TEENOVATORS

Internet Banking System
Software Requirements Specification

College

IT-BHU, Varanasi

Team Members

1) Ajay Kr. Gautam
2) Gaurav Gupta
3) Karan Jain
4) Rohit Jaiswal

Faculty Guide

Prof. A.K.Agarwal
Head, Dept. of Computer
Science and Engineering,
IT-BHU, Varanasi.

Contents

Software Requirements Specification

1. **Introduction:** Online Banking or Internet Banking allows customers to conduct financial transactions on a secure website operated by their retail or virtual bank, credit union or building society.

   i. **Purpose:** With the advent of technologyand introduction of computerization in almost every aspect of life, the Banks do aim at providing ease to one's life. Hence, a demand for an online comprehensive solution to manage Internet Banking exists. It shall be a system accessible to all its customers through a valid userID and password allotted to them. They can avail facilities such as viewing their balance & account statements; transferring funds from one account to another; request for cheque book, stop payments of cheques, bill payments and many more.

   ii. **Scope:**

   a. Includes a Virtual Keyboard option for prevention of Keylogger attacks.
   b. Provision of facility of blocking user's account on 3 succesive invalid attempts to login.
   c. View/Select/Map an account to your userID and view its corresponding information.
   d. View/Edit your profile details.
   e. Change password and other settings of the account.
   f. Map a loan account to your userID.
   g. Pay/Print/Save bills & corresponding receipts and others services such as mobile recharge, card to card transfer, etc.
   h. Issue Standing Instructions.
   i. Request Cheque Book/Draft.

j. View status of Cheques/Drafts; Stop payment of Cheques/Drafts.
k. View Transaction Reports; View Account statements for any desired period in form of statistics & graphs.
l. Enable high security option where SMS is sent to mobile for confirmation.
m. Transfer funds from one account to another and also to 3^{rd} party accounts.
n. A very useful search functionality.
o. Providing letter of Credit/Guarantee, etc.
p. Blocking & Unblocking of account (facility of admin)
q. Proper logs are created and accessible by Administrator.
r. Customizable themes & templates.
s. A separate customizable JAVA based client for Administrator.

iii. Definitions:

a. ACID Properties:

- **A**tomicity: Either all operations of a transaction are reflected properly in a database or none are.

- **C**onsistency: Execution of a transaction in isolation i.e. with no other transaction executing concurrently preserves the consistency of the database.

- **I**solation: Even though multiple transactions may execute concurrently, the system guarantees that for every pair of transaction T_i and T_j , it appears to T_i that either T_j finished execution before T_i started or T_j started execution after T_i finished. Thus, each transaction is unaware of other transactions executing concurrently in the system.

- **D**urability: After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.

b. E-R Model: Based on the perception of a real world that consists of basic objects called Entities and Relations between these. The set of entities of the same type are called Entity Set and relations of same type are called Relationship Set. E-R model places a constraint called mapping cardinalities which expresses the number of entities to which another entity can be associated.

c. **Remittance:** The process of sending money to remove an obligation. This is most often done through an electronic network, wire transfer or mail. The term also refers to the amount of money being sent to remove the obligation.

d. **Standing Instructions:** Alerts that a customer can give to the bank to permit it to debit a fixed amount of money from his account for a particular reason. The bank charges some commission for this purpose.

e. **Keylogger Attacks:** This is a type of attack where a malicious user installs software on the system which logs/records all the keys pressed through the keyboard and can record relevant information.

iv. References:

➢ Online References:

- http://www.chase.com/
- http://www.anz.com/
- http://www.standardchartered.com.sg/
- https://www.bankofamerica.com/
- http://www.citibank.com/us/index.htm
- http://www.usbank.com/
- http://www.marketwatch.com/
- http://www.creditcards.com/

- http://www.stumbleupon.com/demo/?review=1#url=http://www.favoritesworld.com/
- http://www.britishairways.com/travel/informationhubpage/public/en_us?source=TOP_information
- http://www.worldbank.org/

- ➢ IEEE SRS Format
- ➢ Project Scenario and specification given by IBM

v. **Technologies:**

i. Eclipse IDE

ii. WASCE-WebSphere Application Server Community Edition

iii. DB2 Express

iv. AJAX

v. XML

vi. RSA – Rational Software Architect

vi. **Overview:** The remaining portion of SRS si divided into two sections:

- ➢ *Overall Description* will describe the major components of the system, interconnection and external interfaces.
- ➢ *Specific Requirements* will describe the function of actors, their role in the system and constraints.

