TED ÜNİVERSİTESİ

Software Requirements Specification

CMPE 313 – Software Engineering

Project Name: Online Banking System

Instructor: Haydar CUKURTEPE

Students Information:

Ahmet Berkay SARIASLAN – 54490639066

Sümeyra Altaş - 43207666532

Serhat Yokuşoğlu – 14113777438

Gizem Kurtcuoğlu-59719004892

Table of Contents

1. Introduction 4
   1.1 Purpose 4
   1.2 Document Conventions 4
   1.3 Intended Audience and Reading Suggestions 4
   1.4 Product Scope 4
   1.5 References 5
2. Overall Description 5
   2.1 Product Perspective 5
   2.2 Product Functions 5
   2.3 User Classes and Characteristics 6
   2.4 Operating Environment 6
   2.5 Design and Implementation Constraints 6
   2.6 User Documentation 7
   2.7 Assumptions and Dependencies 7
3. External Interface Requirements 7
   3.1 User Interfaces 7
   3.2 Hardware Interfaces 8
   3.3 Software Interfaces 8
   3.4 Communications Interfaces 9
4. System Features 9
   4.1 Advantages Based on the Customer Profile 10
      4.1.1 Description and Priority 10
      4.1.2 Stimulus/Response Sequences 10
      4.1.3 Functional Requirements 10
   4.2 Keep Track of Customer's Past Expenses 10
      4.2.1 Description and Priority 10
      4.2.2 Stimulus/Response Sequences 10
      4.2.3 Functional Requirements 10
   4.3 Checking Permission 11
      4.3.1 Description and Priority 11
      4.3.2 Stimulus/Response Sequences 11
      4.3.3 Functional Requirements 11
   4.4 Satisfaction Survey 11
      4.4.1 Description and Priority 11
      4.4.2 Stimulus/Response Sequences 11

4.4.3 Functional Requirements....11
5. Other Nonfunctional Requirements....11
5.1 Performance Requirements....11
5.3 Security Requirements....12
5.4 Software Quality Attributes....12
5.5 Business Rules....13
6. Other Requirements....14
Appendix A: Glossary....14
Appendix B: Analysis Models....15
1. Customer Sequential Diagram....15
2. System Sequential Diagram....15
3. Customer Registration Sequential Diagram....17
4. Login Activity Diagram....18
5. Customer Sequence Diagram....19
6. Money Transfer Activity Diagram....19
7. System Management Sequence Diagram....20
8. Advantage and Campaign Select Sequence Diagram....20
9. Online Banking Use Case Diagram....21
10. Preparation of Campaigns/Advantages Use Case Diagram....22
11. Show Campaigns/Advantages Use Case Diagram....23
Appendix C: To Be Determined List....23

1. Introduction

1.1 Purpose

This document details the software requirements for the Online Banking System. New Online Banking System software needs to be built. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external events. Users can be diversified as standard, retired and student. Bankers can manage users easily. It will be a more efficient and easier way to have a record on systems through which everyone can easily access it according to his rights as compared to the traditional banking system. Every bank will prefer the online banking system instead of the traditional banking system as it contains many useful features and the fastest methods for transactions

1.2 Document Conventions

Interface models in part 3.1 and this the two main screen of the program. The interface has several buttons and every button have their own unique task to do.

Moreover, there are some graphics which are sequential and activity diagrams. These explains how our system works on the customer's and bankers' computers separately.

- HTTP: Hyper Text Transfer Protocol.
- HTML: Hyper Text Markup Language.
- TCP/IP: Transmission Control Protocol/Internet Protocol.
- FTP: File Transfer Protocol

1.3 Intended Audience and Reading Suggestions

There are different kinds of readers. These are

a. Customers
   - Project scope

b. Employers

c. Developers
   - Use Case Diagrams
   - Sequence Diagrams
   - Activity Diagrams

d. Project Manager
   - System Features
   - Hardware Requirement
   - Software Requirement

1.4 Product Scope

The aim of our project is to create a healthier and more oriented software between bank customers and bank employees. Our goal is to create a customer-friendly software for different types of users in the

long term. We think that it is a very important software for bank employees to offer customers more efficient offers. The first of the functional requirements is the client's profile. The customer profile is important in our software in terms of tracking the bank customer. With this information, the purpose of using the customer's bank is determined. Then, according to the relationship with the bank. For example, according to the loan debt, certain payment transactions on certain dates are reminded of the bank user by the software. In addition to these, other non-functional functions are also considered to be part of the software. For example, voice recording system, enter and exit dates to the bank branches.

1.5 References

The following is an enumerated list of sources that we have been inspired by in this project.

1. Software Engineering, Ian Sommerville, 10th Edition(https://mycourses.aalto.fi/pluginfile.php/1177979/mod_resource/content/1/Sommerville-Software-Engineering-10ed.pdf)
2. Software Engineering, A Practioner's Approach(https://cloudflare-ipfs.com/ipfs/bafykbzaced4qcp5uckxseu25auim26267yz4mhg3nfsgnbr4crav6rnbz4ox6?filename=Roger%20Pressman%2C%20Bruce%20Maxim%20-%20Software%20Engineering_%20A%20Practitioner%27s%20Approach-McGraw-Hill%20Education%20%282019%29.pdf)
3. Extreme Programming Technique(http://www.extremeprogramming.org)
4. The Model View Controller Architecture Pattern(https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/)
5. https://cupdf.com/document/online-banking-srs.html

2. Overall Description

2.1 Product Perspective

Out program working on the existing banking software. We just simply create the new reachable and easily accessible program. Also, there are some advantages for the customers if the bank uses our software. Moreover, bankers can access the customer' information very easily through our software. Also, our software has own database but it has to cooperate with banker's database since customers some special information has to be kept in secret so we cannot reach them with our database. It is against the personal data protection law.

2.2 Product Functions

The Online Banking System consist different modules. These are

1. Login

This module allows to login into the system. Customer logins by entering Account ID.

2. Transfer Funds

The customer can easily transfer funds to other accounts.

3. Withdrawing the loan

The customer can withdraw the loan.

4. Balance check

The customer will be able to check the balance.

5. Customer Specialty

Type of customer is selected in this module

6. View Account

The customer can view account information (balance, transaction history etc.).

2.3 User Classes and Characteristics

There are various users that will be able to use the system. User types are divided into bankers, retired, students and standard users.

For Bankers, can look at the user's accounts in the Bank. Also, they can control the money transfers, investments of the user, debts and credit tracking. If there is needed, our program can find appropriate bank for the users but all these functions system is under the banker's control. Therefore, they will use the system daily and speed and stability are important.

For Retired Users, can be said older customers. They might not know anything about computers for this reason the system has been made much easier and simple to use. Also, they can use the transformation system in their cities freely.

For Student Users, the system offers different advantages. They have discounts for shopping and some conversion tools

For Standard Users, system operates normally. They will do transactions as quickly and efficiently.

2.4 Operating Environment

Our software works on computers and using the latest version of Microsoft operating system, which is Windows 11, and this software works on the already prepared banking software. Also, this software based on java language, and we will use "postgerSQL" for database.

