

CS 319 - Object-Oriented Software Engineering  
Analysis Report

E-Banking System

Group 2-11

Tuna Seyis

Kübra Şen

Kaan Kale

Table Of Contents

1. Introduction.......................................................................................3

2. Requirement Analysis.......................................................................3

2.1. Overview..............................................................................................3

2.2. Functional Requirements.....................................................................4

2.3. Non-Functional Requirements.............................................................4

2.4. Constraints...........................................................................................4

2.5. Scenarios..............................................................................................5

2.6. Use Case Models.................................................................................6

2.7. User Interface.......................................................................................9

3. Analysis..........................................................................................................10

3.1. Object Model.....................................................................................10

3.1.1. Domain Lexicon...................................................................10

3.1.2. Class Diagrams.....................................................................11

3.2. Dynamic Models...............................................................................12

3.2.1. State Chart............................................................................12

3.2.2. Sequence Diagrams..............................................................13

4. Conclusion......................................................................................................19

5.References........................................................................................................20

1. **Introduction**

E-Banking System is a basic software program which we decided to develop. The main purpose of this program is making banking operations easier, faster and more useful than other banking programs for our users.

In this program our customers can open saving account or checking account, apply to take loan and get reply from the server whether it accepted or denied, credit money, debit money, transfer money to other customers who also uses our bank and users can see their operations which they did before.

The program will work on desktop and internet connection mandatory. Users can control it by mouse and fill fields with their keyboard.

1. **Requirements**
   1. **Overview**

To use the program for first time the user must be sign up for our e-banking

system with their own information after this they encounter with login page and with their ID’s and password user can easily access to main menu. In this menu user will see six different choices. These are “Create Account”, “Take loan”, “Debit Money”, “Credit Money”, “Transfer Money”, “List Operations”. In first choice which is “Create Account” user can open a new “Saving Account” or “Checking Account”. When “Taking Loan” selected user enters amount of money and progress of their payments in number of months. In “Debit Money” and “Credit Money” customer can draw money from their accounts or put money to their accounts respectively. When “Transfer Money” button pressed user can send money to other customers however they also must use our program to succeed. Last button is “List Operations” and when this clicked users can see their last transactions which they did before.

**2.2.** **Functional Requirements**

1. Users will be able to create an account (saving/checking) via system.

2. Users will be able to credit money to their accounts and debit money from their accounts.

3. Users can transfer money from their accounts to other accounts.

4. Users can take a loan and choose its type with program.

5. System will make loan transactions automatically from indebted user’s account.

**2.3. Non-Functional Requirements**

1. User interface will be designed in order to be easy to use by all end users. So, minimalistic and simple design will be used. All operation controls should be simple and identifiable.

2. System should be secure. In order to provide that user passwords will be stored encrypted.

3. Program should be time-efficient so users can do most of the operations at most 3 clicks.

**2.4. Constraints**

1. Program will be implemented in C++

2. User interface will be implemented by WinAPI for C++

**2.5. Scenarios**

**1. Opening First Accounts**

After getting his first paycheck Ahmet wants to open a bank account. He goes in bank and tells bank employee that he wants to have a bank account. Employee opens an account for him, enters his information and adds given amount of money to his account. After couple of months when he got bonus from his company, he wants to keep his money from a saving account. Firstly he creates his saving account and pick an account type. After that he credits his bonus to his account and transfers it to a saving account.

**2. Getting Loan**

After 2 years Ahmet get bored from using public transport and wants to buy a car. But his money is not enough for it. So he opens his account and goes to loans segment. He enters needed amount and payment time. After he got confirmation from bank he withdraws his money and buy a new car. During payment time his debt will transacted directly from his checking account.

**3. Sending money**

When Ahmet able to support himself completely with his income he wants to support his family and wants to send money to his home. He calls his parents and asks their account number. After receiving their number Ahmet opens the program and goes to transfer segment. He enters his parents account number and transfer amount and clicks the confirmation button.