2. Overall Description:

I. **Product Perspective:** There are 3 aspects of perspection to this project:

   i. Customer View

   ii. Industrialists'/Organizations'/Entrepreneurs' View

   iii. Administrator View

II. **Product Function:** The various sections of the product are described below with their function:

   i. **Login Module:**

   This portion is meant for authenticating a valid user. Since this is the only entry point into the system, proper security measures like encryption of data, virtual keyboards, prevention of script execution by image verification, blocking account on 3 successive invalid login attempts have been implemented.

   ii. **MyProfile Module:**

   This section shall primarily focus on my personal information. It has the options of View/Select/Map accounts to the userID, View/Edit account information, changing Account Settings such as the security level, password, etc. One can also map his loan account to the userID.

iii. **Services Modules:**

It includes features such as bill payment facilities, mobile phone recharge, sending money to some address, card to card transfer, etc.

iv. **Standing Instructions:**

Under this section, we provide customers a facility to issue an instruction to the bank to debit/transfer money from his/her account and credit it to someone else's account on a particular date repeatedly after a fixed interval of time.

v. **Cheques/Drafts:**

Under this section we deal with viewing the status of Cheques/Drafts. We can also request to stop the payment of cheques/drafts. Besides, there are also options for requesting to issue a Cheque Book.

vi. **Letter of Credit/Guarantee:**

This is a facility of most use to the entrepreneurs/organizations. This is meant in order to transfer large amounts of money which cannot be taken in hand for business purposes.

vii. **Funds Transfer:**

Under this one can transfer money between 2 accounts mapped to a single userID as well as transfer money to a 3rd party as well.