2.5 Design and Implementation Constraints

Our program works on the bankers existing software programs. But it is running on the servers so it should be handled large amount of traffic. It has to be so fast and in balance. So, good hardware must be purchased. Also, there will be so much information it leads to overload the storage areas such as HDD

and SSD storage areas. Also, some information cannot be acceptable for our software such as person's id card and password. Since it has to be given by the banks so separation the database necessity.

Moreover, the software works on the Microsoft 11 so the all the computer versions must be the same. If it's different, it may cause a version problem for both customers and bankers. After the submission of the work, all the maintaining of the software is under the control of our customer.

2.6 User Documentation

We have to kind of users such as bankers and customers. So, there are two manuals for this. First of all, bankers can manage the customer's accounts. There are some buttons to see the customers identity, bank information such as loans. For the customers, there are some buttons too. They can pay their loans or see their special account information and there is one help button to the advantages of their special accounts.

Customers have to enter their password in order to use the system and also bankers use the specific id to reach the wanted customer's account. This id given by the bankers, so we cannot interfere with it.

2.7 Assumptions and Dependencies

Some factors could affect the requirements and the system depends on several factors in the SRS. While working with the bank, the policies and decisions decided by the bank may change over time. Depending on this change, the system may also show changes within itself. In addition, we are specific to PostgreSQL Database. This is constraint relating to the operating environment. When the operating system is renewed operation of the system cannot be guaranteed unless is not fully compatible.

On the other hand, power source, servers, communication mediums(wired/wireless), internet connection are the factors on which the system depends.

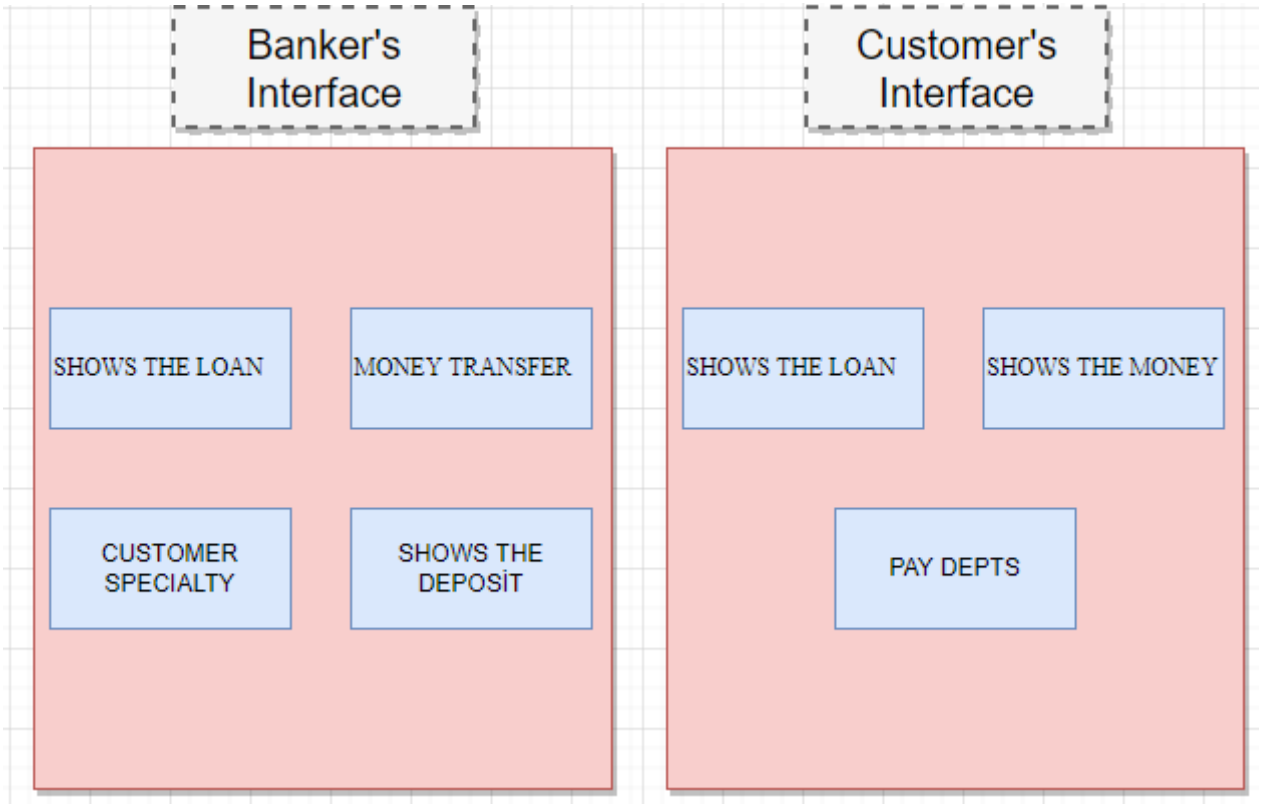
3. External Interface Requirements

3.1 User Interfaces

With our computer software, when bankers open our software, they will see the screen which has 4 buttons. Each button has own duty. If they click the first button, they see the customers loans. Second button shows the customers money transfers. Third button shows the customers specialty that are "student", "retired", "standard". Fourth button shows the customers information such as how many monies they have. If customer try to withdraw more money than they have, the error message appears on the screen and then the bankers can inform the customer in a proper way.

Also, our software includes a customer interface. There is a screen and also there are 3 buttons that they can click. But before that, when the customers try to open their account. They have to log in. They have to fill their passwords. If they cannot fill their passwords properly, the error message appears and it

says, “wrong password, please try again”. After they correctly enter their password, they will see the main screen and there are buttons. First button shows the money that they have. Second button shows the loans and also there is a button for paying their debts. If the account has not enough money for paying their loan, there is an error message that it says, “not enough money to pay”. Third button shows the money transfer to another account. There is another button to transfer the money to another account.



3.2 Hardware Interfaces

There are different hardware components. These are

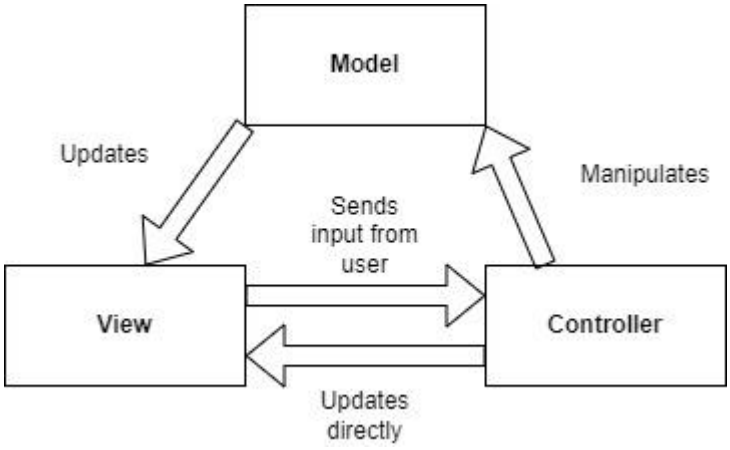
- Touch screen/Monitor
- Computer (OS, Windows)
- Smart mobile device

This software application works actively on computers. However, the customer can also see their own profile on their mobile device. Bank employees can access and use this application from computers within the bank.