**4. Paying loans early**

After his 5th year at the company Ahmet receives a promotion and thanks to his efforts he get huge bonus from his company. With that money Ahmet wants to pay his loan early. He opens the program, goes to loan segment and chooses pay option. In new segment he enter remaining amount of the debt to text field and hits the confirmation button. Amount will be deducted from his checking account.

**2.6. Use Case Models**

This section provides information about the main use case model of Banking System. Main actor is Customer. Customer can sign up the system then can log in. After that he/she see the main menu and can choose the one of the options which are in the use case diagram. After the all these steps, operation checking step in and inform the customer about whether operation is done or not.

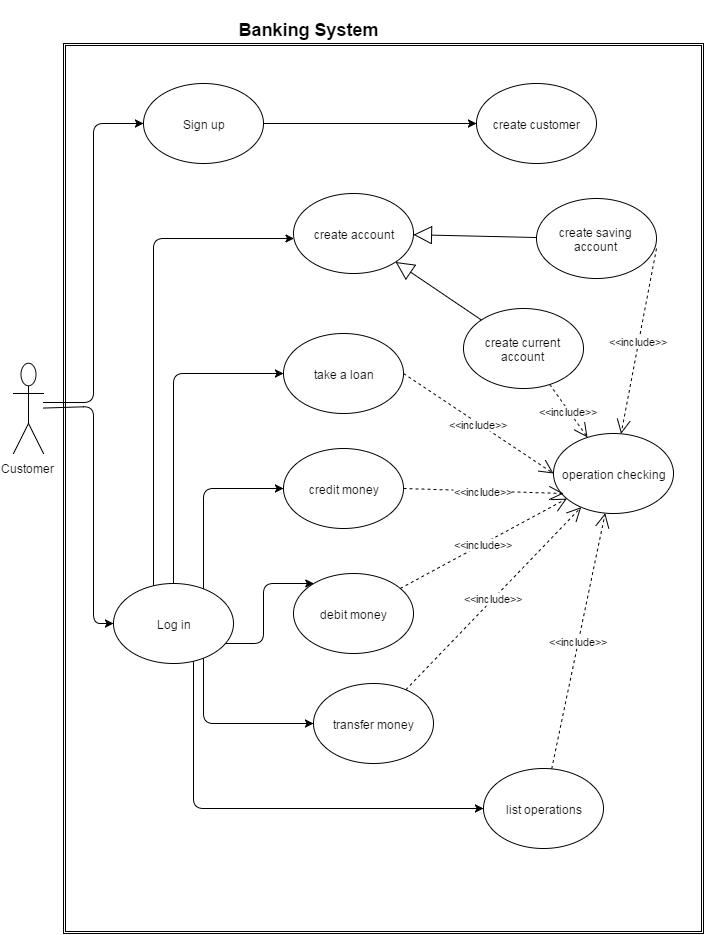


Figure 1 use case diagram

Some detailed use case explanations:

**Use Case Name: Sign up**

**Primary Actor**: Customer  
**Stakeholders and Interests:**  
-Customer wants to sign up the banking System  
-System ask the information about customer  
-System wants customer to determine a password  
  
**Pre-conditions:** -  
**Post-condition: -**  
  
**Entry Condition:** Customer selects “Sign up” from Start menu.  
**Exit Condition:** Customer selects “OK” and return Start menu.

**Success Scenario Event Flow:**  
  
1.System accept the customer.  
  
**Alternative Flows:**  
A. If customer desires to return start menu at any time:  
 A.1. Customer selects “Return to Start Menu” button to return start menu.  
 A.2. System displays start Menu.

B. System does not accept the customer:

B.1. System report that there is another person with the same phone number  
 B.2. Customer selects another phone number

**Use Case Name: Log in**  
  
**Primary Actor**: Customer  
**Stakeholders and Interests:**  
-Customer wants to log in the Banking System  
-System ask user information to the customer  
  
**Pre-conditions:** Customer registered before  
**Post-condition: -**  
  
**Entry Condition:** Customer selects “Log in” from Start Menu.  
**Exit Condition:** Customer selects “Back” and return Start Menu.

