Software Requirements and Design Document

for

Bank Management System

Prepared by Fahad Kamran

Fradez Tech

27TH November 2022

Table of Contents

1. Introduction

1.1 Purpose

An application that can be operation on various operating systems under the same code using java, javafx and oracle 11g enterprise edition.

1.2 Product Scope

Product has a scope of a single bank. To be used by either the bank's admin or its customers. Admin can handle customer loan requests in real time and customers can apply for a single loan at a time along with multiple services.

1.3 Title

Bank management system.

1.4 Objectives

Major aims achieved through this project are interactive user interface, easily to understand and use, fast processing and avoiding unnecessary data storage in separate files, rather using a single database. To make an easier to use bank program with easy usage of database.

1.5 Problem Statement

Banks rely a lot on hard copy of data due to loss of their files, by making a better program which is loss less and secure, bank won't need to store lots of hard copy documents. Transactions stored in a better format. System in sync with recent times so will last longer than previously used system.

2. Overall Description

2.1 Product Perspective

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Product Functions

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

2.3 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

2.4 Extended Use Cases

Use Case Name: Make Transaction

Scope: Bank Management System

Level: Usergroup

Primary Actor: Customer

Stakeholders & Interests:

- Customer Making the payment to acquire product/service of need/want
- Acquiring Bank that will receive the payment capital
- Customer's Bank which will be managing the transactions through bank system

Precondition:

- Customer wants to make purchase
- Customer has enough balance to make purchase
- System is active

Postcondition:

- Customer has made payment and transaction is successful
- Customer receives purchased item/service
- System records updated

Main Success Scenario:

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

Use Case Name: Check Balance

Scope: Bank Management System

Level: Usergroup

Primary Actor: Customer

Stakeholders & Interests:

- Customer has satisfaction with having plenty of money or knowing how much more they need to earn
- Bank with having more capital in the bank vault

Precondition:

- System is active
- Customer is logged in
- Customer has account/s
- Customer wants to check balance

Postcondition:

- Customer attains information about the balance state of his account and leaved
- Customer is logged out of system

Main Success Scenario:

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

Extensions:

1. System is not active
2. Login failure
   a) Due to wrong password
   b) Due to poor internet connection
   c) Due to failure of system response
3. System fails to load menu
4. Customer can not see checking balance in menu
5. System fails to load balance page/ shows wrong balance
6. Customer can not spot where balance is written on the page
7. Customer does not log out of account

Use Case Name: Display Account Information/History

Scope: Bank Management System

Level: Usergroup

Primary Actor: Customer

Stakeholders & Interests:

- Customer has the interest to know if account details are correct
- Customer/bank admin may want to view account history of use and logins

Precondition:

- System is active
- System has option to display account information/history
- Customer wants to check information/history

Postcondition:

- Customer acquired information

Main Success Scenario:

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

Extensions:

1. System is inactive so customer cannot use it
2. Customer cannot login to system
   a) Due to wrong password entry
   b) Due to poor internet connection
   c) Due to system failure
3. System fails to display menu
4. System does not contain option to display account information/history
5. System fails to open page
6. Customer cannot view/manage account information
7. System fails to update records
8. Customer does not log out

Use Case Name: Transfer Amount

Scope: Bank Management System

Level: Usergroup

Primary Actor: Customer

Stakeholders & Interests:

- Customer having interest in the amount of transfer
- Recipient having interest in receiving the amount
- Customer's bank having interest in the transfer of amount
- Recipient's bank having interest in handling the receiving of the amount

Precondition:

- System is active
- Customer wants to transfer amount
- Customer can access the system

Postcondition:

- Transfer made successfully
- System is updated

Main Success Scenario:

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

Extensions:

1. Customer cannot access the system as it is inactive
2. Customer fails to log into system
   a) Due to wrong password
   b) Due to poor internet connection
   c) Due to failure of system
3. System fails to display menu
4. Menu does not contain option to transfer amount
5. System fails to open the page due to server failure or poor internet connection
6. Customer cannot enter details into page
7. Customer forgets to press enter
8. System does not manage the transfer correctly
9. Recipient does not receive amount
10. Customer does not log out of system

Use Case Name: Make Withdrawal
Scope: Bank Management System
Level: Usergroup
Primary Actor: Customer
Stakeholders & Interests:

- Customer having interest in the amount of cash to withdraw
- Bank having interest in the amount of cash is being taken from their vault/atm

Precondition:

- System is active
- Customer wants to make a withdrawal
- System has enough cash for withdrawal

Postcondition:

- Customer leaves with correctly received cash
- Customer account balance is updated

Main Success Scenario:

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

Extensions:

1. Customer cannot access the system
2. Customer fails to login
3. System does not display menu
4. There is no option to make withdrawal
5. System fails to open the page for withdrawal
6. Customer cannot enter amount
7. System does not ask for confirmation and ends up withdrawing more/less cash than customer wanted
8. Customer cancels confirmation
9. System does not have enough cash to manage withdrawal or fails to withdraw with customer balance deducted
10. System does not update customer balance
11. Customer does not receive cash of correct amount
12. Customer does not log out

Use Case Name: Track Loan Status/Progress

Scope: Bank Management System

Level: Usergroup

Primary Actor: Customer

Stakeholders & Interests:

- Customer having interest in checking the amount of money he needs to pay the bank by a fixed date
- Bank having interest in receiving payments on time from customer

Precondition:

- System is active
- Customer wants to track loan progress
- Customer has a loan in progress with the bank

Postcondition:

- Customer has identified how much is due and how much are paid already
- Bank has identified how much more money they will receive

Main Success Scenario:

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

Extensions:

1. Customer fails to access system
2. Customer fails to login to system
3. System fails to load the menu
4. There is no option to track loan progress
5. System fails to load page of loan information
6. Customer cannot see necessary information
7. Customer does not log out

Use Case Name: Login of Customer/Admin

Scope: Bank Management System

Level: Usergroup

Primary Actor: Customer & Admin

Stakeholders & Interests:

- Customer having interest to be able to access the system
- Admin having interest to be able to access the system
- Bank having interest to be able to track the number of users they have

Precondition:

- System is active
- User has an account
- Internet connection is good
- User wants to use the system

Postcondition:

- User successfully accesses the system

Main Success Scenario:

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	

Extensions:

1. User cannot access the system
2. User cannot enter details
3. User enters invalid details
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 logins 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
Primary Actor: Customer & Admin
Stakeholders & Interests:

- Customer having interest in wanting a loan
- Admin having interest in providing a loan which will profit the bank

Precondition:

- System is active
- Customer wants a loan
- Customer has good financial status for loan
- Admin can hand such loan profitably
- Bank has enough capital to give the loan

Postcondition:

- Customer receives money on loan
- System is updated

Main Success Scenario:

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

Extensions:

1. Customer fails to access system
2. Customer cannot login to system
3. System does not display menu page
4. Customer cannot see loan applications option
5. System fails to open loan application page
6. Customer cannot enter information in loan application form
7. Customer does not press send
8. System does not add file to loan applications records
9. Admin does not arrive at system
10. Admin does not access customer's loan application
11. Admin does not check customer's financial background
12.
    a) Admin reject loan application
    b) Admin enters wrong information for loan
    c) Admin fails to press save
    d) Admin loses internet connection
    e) Admin accidentally deletes customer's loan application
13.
    a) Admin reject loan application
    b) Admin enters wrong information for loan
    c) Admin fails to press save
    d) Admin loses internet connection
    e) Admin accidentally deletes customer's loan application
14. Admin does not enter further information for loan
15. Admin does not logout of system
16. System does not update records
17. System does not add loan amount to customer's balance and/or deduct amount from bank's own balance
18. Customer does not receive message of approval and/or with the balance updated
19. Customer does not log out of system

Use Case Name: Manage Transaction History
Scope: Bank Management System
Level: Usergroup
Primary Actor: Admin
Stakeholders & Interests: - Customer having interest in viewing their transaction history and pointing out errors

- 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 logins 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

Use Case Name: Manage System Records
Scope: Bank Management System
Level: Usergroup
Primary Actor: Admin
Stakeholders & Interests:
- Admin has interest in fixing and maintaining system records to avoid errors and malfunctions
- Customer has interest to has the errors in their bank accounts fixed
- Bank has interest in making sure their system is running smoothly

Precondition:
- System is active
- Admin needs/wants to manage records
- Admin can access system

Postcondition:
- System records update
- Changes to records made successfully

Main Success Scenario:

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

Use Case Name: Generate Reports

Scope: Bank Management System

Level: Usergroup

Primary Actor: Admin

Stakeholders & Interests:

- Admin having interest to generate reports
- Manager having interest in receiving reports of the bank's profits, financial status, etc.
- Customer having interest in receiving their balance statement report

Precondition:

- System is active
- Admins wants/needs to generate reports
- Admin can access the system
- There is enough data to generate reports

Postcondition:

- Report/s generated successfully

Main Success Scenario:

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

Extensions:

1. Admin fails to access system
2. Admin fails to log into system
3. System fails to load main page or display menu
4. Admin cannot see/access option to generate reports
5. System fails to load page of generate reports
6. System does not ask admin for report type and relevant information
7. Admin enters wrong type and information for the report
8. System fails to generate report
   a) Due to poor internet connect
   b) Due to invalid query conditions
   c) Due to data and type mismatch
9.
   a) System fails to display report in screen
   b) System fails to provide option to print/save
10. Admin does not choose to print/save
11. System fails to print/download report
12. Admin does not log out of system