3.3 Software Interfaces

This software product contains MVC architectural pattern steps. We need data abstraction because a lot of data will be presented to the user, i.e. the customer. Thus, the data and the user interface are analyzed by data representation with the "controller" component without affecting each other. The

interface must be able to connect to a database. This database model should be kept in the system integrated with the bank. A programming language should be used for processing and displaying data, and a markup language should be used for displaying them. The Java programming language should be used behind the software application. But this complex arrangement should not be reflected in the user. The operations determined by this programming language must be relative to American Standard Code for Information Interchange (ASCII).



On the other hand, it is important for our software product to be sustainable to be used regardless of the operating system. Messages entering the system are usually customer messages, while messages exiting the system should be returned to the customer as a response or feedback. In this system, the nature of communication is to be understood dec the bank employee and the bank customer without the need for face-to-face contact. That is why it is important that the customer returns to the system.

3.4 Communications Interfaces

System needs to communicate with different sessions. So that, the various communication interface requirements that are needed in order to successfully system. Communication protocol will be TCP/IP. Also, for data sharing should be File Transfer Protocol (FTP) OR HTTP. Server shall communicate with customer and customer information stored in database. Also, if the customer cannot fill the password section correctly return the error massage. Also, bankers have their own communication between servers and database. Bankers directly communicate with database that's why they can reach all the information about customers. But more importantly there is a panel that customers and bankers can see differently and there are some buttons to reach desired information.

4. System Features

This section of the SRS describes the requirements for the system's features.

4.1 Advantages Based on the Customer Profile

4.1.1 Description and Priority

This feature in our software program is a high priority feature. Because the main purpose of this software application is to provide the right service to the customer according to the customer profile. The services that a customer in the retirement profile will use are directly different from the services that a customer in the student profile will use. This software application is a high priority feature so it will be used for the purposes of using this difference.

4.1.2 Stimulus/Response Sequences

After a valid login is made from the application screen, the third button is used to view the benefits according to the customers' profiles.

When you click on the button, the opportunities that you will come across will be opened in the form of a menu.

It is possible to access more subcategory menus according to the need from the menus. If you are in the student client class, it will be easier for you to access the category of cultural activities such as concerts. Because the system has registered you as a student and will put the advantages in front of you accordingly.

After the selection of advantages or opportunities is made, you will be redirected to the corresponding page. For this operation, it is possible to send it directly to the "click and go" or the application download menu for smartphones.

4.1.3 Functional Requirements

The user sees the offered advantage. after you have seen it, you can open the details section for details. After clicking on the "apply" button, you will be redirected to the corresponding site of the corresponding campaign. The main procedure here is to offer an advantage over the customer profile.

4.2 Keep Track of Customer’s Past Expenses

4.2.1 Description and Priority

This feature's purpose is to keep track of the customer's past expenses. The customer can observe his past expenditures whenever he/she wants, and at the same time, the bank employee can be aware of the past expenditures of the customers. It has priority for our system, but it is not high priority.

4.2.2 Stimulus/Response Sequences

When the customer clicks to the shows the money button, all the past expenses and the current money appears on the screen. Also, bankers can see them with click to the shows the deposit button.

4.2.3 Functional Requirements

If there is no money the error massage appears on the customers and also the bankers screen and it says that “There is no money and past expenses”.

4.3 Checking Permission

4.3.1 Description and Priority

Checking the permission of the customer and the bankers both are in high priority since there are several laws and regulations. When customer wants to register the system, he/she must accept law on the use of personal data. Also, bankers cannot share the customers personal information anywhere else since the law so strict and punishment will be so high if they do that.

4.3.2 Stimulus/Response Sequences

When the customer wants to register the system. There is one button to accept the rules. These rules defined by the banks so as a developer we cannot interfere it. Customer has to accept it and then register will be successfully done.

4.3.3 Functional Requirements

If the customer did not accept the rules. The registration failed and the error massage appears on the customer's screen.

4.4 Satisfaction Survey

4.4.1 Description and Priority

Our system request satisfaction survey from customers. The main goal is improved the system according to feedback. There is no priority for this feature. It is a feature left to the customer's request.

4.4.2 Stimulus/Response Sequences

It is a feature that comes out at the end of all operations.

4.4.3 Functional Requirements

The system will give a warning message when the user provides an incomplete entry. The start and end of the entry to the survey are indicated

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Since it is a large-scale system, this system should use less memory and be easily accessible. Memory management should be wisely for system performance. The transactions that the customers will make in the system should be done in a short time. The system should run smoothly until it is turned off. Also, the system must have 5.2 Safety Requirements

This system opens the internet. All the works done in the online environment, or the bankers can use this system on their computers. So, people who wants to destroy the system can exploit the system if they crash the security system. So, our program does not contain the customer's private information such as ID card numbers, passwords. All these information comes from the bank's database that's why people cannot reach this kind of information. However, we have some process in order to keep the software safe.

-Our database backup itself every week. So, people's last processes kept in another database safely and if needed we can reach immediately to this database.

-If the customers try to identify themselves as another person, program immediately report the bankers.

-If there is an attack to our software, bankers reported, and they can shout the system down or counterattack the attackers.

5.3 Security Requirements

Security of customers' account and information is the highest information in banking system. So that, we used the latest technology security tools. There are different security features. These are Account ID and Password (PIN) Protection, resetting all information when exiting the system and encryption.

For Account ID and Password (PIN) Protection, users need to enter an ID and password. If the user enters this information incorrectly, login to the system is denied. In order to increase security, the user is expected to enter special characters (!'^+%&/()= etc.), uppercase and lowercase letters when setting the password. In addition, it requests to change the password at certain intervals.

For resetting all information when exiting the system, after a customer logs out of the system, the actions performed by the customer on the system are reset so that security is ensured as much as possible.

For Encryption, increase the security of passwords by using data Encryption. It converts these passwords into unintelligible form.

5.4 Software Quality Attributes

The Quality of the System is maintained in such a way so that it can be very user friendly to all the users. The software quality attributes are assumed as under:

a. Reliability

b. Maintainability

c. Usability

d. Efficiency

e. Flexibility

Reliability:

The failure rate in the online banking system should be least or negligible as the system is supposed to be reliable. Reliability of the system depends on the failure free transactions and how fast the system is able to recover from the failure. Measure if product is reliable enough to sustain in any condition. It should give consistently correct results. Product reliability is measured in terms of working of project under different working environment and different conditions.

Maintainability:

Different versions of the product should be easy to maintain. For development, it should be easy to upgrade for new features and new technologies time to time. Maintenance should be cost effective and easy. System be easy to maintain and correcting defects or making a change in the software.