**Success Scenario Event Flow:**  
1.Customer Log in the System and transferred to the main menu  
  
**Alternative Flows:**  
A. If customer desires to return start menu at any time:  
 A.1. Customer selects “Return to Start Menu” button to return Start menu.  
 A.2. System displays Start Menu.

B. if the log in information is wrong

B.1. System report that there is wrong information  
 B.2.a Customer write the right information  
 B.2.b Customer return the start menu

**Use Case Name: Transfer Money**  
  
**Primary Actor**: Customer  
**Stakeholders and Interests:**  
-Customer choose the transfer money option in the main menu  
-System ask which account customer wants to transfer from  
- System ask which account customer wants to transfer to  
- System ask the amount of money which customer wants to transfer  
  
  
**Pre-conditions:** account that customer wants to transfer from has the determined amount of money  
**Post-condition: -**  
  
**Entry Condition:** Customer selects “Transfer Money” from main menu.  
**Exit Condition: -**Customer selects “back” and return main menu.  
 -Money transferred and successful page turned over the main menu

**Success Scenario Event Flow:**  
  
1.System transfer the money  
  
**Alternative Flows:**  
A. If customer desires to return main menu at any time:  
 A.1. Customer selects “Back” button to return main menu.  
 A.2. System displays main Menu.

Account dos not have the amount of money:

B.1. System report that the money is not enough   
B.2.a Customer selects another account  
B.2.b Customer return to the main menu

**2.7. User Interface**

User firstly sees the start menu. This includes sign up and log in option. After the log in, there is a main menu consist of the option list ; Take Loan, Create Account, Credit Money, Debit Money, Transfer Money, List operation and return.

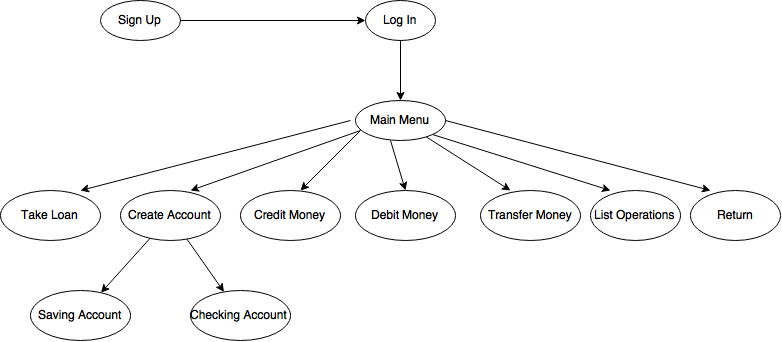
****

Figure 2 : User interface

**3. Analysis**

**3.1. Object Model**

**3.1.1. Domain Lexicon**

We mingle with some banking term. Explanation of the term which we use in the project is below

Loan: taking the money from bank for certain time. In this operation bank transfer money to customer’s account and provide them payment plan.

Credit Money: putting the money to a certain account. All customers can credit money to their own account

Debit Money: taking money from a certain account. All customers can take money from their own account

Transfer Money: transferring the money from a one account to another. All customer can transfer money from their own account to any other account

Saving Account: accounts maintained by retail financial institutions that pay interest but cannot be used directly as money in the narrow sense of a medium of exchange.

Checking Account : A transactional deposit account held at a financial institution that allows for withdrawals and deposits

**3.1.2. Class Diagram**    
 There are 7 classes in the Project. User Menu class is to organize main menu and start menu. Bank class is the main class in the code. İt include main function of the bank system. Account class is parent class of saving account and current account and has a relation with customer class.