Use Case Name: Add Account

Scope: Bank Management System

Level: Usergroup

Primary Actor: Admin

Stakeholders & Interests:

- Admin having interest in gaining customer
- Customer having interest in holding an account in the bank
- Customer's occupation company having interest in which bank and bank account to send the customer's salary
- Bank having interest in gaining number of accounts being used

Precondition:

- System is active
- Customer wants to open an account
- Admin can access the system
- Admin is present

Postcondition:

- Account created successfully
- User has his account login information
- System is updated

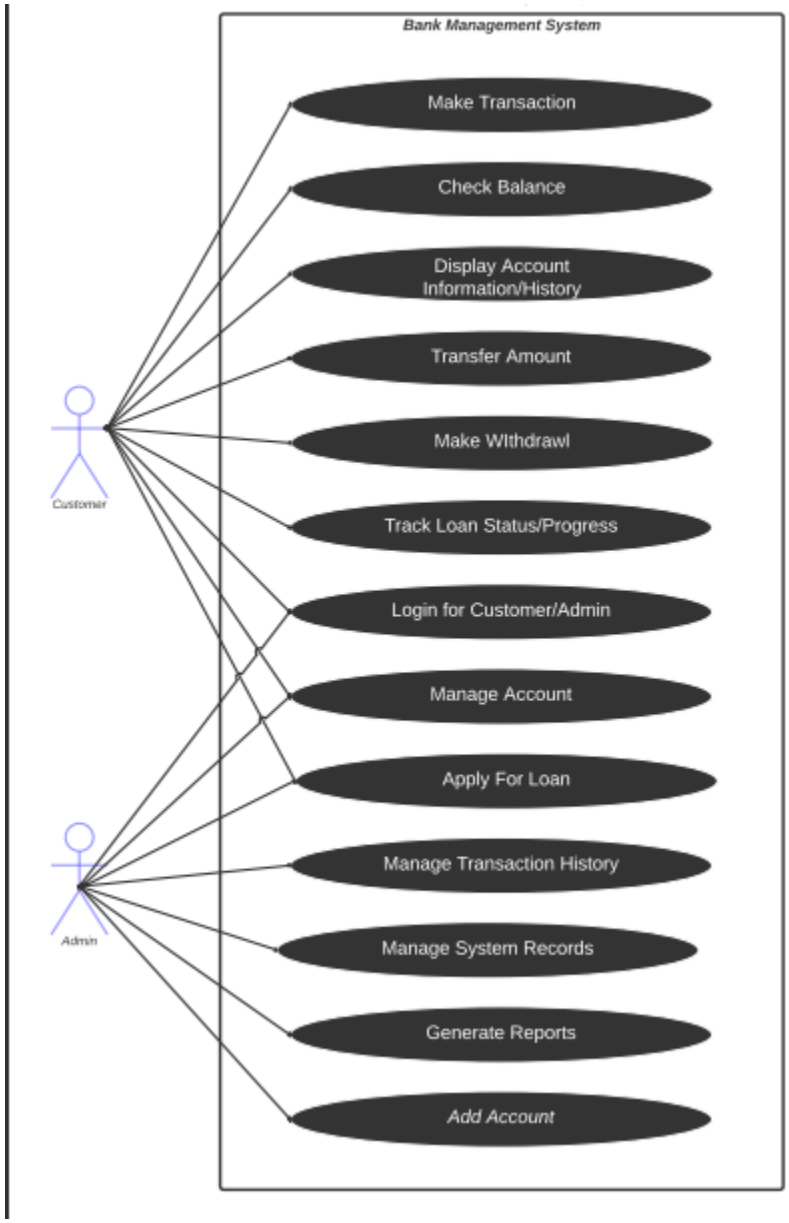
Main Success Scenario:

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

Extensions:

1. Customer fails to arrive at registration point
2. Admin fails to access the system
3. Admin fails to login to system
4. System fails to load main menu
5. Admin cannot find the registration option
6. System fails to load registration page
7. Admin does not ask all necessary details from customer
8. Customer does not give correct details
9. Admin cannot enter details into system or fails to enter all details
10. Admin does not press 'create account'
11. System does not create new account with/without user data
12. Admin does not give customer his/her username and password
13. Customer does not receive login and password to access the system
14. Admin does not log out of system

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

Required the use of Oracle 11g with the set up database provided below, javafx sdk 8, java sdk 15, any operating system.
CODE for Oracle :

```
drop table BankLoanApplication;
drop table BankTransaction;
drop table BankLoan;
drop table BankAccount;
drop table BankAdmin;
drop table BankCustomer;

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 into BankCustomer values('1','Fahad
Kamran','+923333333333','fradez@gmail.com','fahadcust','fahad');
insert into BankCustomer values('2','Mahad
Kamran','+923333333334','mradez@gmail.com','mahadcust','mahad');
insert into BankCustomer values('3','Inaya Kamran','+923333333335','inaya@gmail.com','inayacust','inaya');

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 into BankAdmin values('1','Fahad Kamran
Kundi','+923333333336','fradez2@gmail.com','fahadadm','fahad');

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 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 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 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 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);
```

3.2 Safety Requirements

There is no such potential loss of data possible as all important data is stored in files and database tables.

3.3 Security Requirements

Oracle is a trusted secure system for securing data with outside access only through a set login system. Just make sure that the installed sql system is a secure version and not crack to avoid any hackable code segments.

3.4 Software Quality Attributes

All functionality is reusable with easy debugging and validation checking. Updates can easily be added without having to take the product offline for very long as each function is divided into separate classes controllers.

3.5 Business Rules

Only Admin can view all accounts, transactions, generate reports and make loan decisions where as customer can only access basic customer transaction facilities.

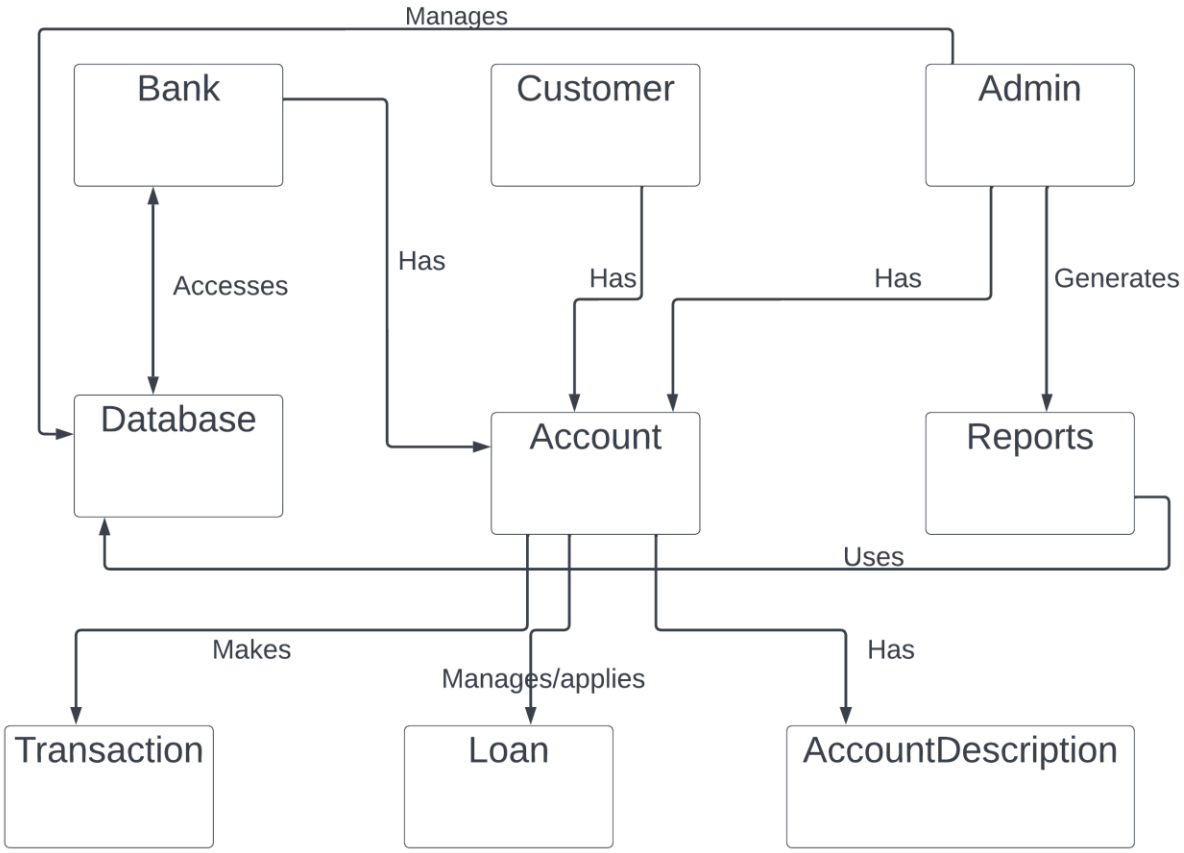
3.6 Operating Environment

As long as the system is up-to-date and necessary software specified above are installed, the program will run smoothly on any operating system.