Usability:

As online banking is carried by various types of clients i.e. whether they have knowledge of computers or not so the application designed for online baking must be easy to use and enable the client to manage their accounts or transactions with simplicity. The interfaces of the system ought to be clear, easy and simple to use and understand This can be measured in terms of ease of use. Online Banking System should be user friendly. It should be easy to learn.

Efficiency:

Measured in terms of time required to complete any task given to the system. For give an example is that system should utilize processor capacity, disk space and memory efficiently. If system is using all the available resources, then user will get degraded performance failing the system for efficiency. If system is not efficient then it cannot be used in real time applications.

Flexibility:

System should be flexible enough to modify. Adaptable to other products with which it needs interaction and should be easy to interface with other standard third-party party components. The third-party software component market thrives because in such a way many programmers believe that **component-oriented development** improves the efficiency and the quality of developing custom applications.

5.5 Business Rules

For The Administrator, regarding all kinds of products and services they offer to their customers, they transfer accurate, complete and timely information at every stage of their relationship and in every subject, by complying with the limitations set forth in the legislation. Except for the persons and authorities authorized to request information and documents, they are obliged to keep all kinds of information and documents regarding their customers confidential and diligently. They offer the same quality and the same level of service to all their customers. Also, they inform their employees in order to correct the erroneous practices that cause complaints and to prevent their recurrence.

For The staff, they establish a system that will answer all kinds of questions of their customers arising from the services provided and inform their customers about this service. By investigating the causes of customer complaints, they take the necessary measures to prevent the repetition of justified complaints.

The enrollment system shall include a mandatory customer ID input on the on the main page. All customers need to enter password correctly. If the customer successfully login the system, then system allow customer to access the system.

6. Other Requirements

When implementing this software, we need some requirements for the database. These requirements are the customer's personal information, services to the bank, such as the date of becoming a customer. In addition to these, both the identity information and the business information of each bank employee are stored in the database.

In order to be used in the sense of a global brand, it must be in the use of a universal language. In other words, a user who is not technology literate and a technologist should also be able to use this software.

The reuse targets for the project can be summarized as the following. While campaigns are constantly updated in the project, some information that the application keeps statically is available again. In other words, even if the customer leaves the bank, the information that was stored can be used here again when the customer is in the same bank again.

One of the most important shares in legal processes is the protection of personal data. In this software project, the application should be used by receiving the text of mutual consent. Because user-specific information, such as personal information and interests, is required.

Appendix A: Glossary

PIN: Personal Identification Number

FTP: File Transfer Protocol

HTTP: Hyper Text Transfer Protocol.

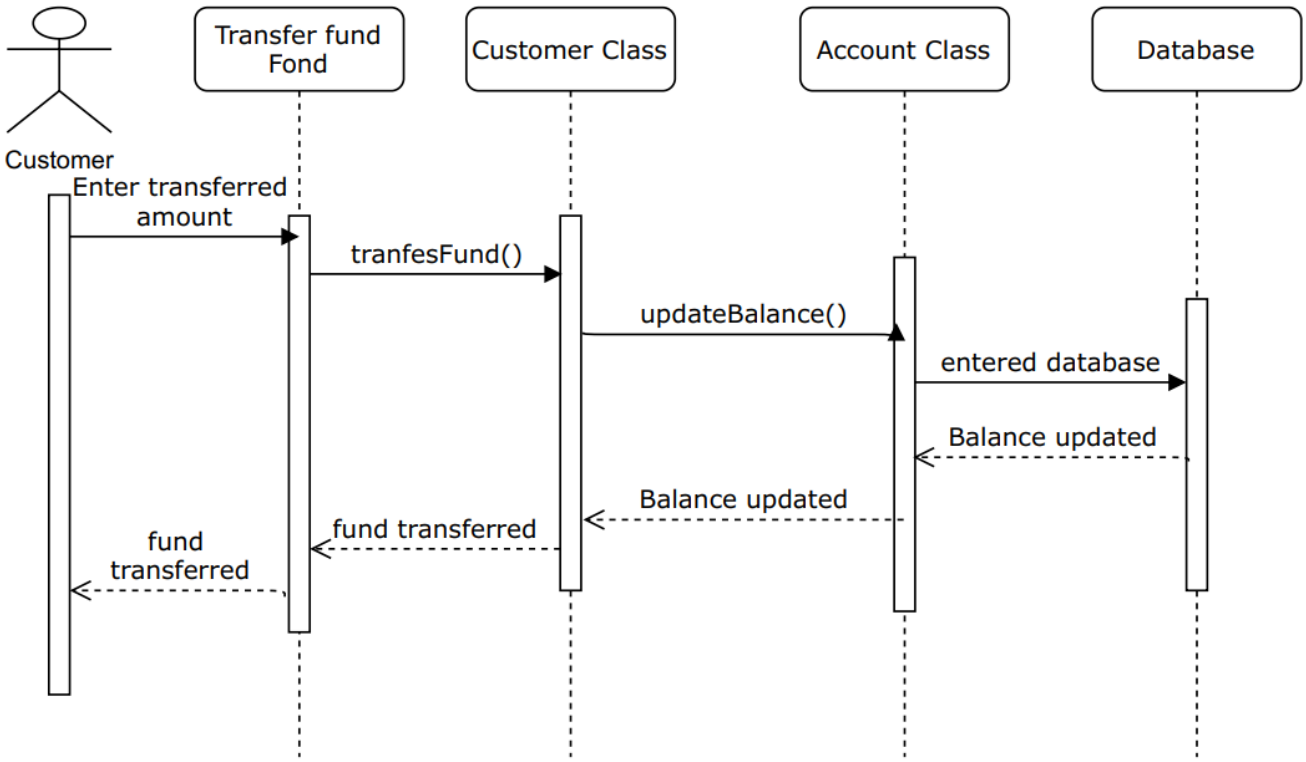
HTML: Hyper Text Markup Language.

TCP/IP: Transmission Control Protocol/Internet Protocol.