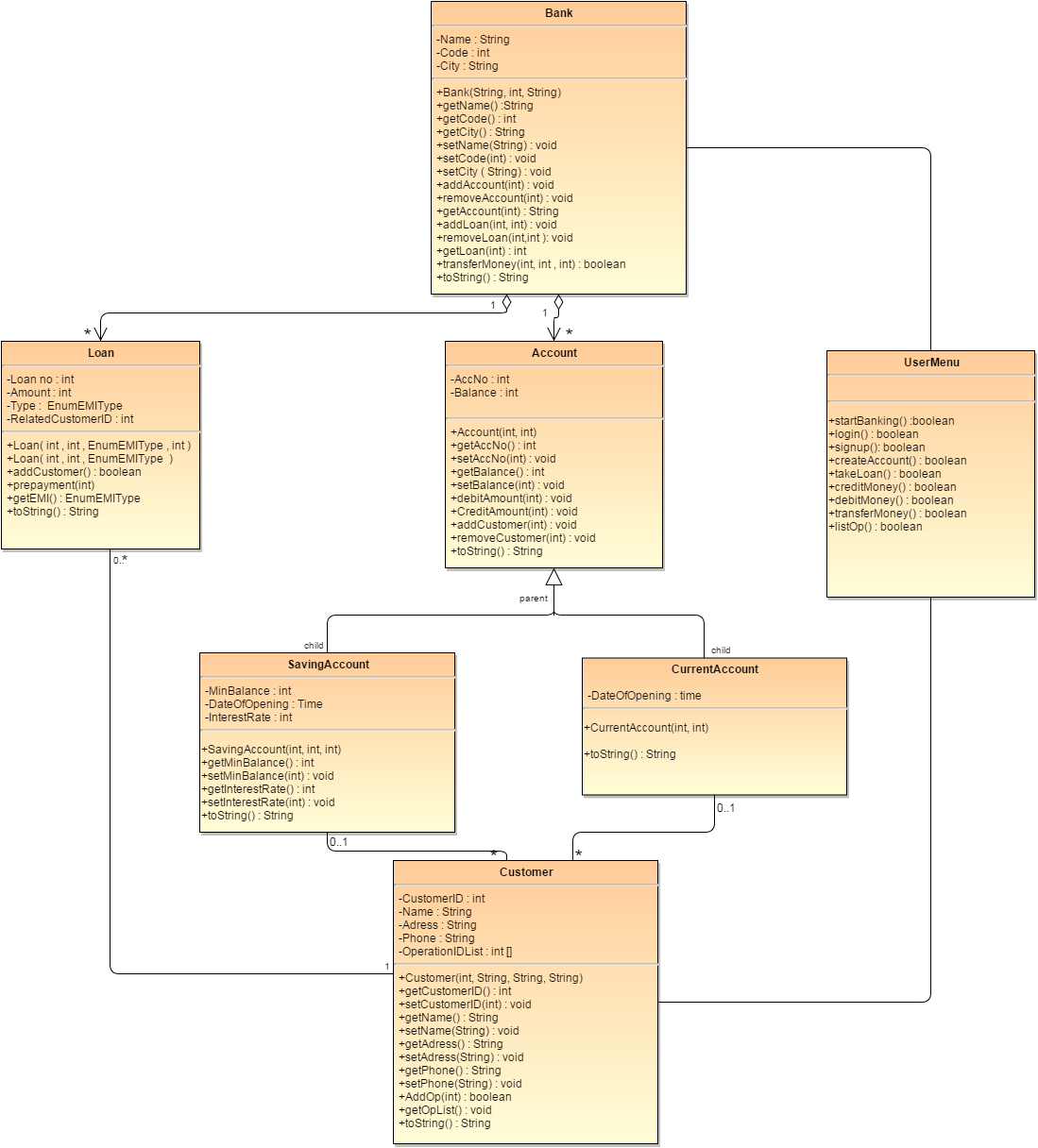
****

Figure 3 : Class diagram

**3.2.1. State Chart**

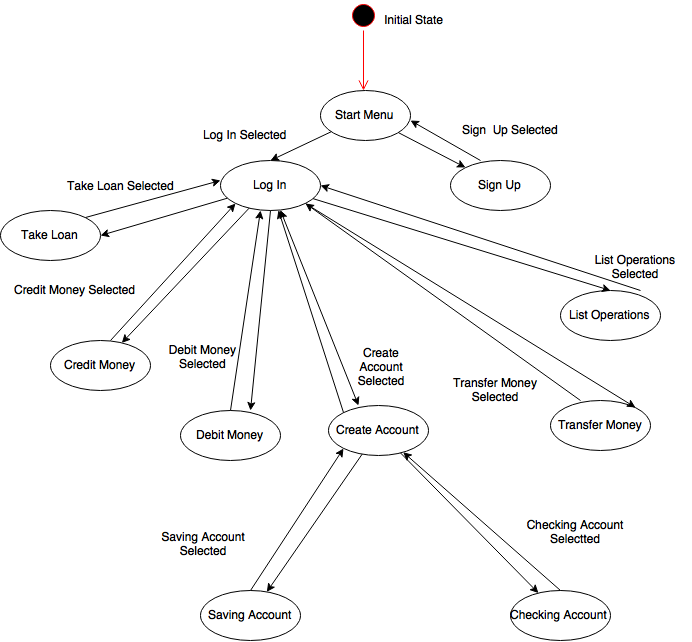


Figure 4 : State Chart Diagram

**3.2.2 Sequence Diagrams**

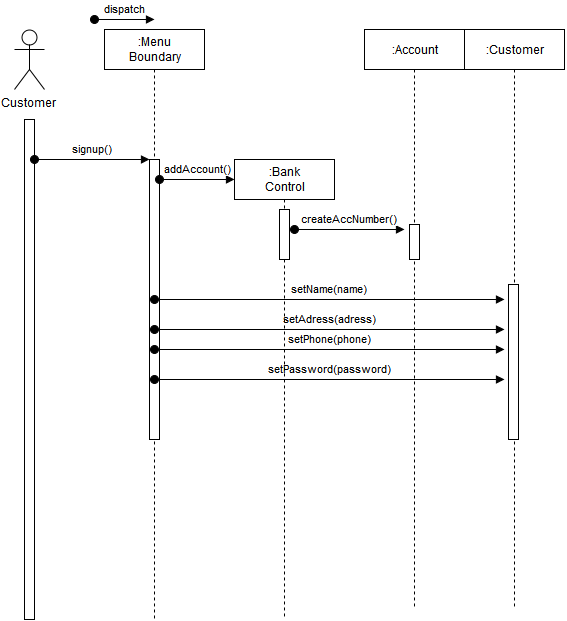


Figure 5 : Sequence Diagram 1

When customer chooses creating new membership to the bank, bank automatically creates account number for the customer and asks for customer