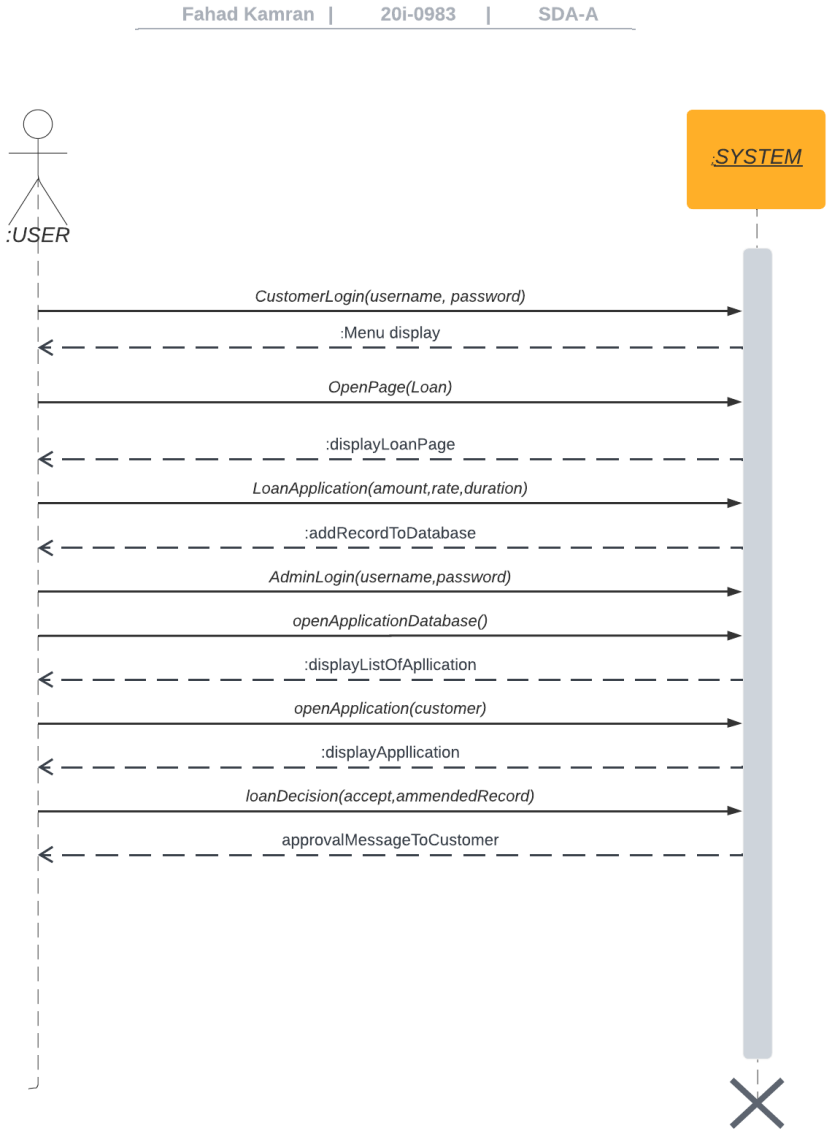
3.7 User Interfaces

The user Interface is simple and minimalistic with proper prompt messages to user incase of invalid entry in any particular field and guides user to making the right entry. User only has access to what they should have access to based on their account type and there data is securely kept in a database so that it does not get deleted and they can access/view their own data upon request to the software.