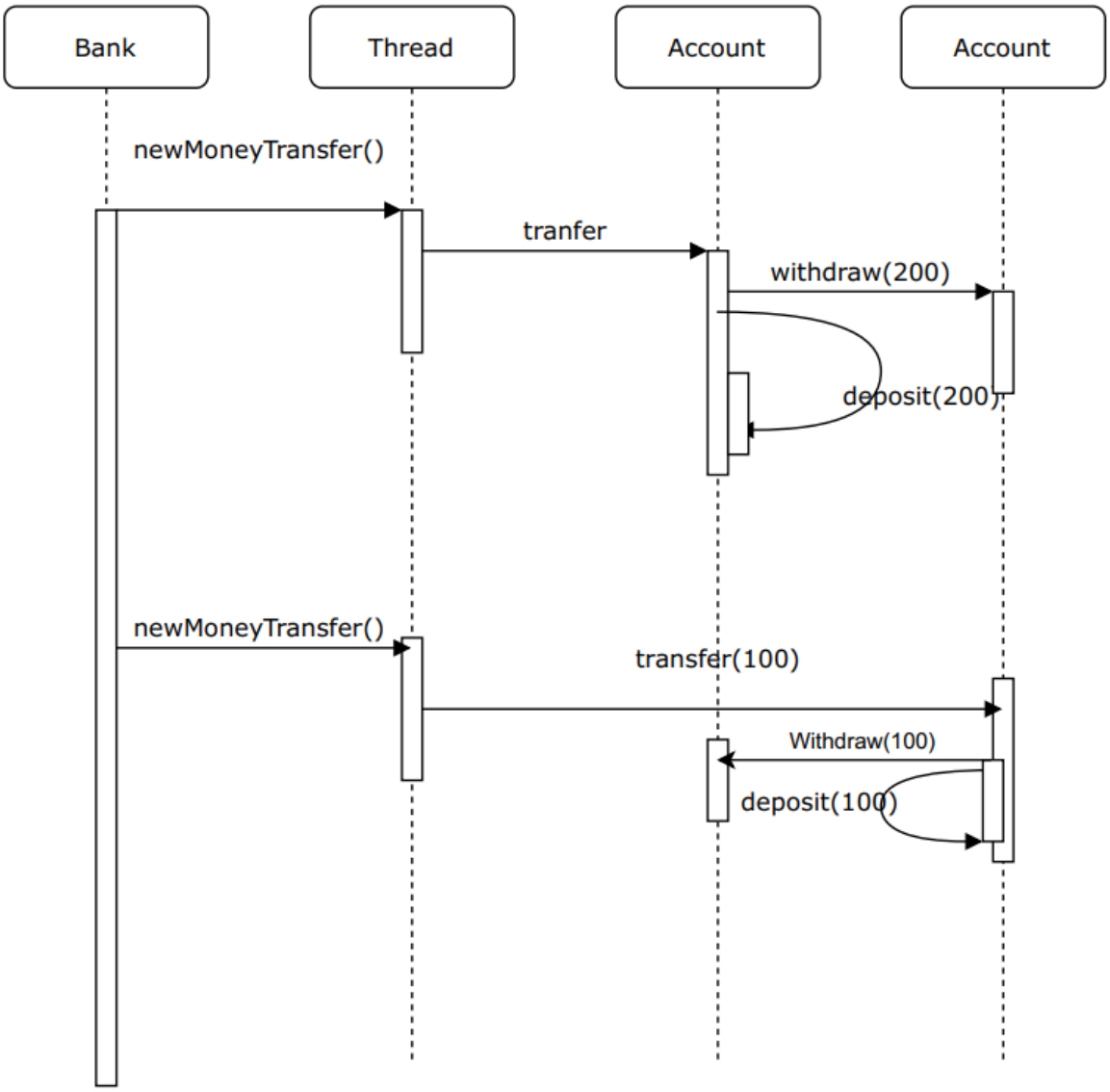
IP: Internet Protocol Address

Appendix B: Analysis Models

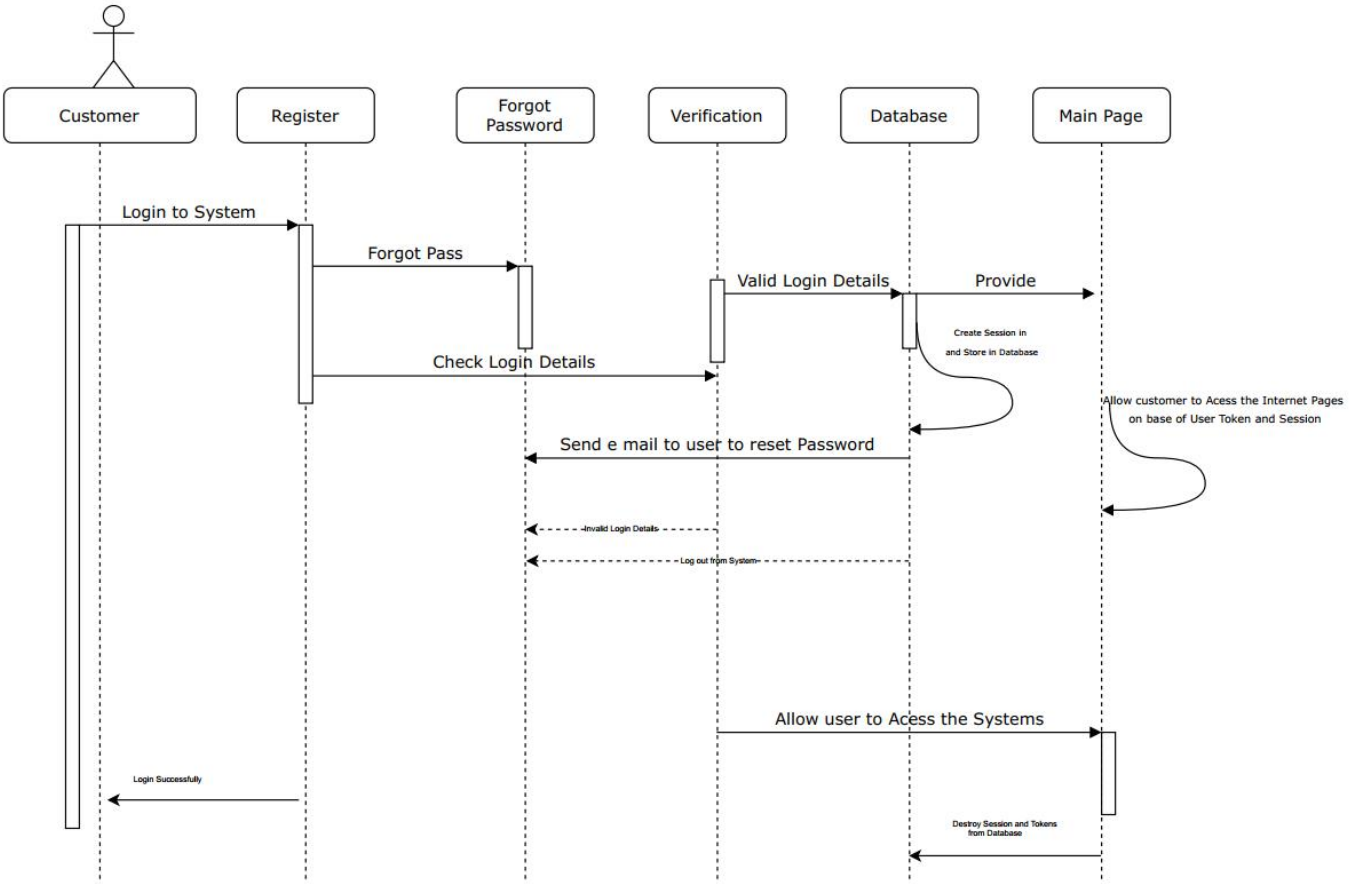
1. Customer Sequential Diagram



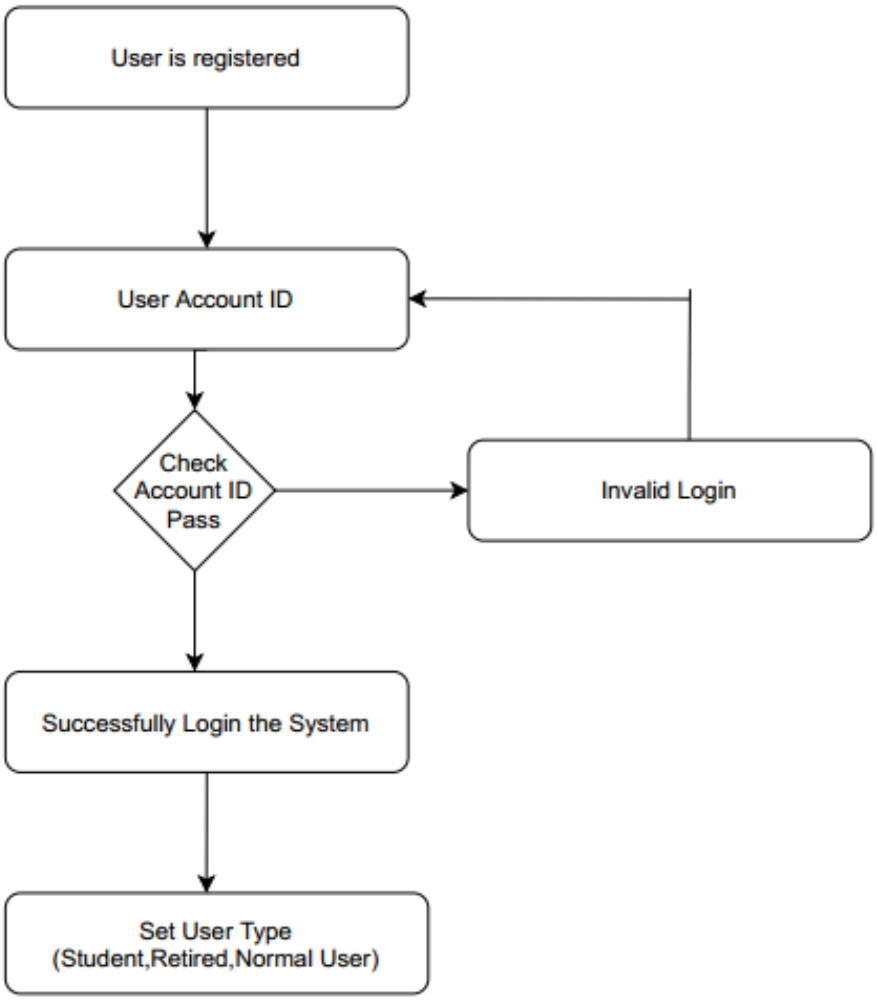
2. System Sequential Diagram