name, address, phone and password. Program stores the information, encrypts the password and returns a confirmation message to user.

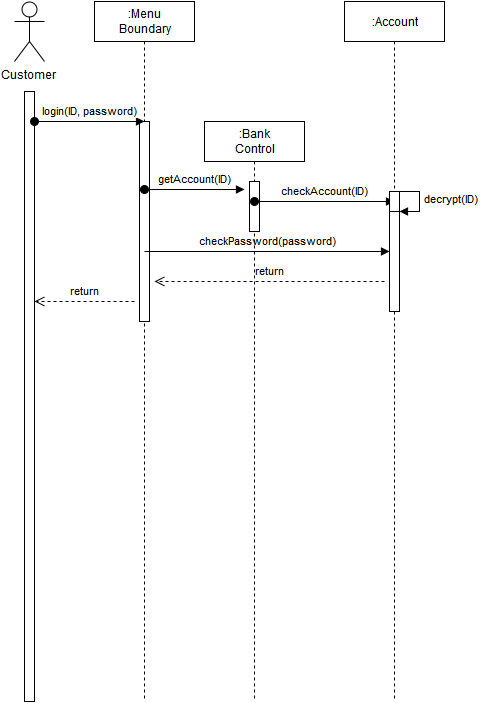


Figure 6 : Sequence Diagram 2

Customer enters ID and password through the menu and control unit checks ID and password matches with password in storage and if passwords checks with each other, main menu will show up or an error message will show up.