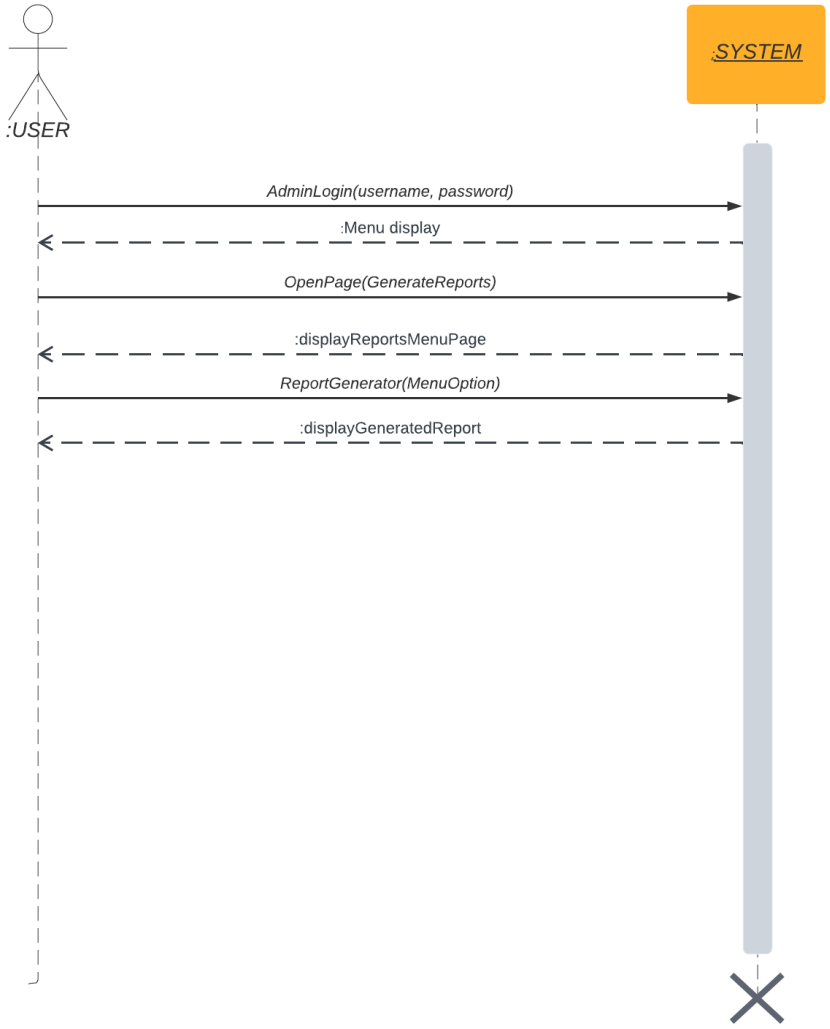
4. Domain Model



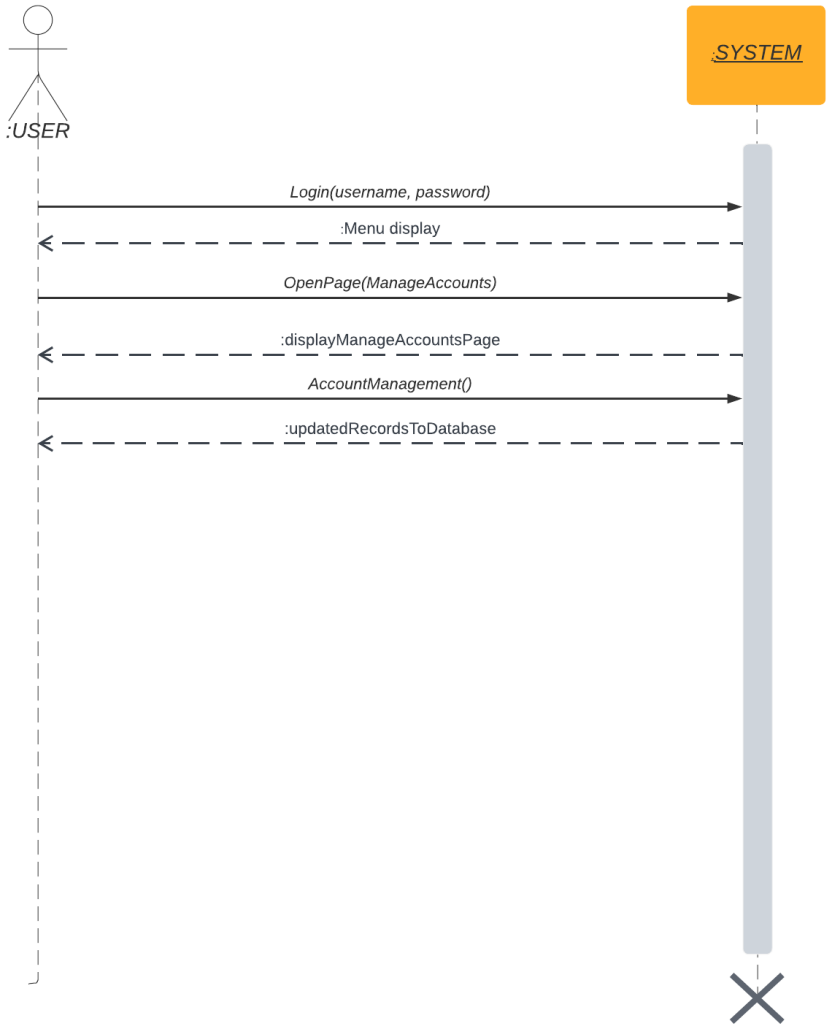
5. System Sequence Diagram

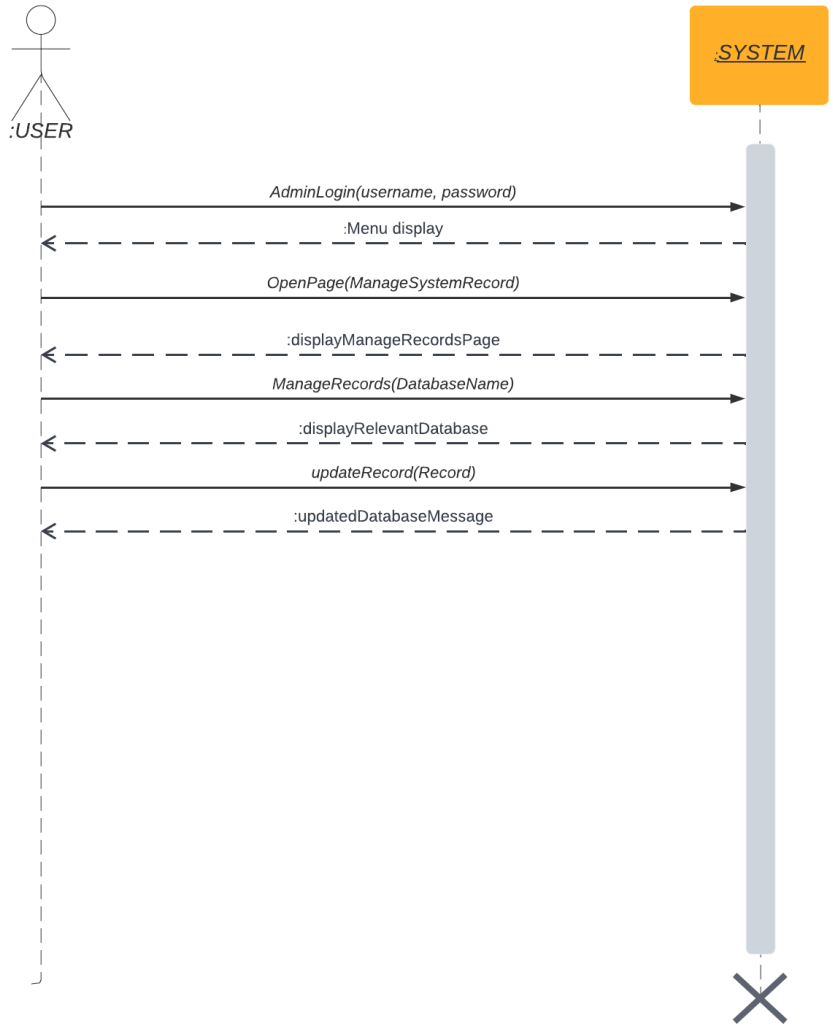


Fahad Kamran | 20i-0983 | SDA-A

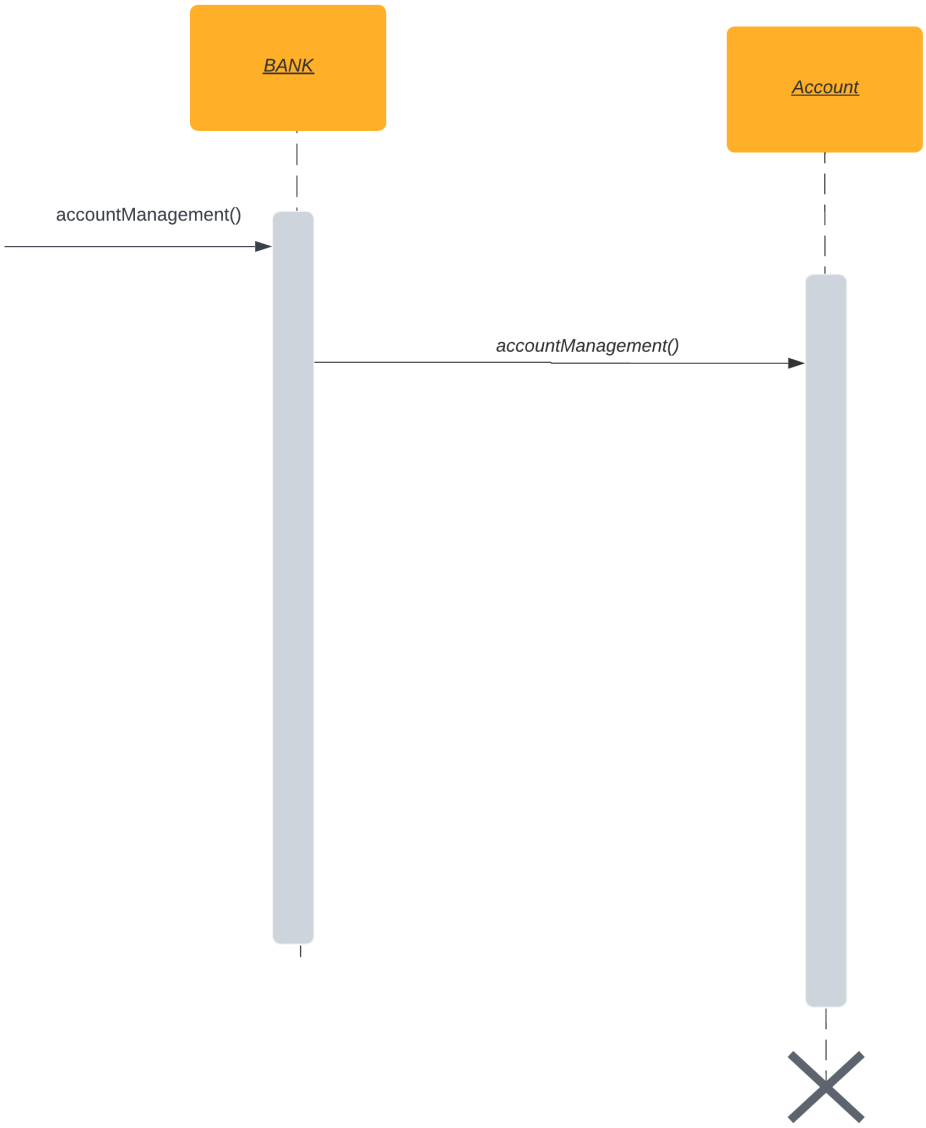


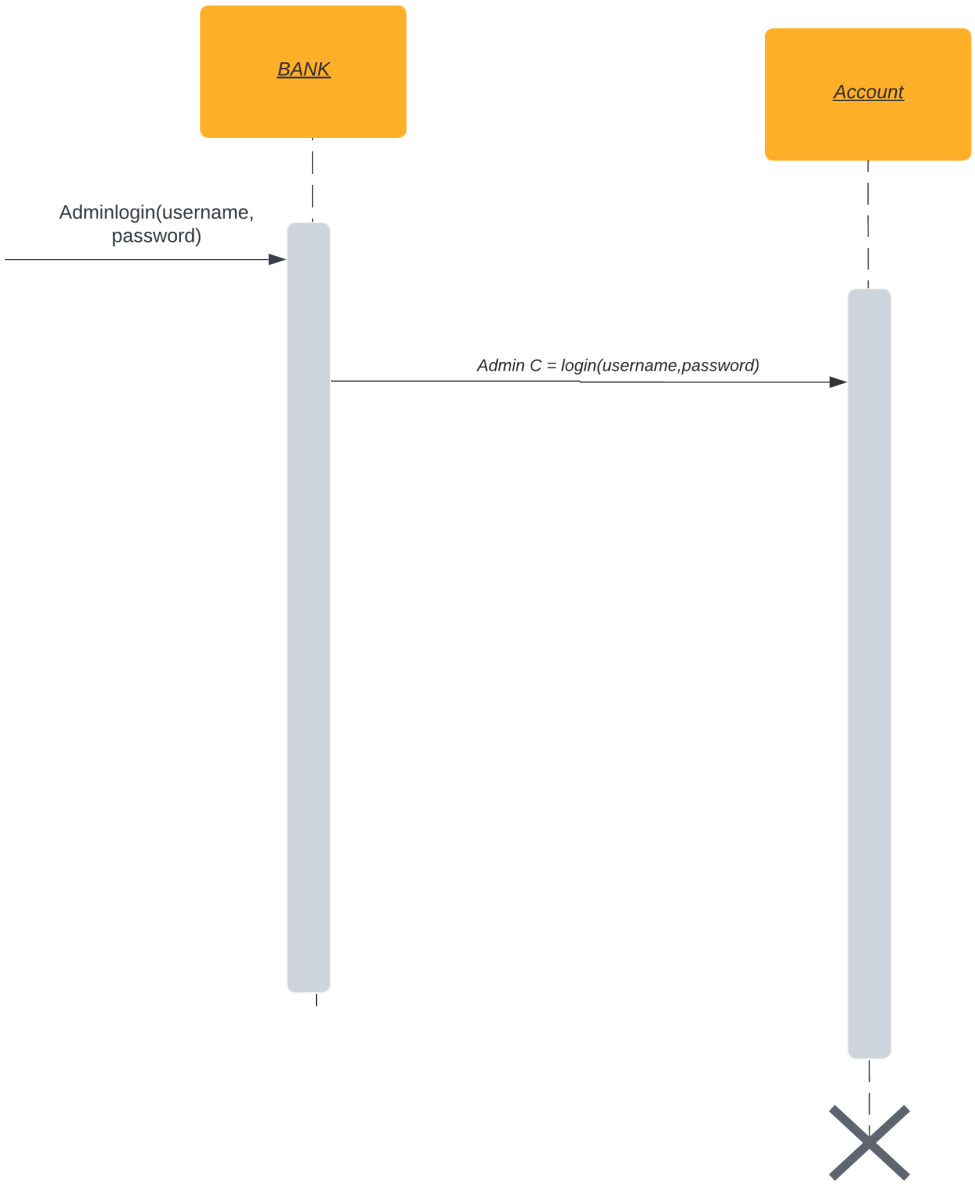
Fahad Kamran | 20i-0983 | SDA-A