Bank
Thread
Account
Account
newMoneyTransfer()
tranfer
withdraw(200)
deposit(200)
newMoneyTransfer()
transfer(100)
Withdraw(100)
deposit(100)

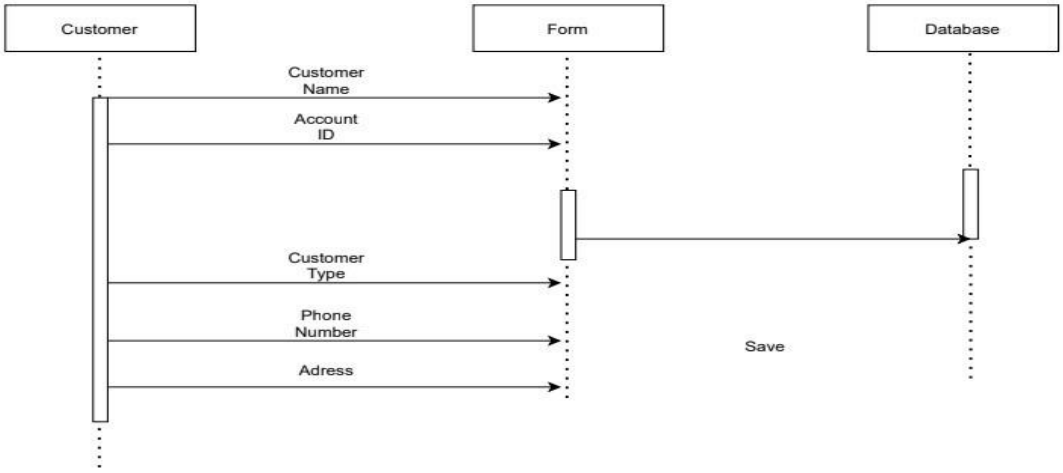
3. Customer Registration Sequential Diagram