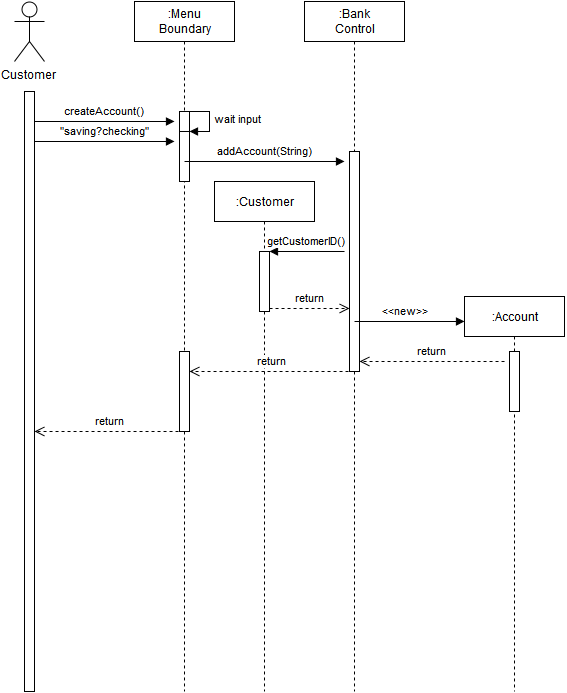


Figure 7 : Sequence Diagram 3

When a customer choose create an account option, menu will ask to user type of the account. After that according the type of the account, program will create new account object with the selected type and bind that account to user’s account ID. New account initially created empty.

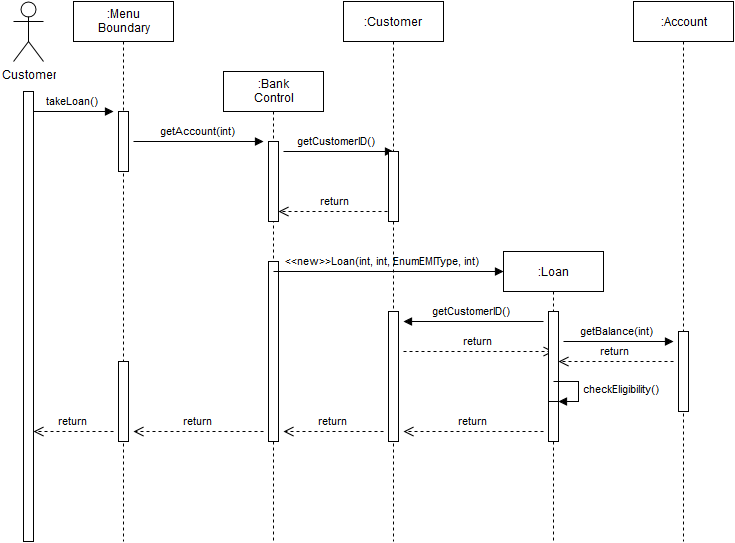


Figure 8 Sequence Diagram 4

When a customer wants to take a loan from the bank, control unit will find account number of the user, then creates a new loan object and ask user loan amount and amount of installments. After user entered desired amount of money and installment program calculates total debt and check eligibility user according to the customer account.

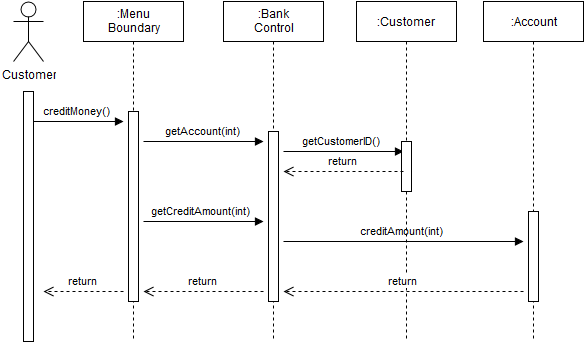


Figure 9 : Sequence Diagram 5

When users choose credit money to own account, control unit gets the customer’s account then adds the entered amount.