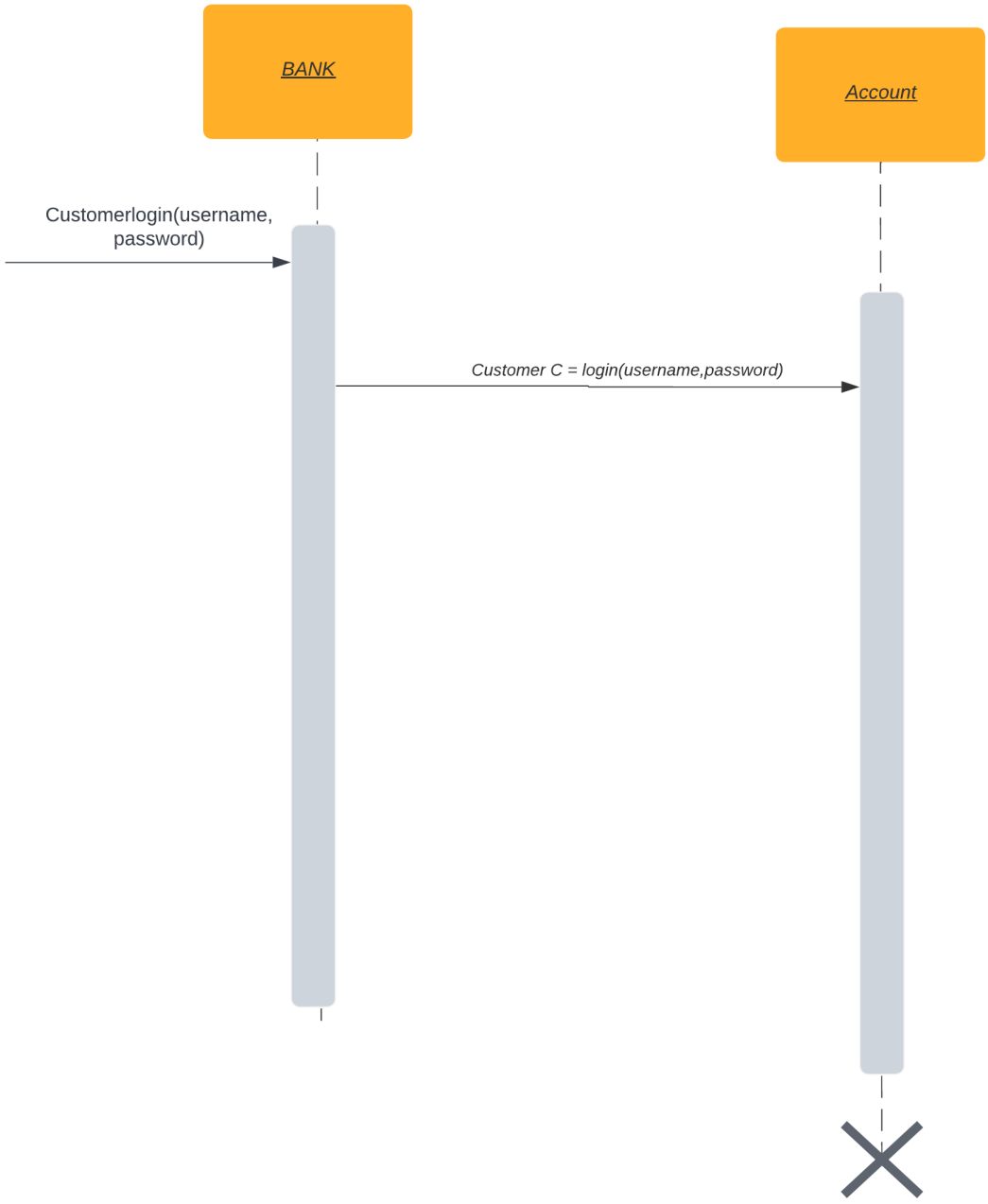


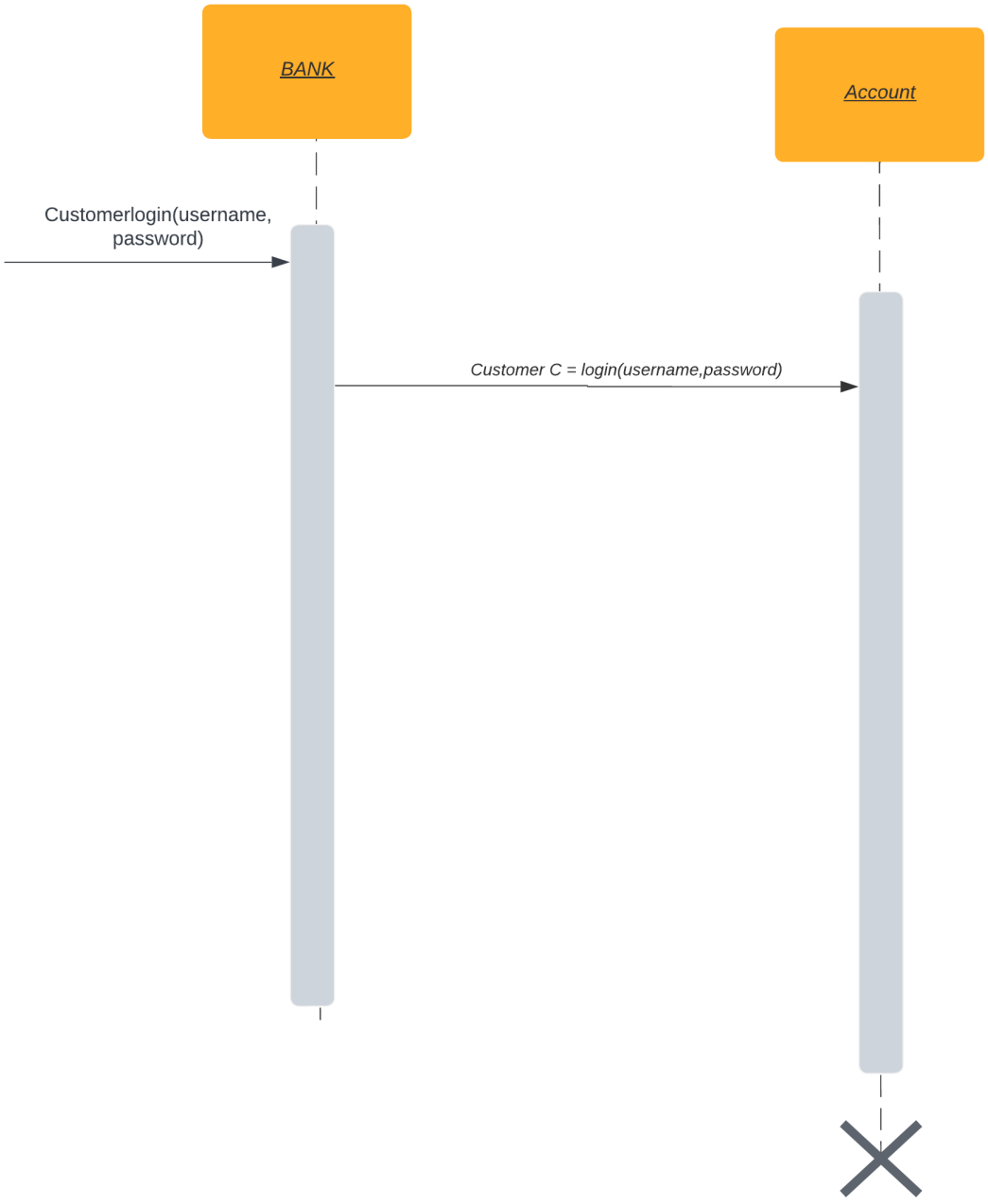

:USER
:SYSTEM
AdminLogin(username, password)
:Menu display
OpenPage(ManageSystemRecord)
:displayManageRecordsPage
ManageRecords(DatabaseName)
:displayRelevantDatabase
updateRecord(Record)
:updatedDatabaseMessage