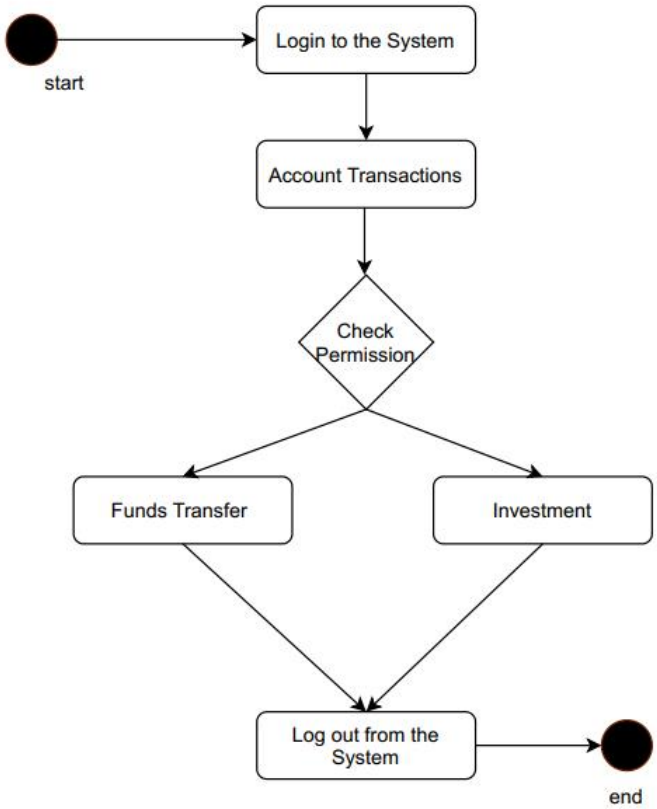
4. Login Activity Diagram



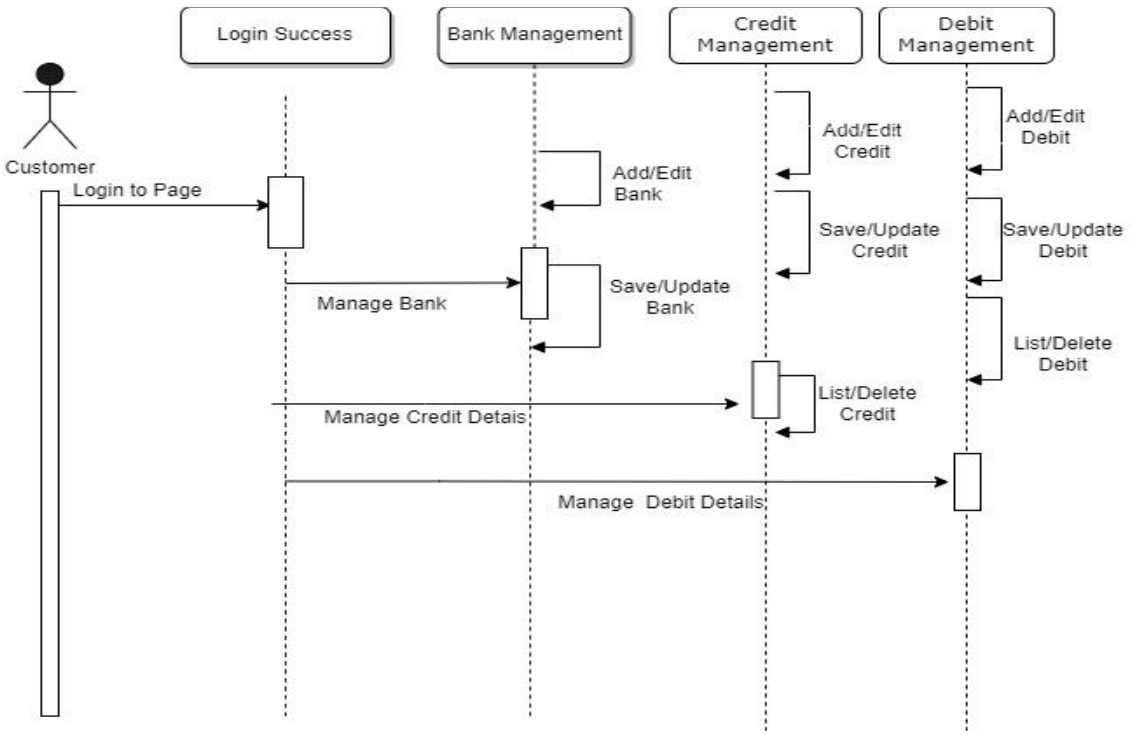
5. Customer Sequence Diagram



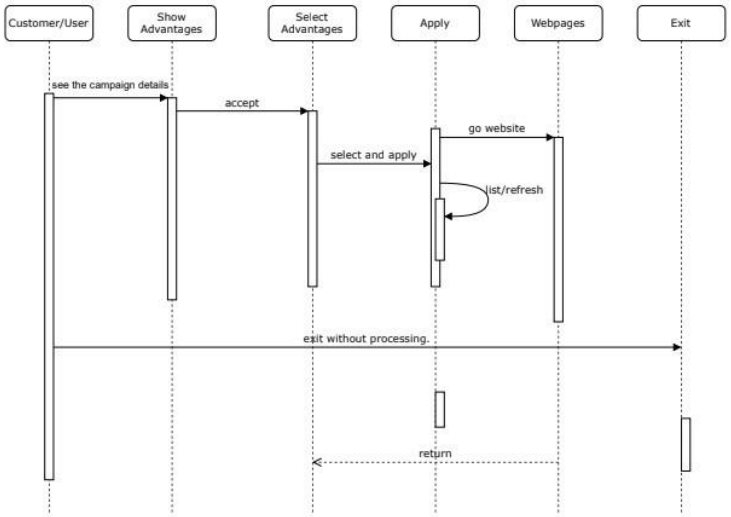
6. Money Transfer Activity Diagram



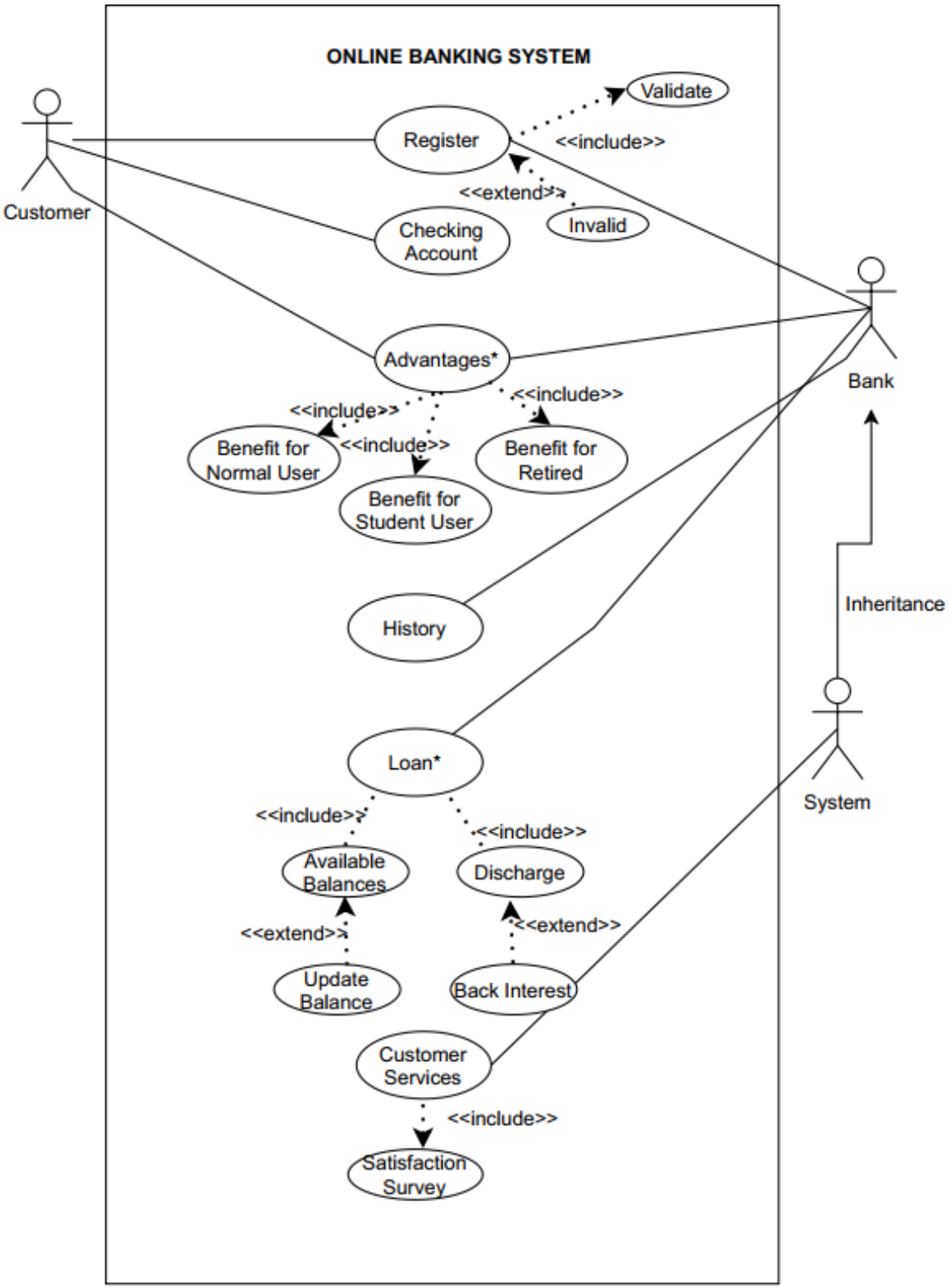
7. System Management Sequence Diagram



