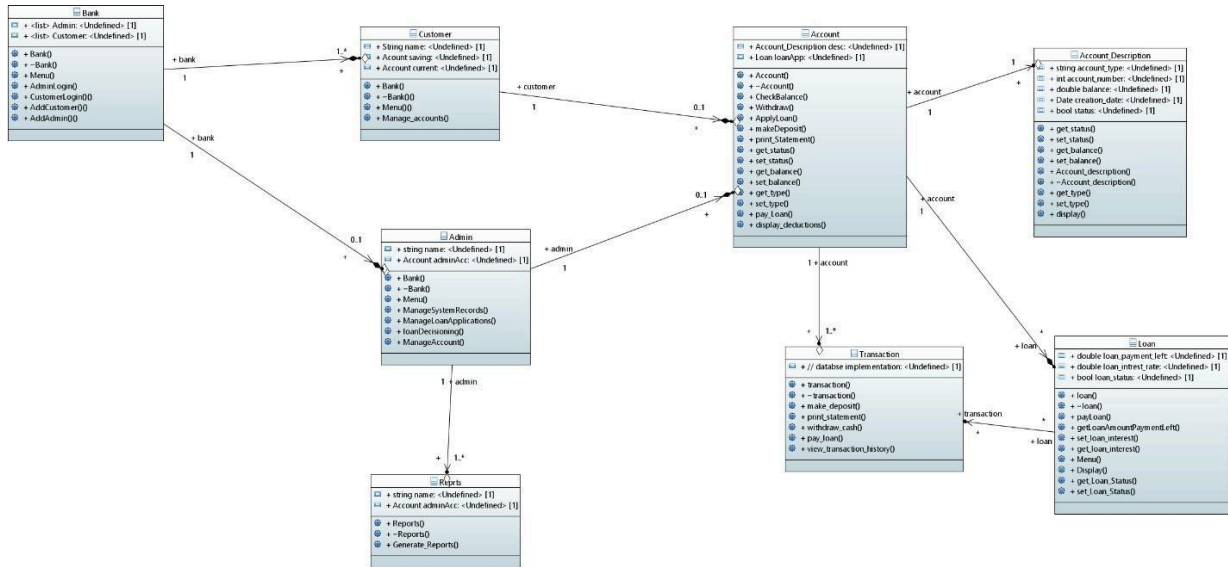




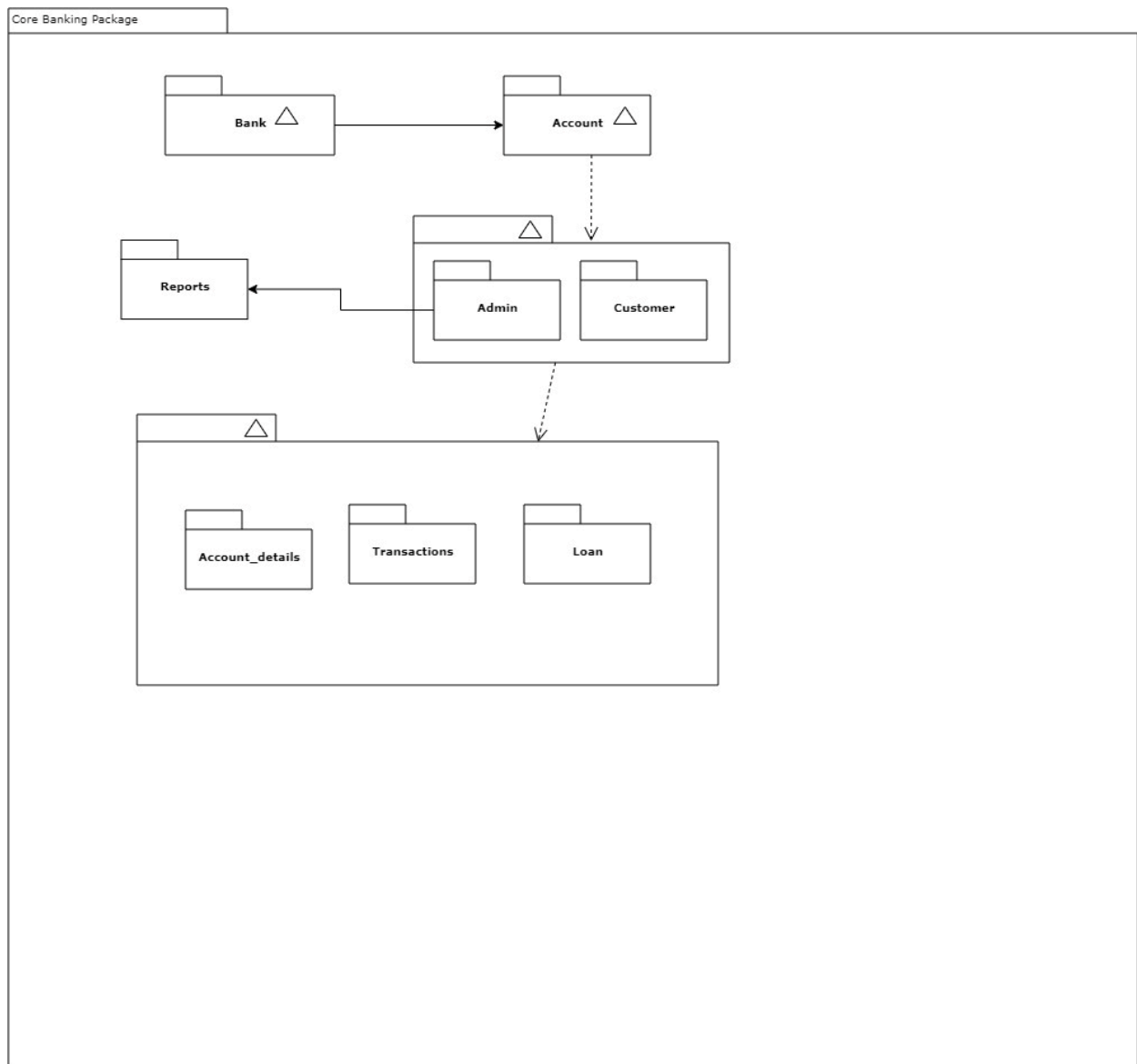




## 7 Class Diagram



## 8 Package Diagram



## 9 Deployment Diagram