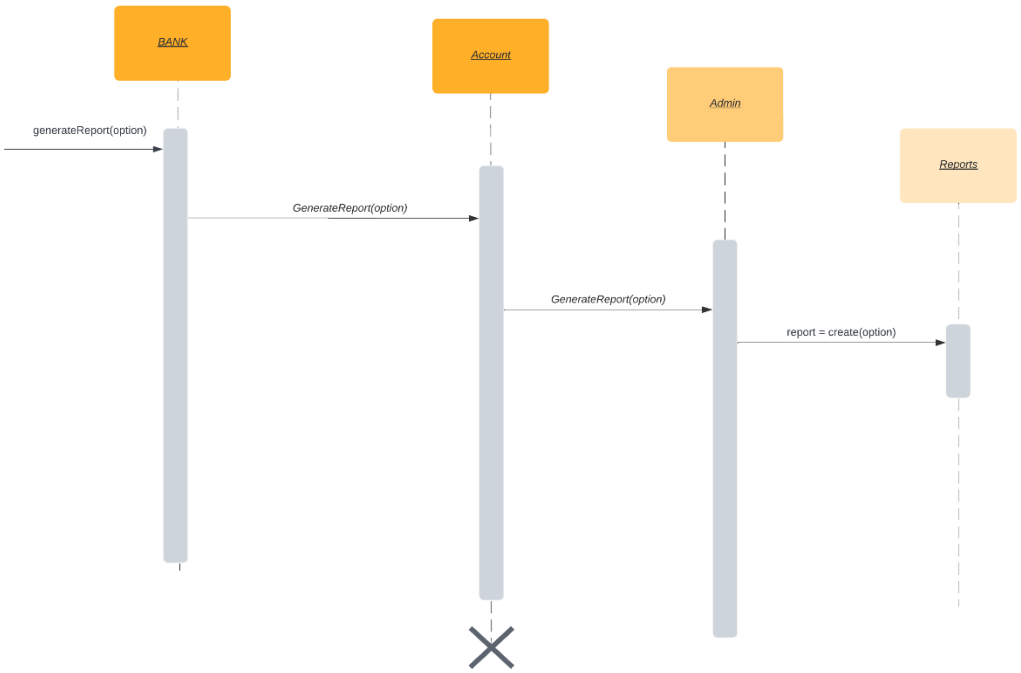
6. Sequence Diagram

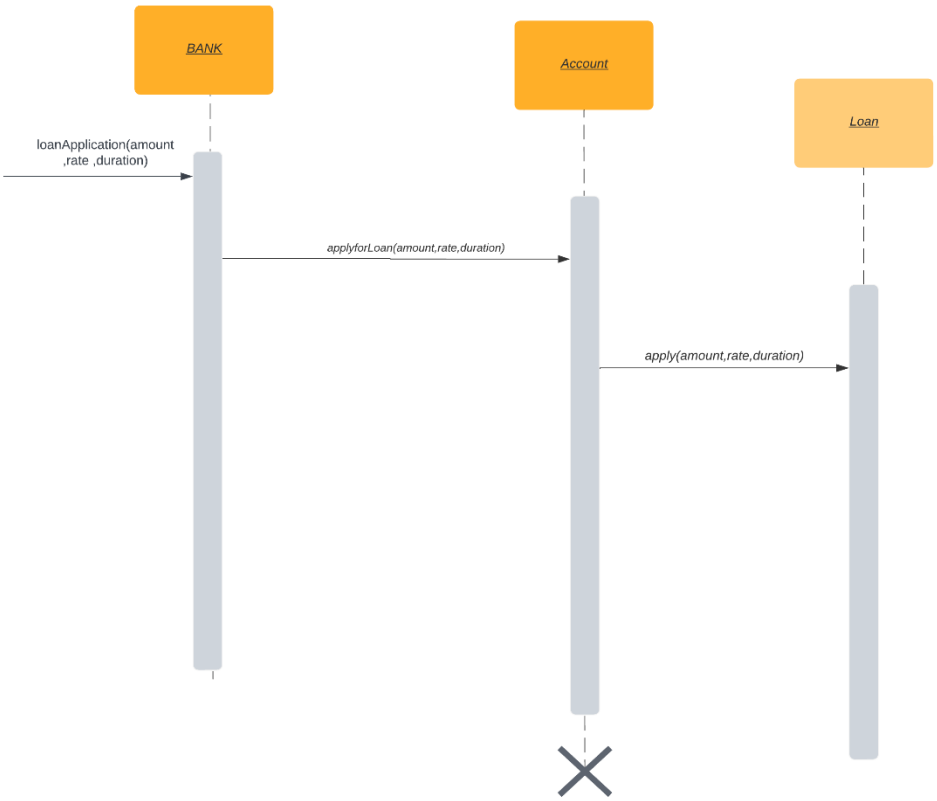


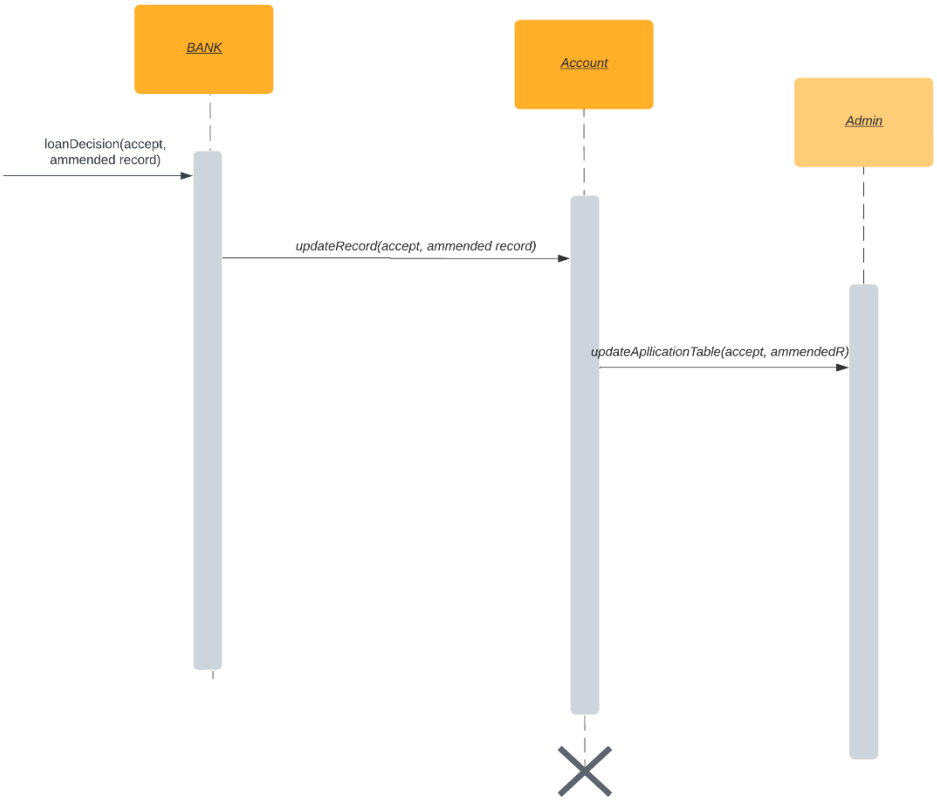

BANK
Account
Adminlogin(username, password)
Admin C = login(username,password)

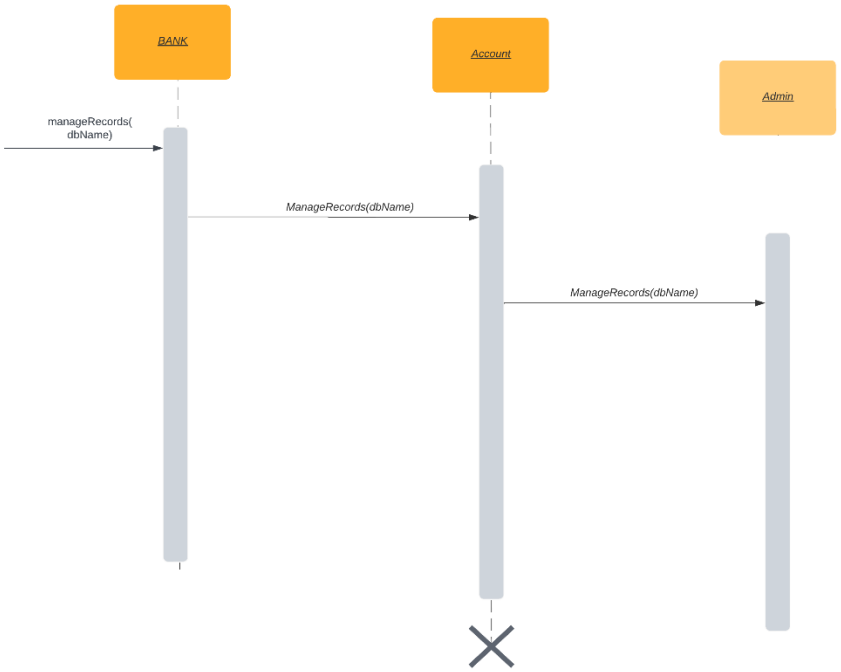

BANK
Account
Customerlogin(username, password)
Customer C = login(username,password)