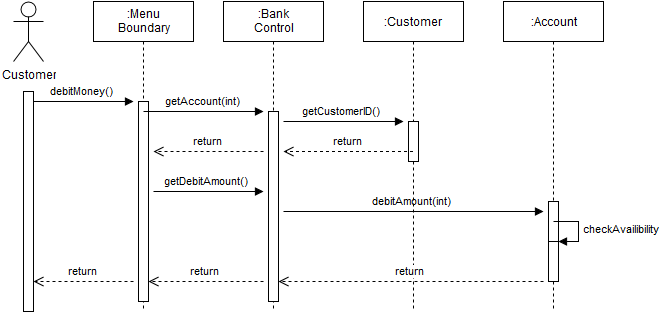


Figure 10 : Sequence Diagram 6

When users choose debit money, program brings the account of the customer and checks availability of entered money. If money is available in the account money will be given.

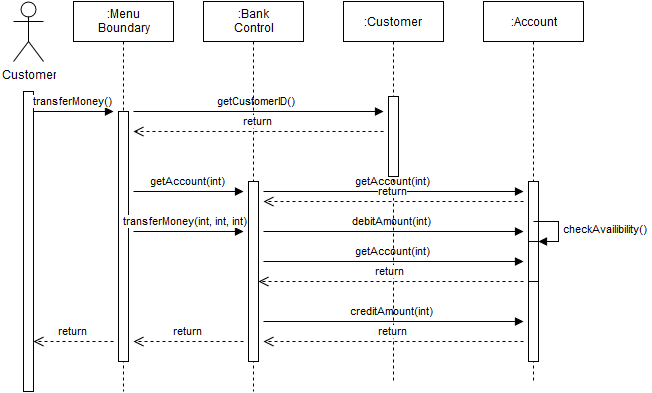


Figure 11 : Sequence Diagram 7

If user wants to transfer money to another account, control unit bring the account of the customer and debits the entered amount from user’s account and credits the same amount of money to desired account.

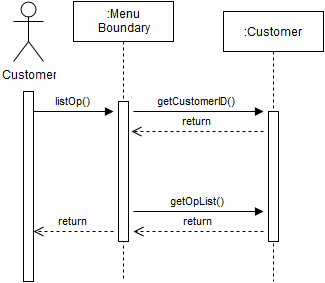


Figure 12 : Sequence Diagram 8

When an user wants to see previous list of operations, control unit brings up the previous operations list.

**4.Conclusion**

In this project, we did our analysis report to design an e-Banking System. Firstly we follow the requirements. In our project design we implement our program with respect to this requirement specification part. We worked on functional and non-functional requirements, constraints. We always tried to do our project design simple and useful for our users to handle easier. We thought scenarios which our customers can encounter and work on those scenarios to settle our bugs and errors.

As we mentioned before this is a desktop application which works with internet and it is a simple program because of after the sign up action users can easily log in with their passwords and use whenever they want. Users can create new saving and checking accounts to save their money securely. If they have accounts and money in it they can credit money or transfer money to other users. They can also debit money to their account. If the customer needs money they can apply loan and select amount of money they want and get reply from our system in short time interval. When customers select “List Operations” choice they can see their last transactions.

Then, we started to work on models with considering our requirements. We did use-case models for sign up, log in and transfer money options and in these parts we always thought best for our customers and their interests. We consider pre-post conditions and entry-exit conditions. Then we worked on user interface and state chart. We made distinct and explanatory models in these parts to find every option while customer uses this program. In class diagram we generate all classes. Under these classes we state all of the methods, functions and parameters they include. Moreover, eight different sequence diagrams take place in this project analysis report.

As a result, we attempted to make a perfect project analysis report to pull off our project in upcoming parts.

**5.References**

Draw.io,. 'Flow Chart Maker & Online Diagram Software'. N.p., 2015. Web. 28 Oct. 2015.