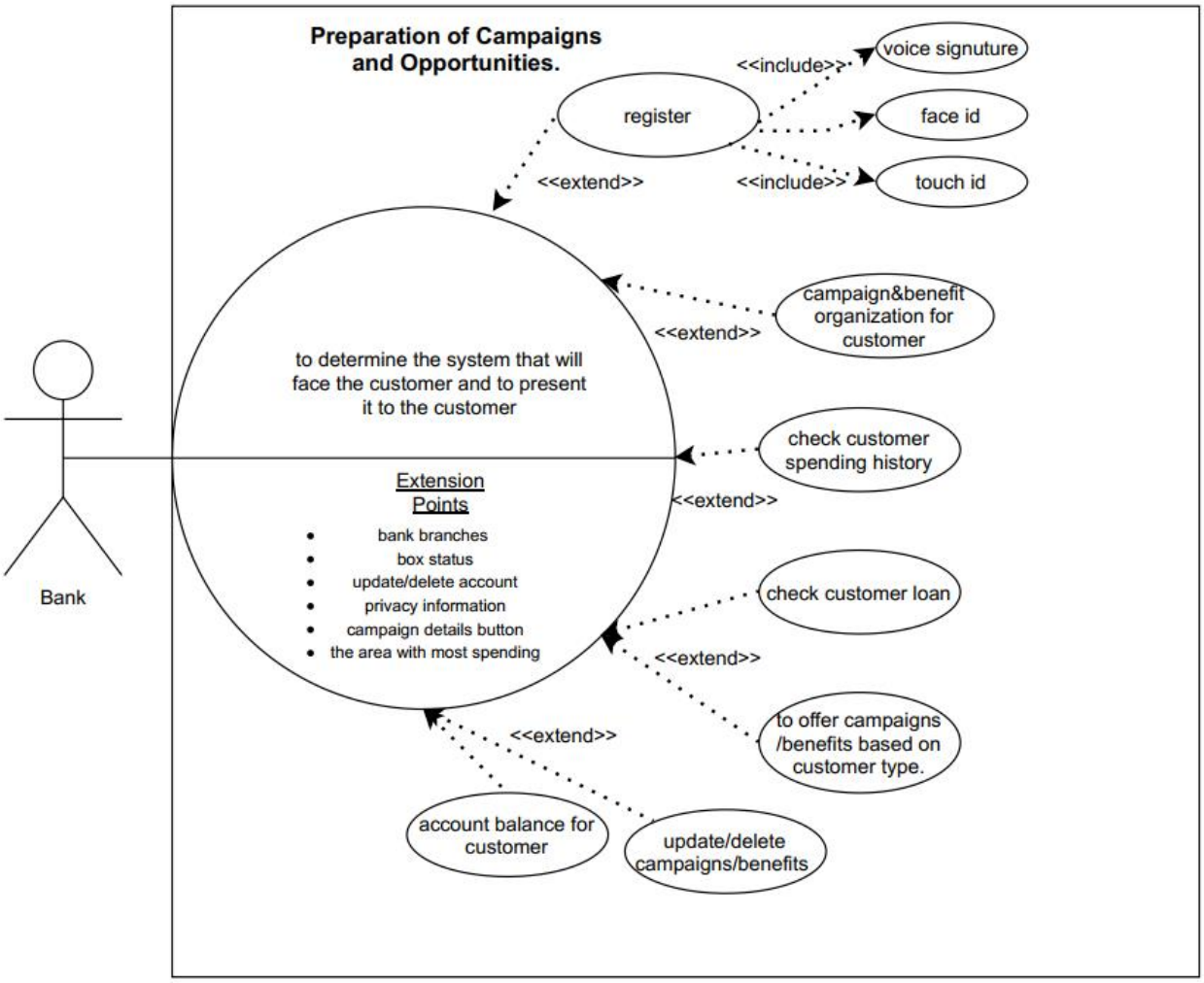
8. Advantage and Campaign Select Sequence Diagram



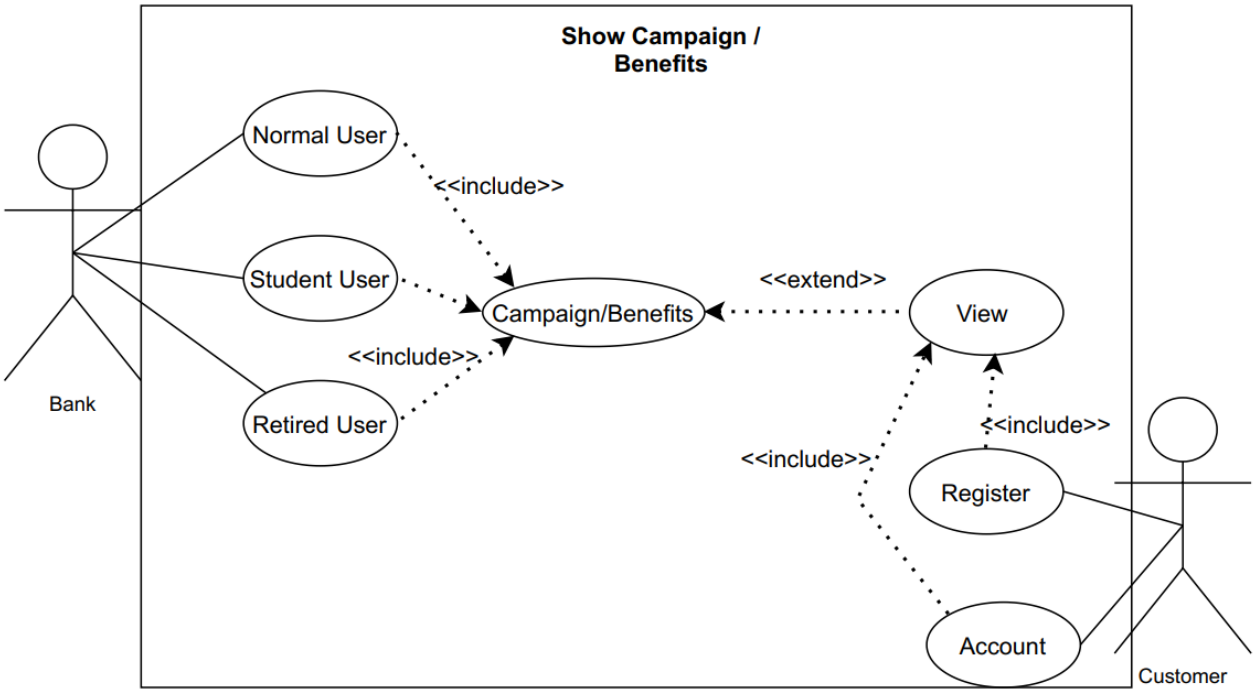
9. Online Banking Use Case Diagram



10. Preparation of Campaigns/Advantages Use Case Diagram



11. Show Campaigns/Advantages Use Case Diagram



Appendix C: To Be Determined List