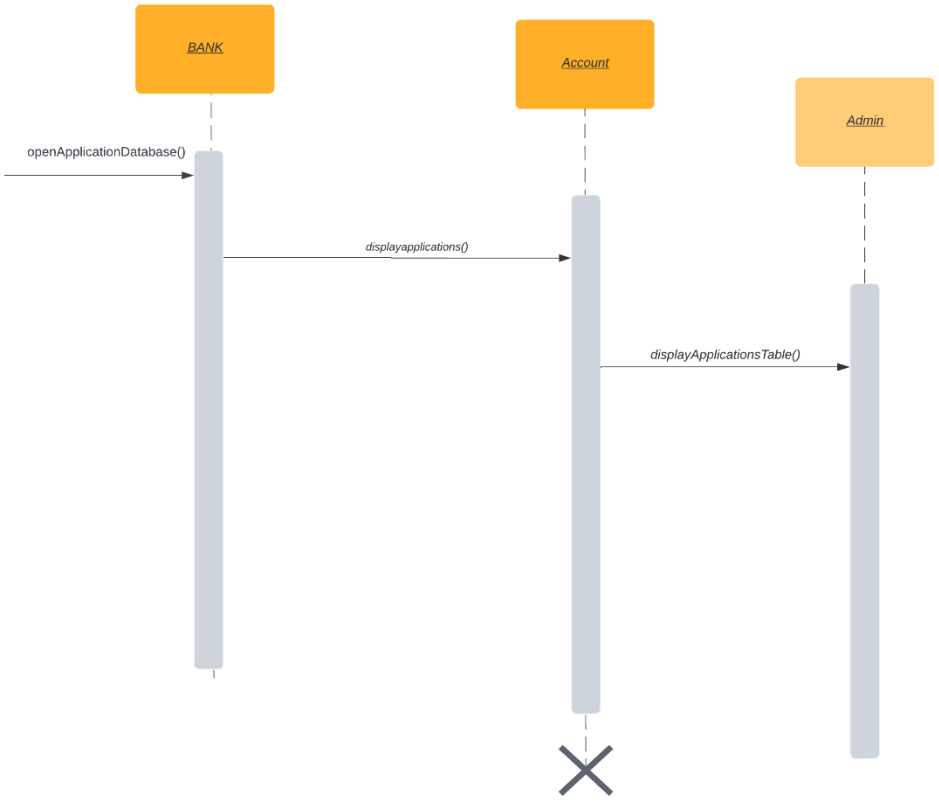

BANK
Account
Customerlogin(username, password)
Customer C = login(username,password)

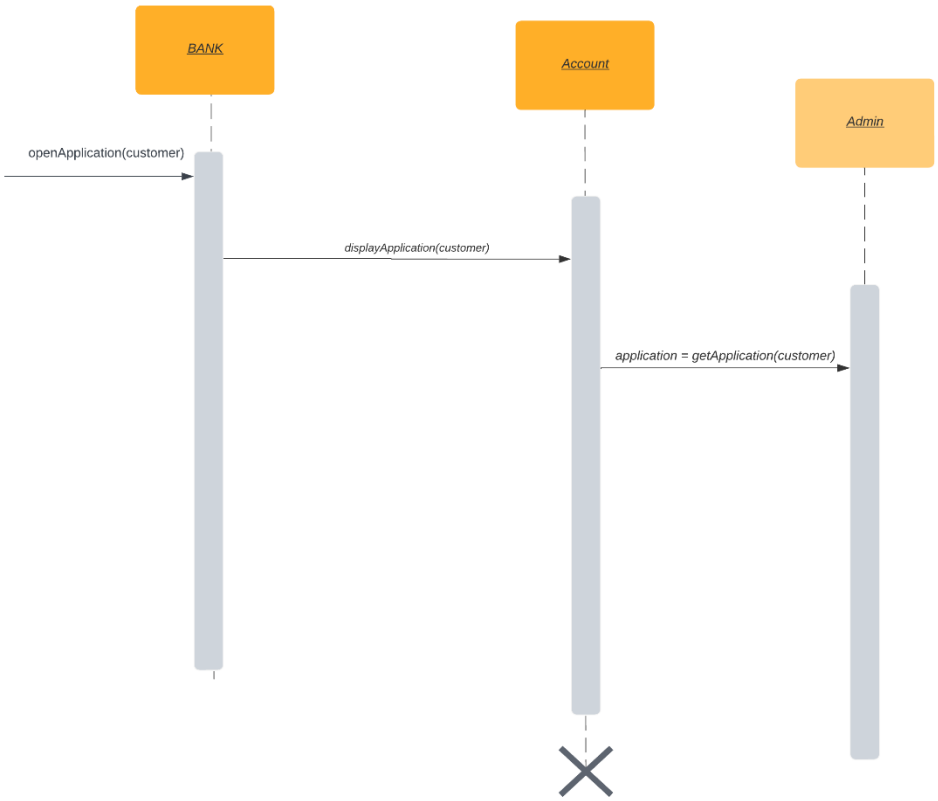

BANK
Account
Admin
Reports
generateReport(option)
GenerateReport(option)
GenerateReport(option)
report = create(option)

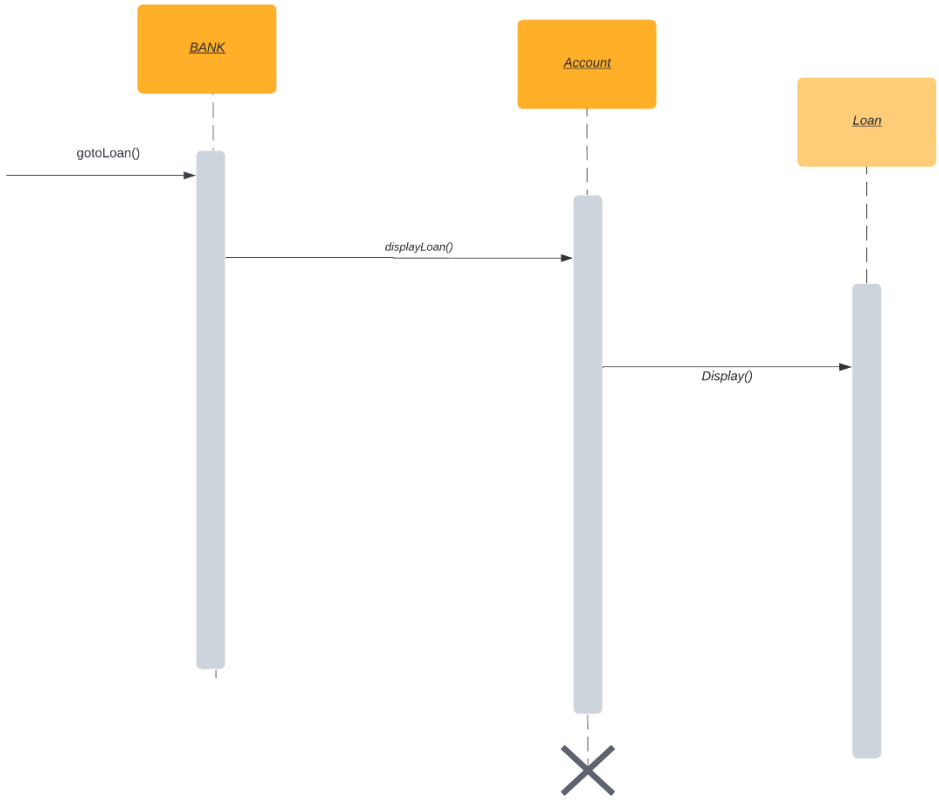

BANK
Account
Loan
loanApplication(amount
,rate ,duration)
applyforLoan(amount,rate,duration)
apply(amount,rate,duration)

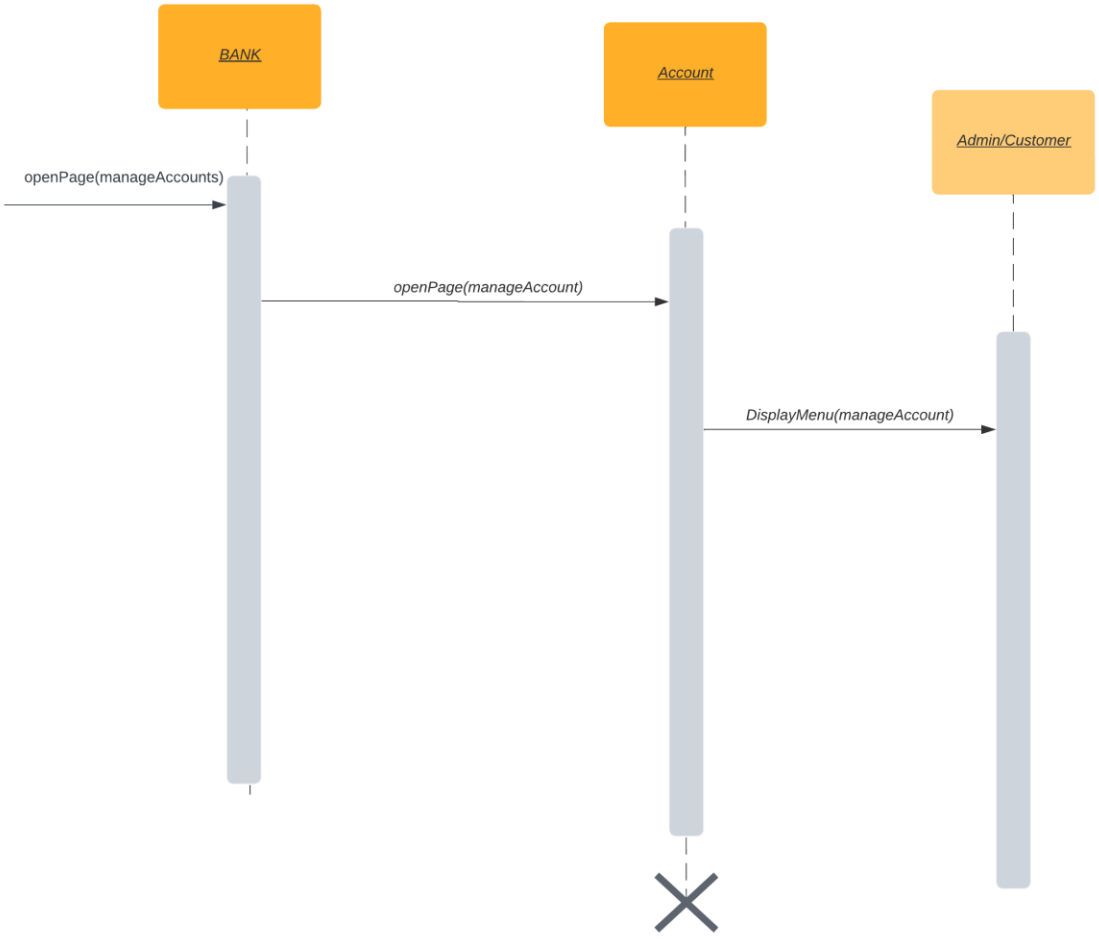

BANK
Account
Admin
loanDecision(accept, ammended record)
updateRecord(accept, ammended record)
updateApllicationTable(accept, ammendedR)

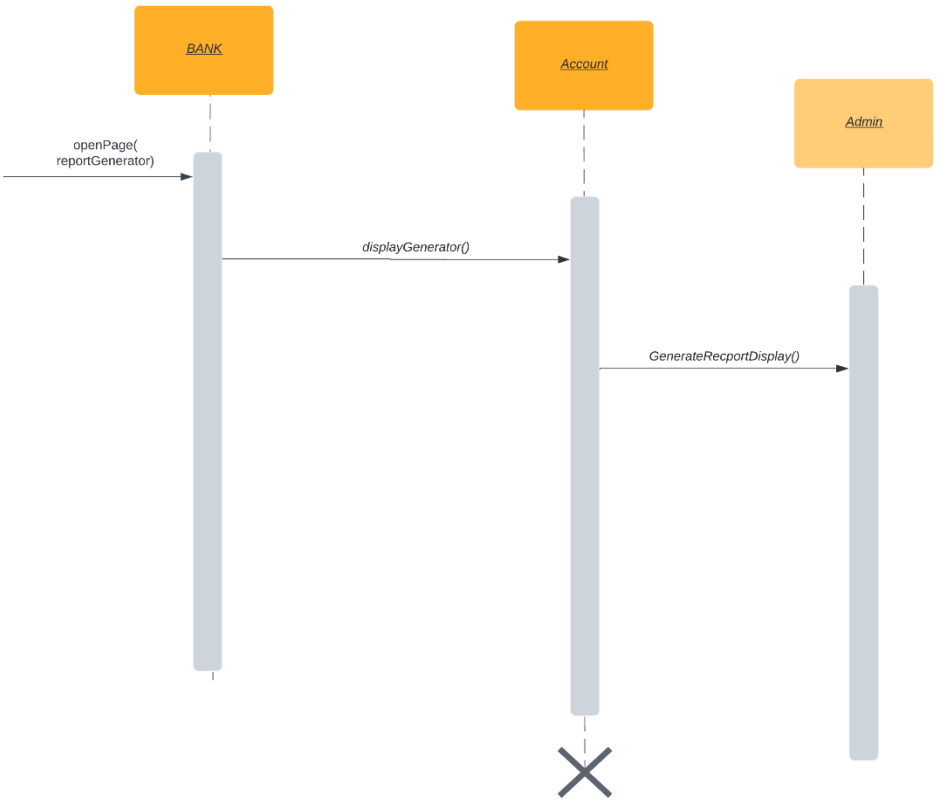

BANK
Account
Admin
manageRecords(
dbName)
ManageRecords(dbName)
ManageRecords(dbName)

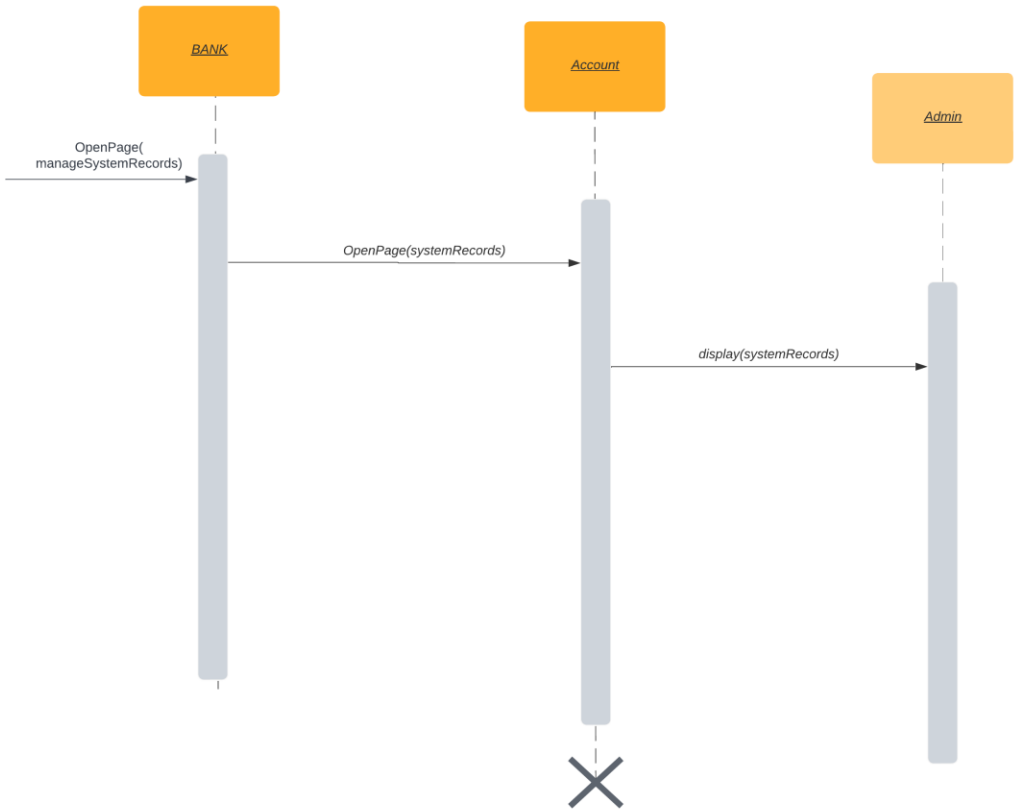

BANK
Account
Admin
openApplicationDatabase()
displayapplications()
displayApplicationsTable()

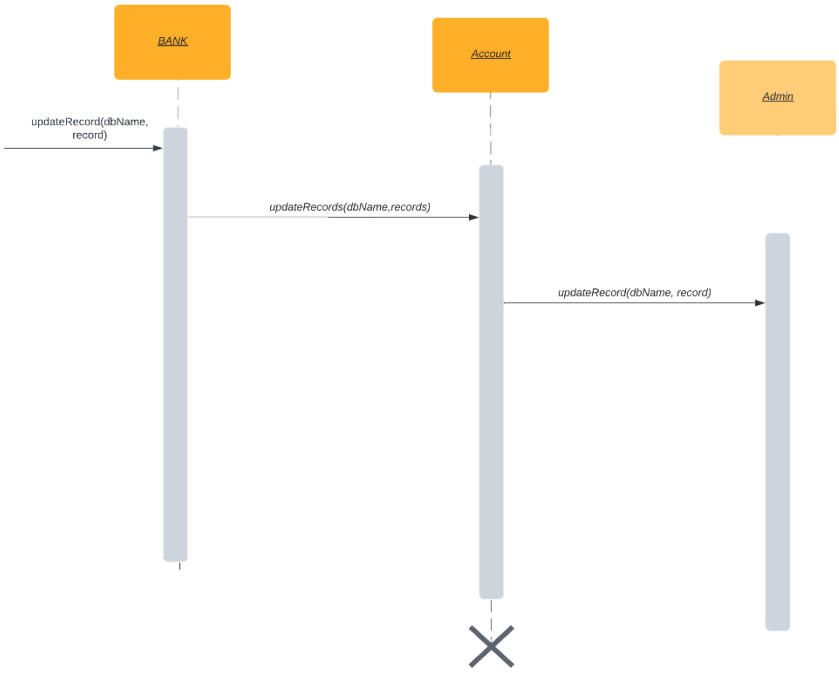

BANK
Account
Admin
openApplication(customer)
displayApplication(customer)
application = getApplication(customer)


BANK
Account
Loan
gotoLoan()
displayLoan()
Display()


BANK
Account
Admin/Customer
openPage(manageAccounts)
openPage(manageAccount)
DisplayMenu(manageAccount)


BANK
Account
Admin
openPage(
reportGenerator)
displayGenerator()
GenerateRecportDisplay()


BANK
Account
Admin
OpenPage(
manageSystemRecords)
OpenPage(systemRecords)
display(systemRecords)


BANK
Account
Admin
updateRecord(dbName, record)
updateRecords(dbName,records)
updateRecord(dbName, record)

7. Class Diagram