III. **Data Flow Diagram:** The following diagram indicates the flow of information data through our system.

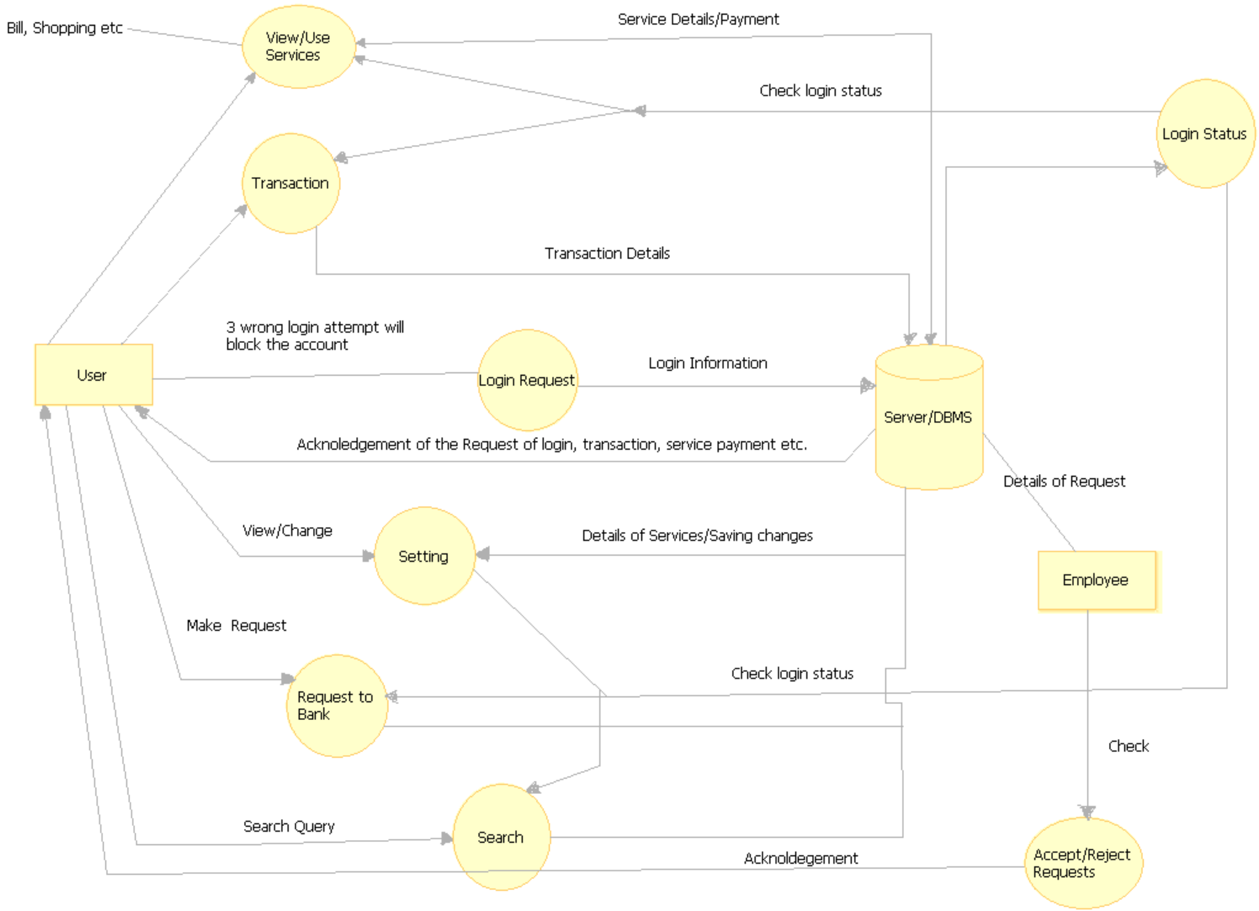


Figure 1 : Data Flow Diagram for Internet Banking System

IV. **Class Diagram:**

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes and the relationships among the classes.

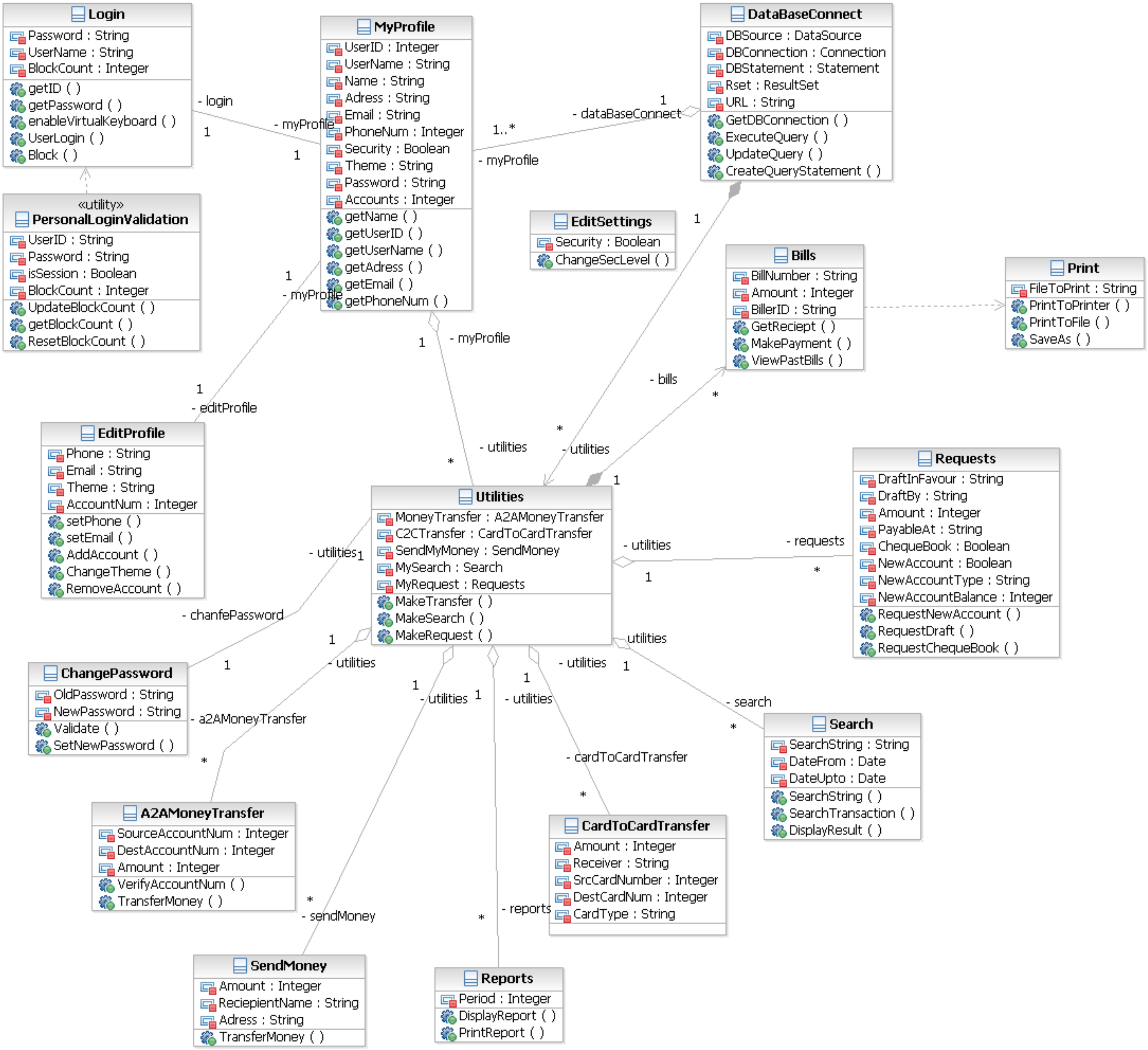


Figure 2: Class Diagram for Internet Banking System

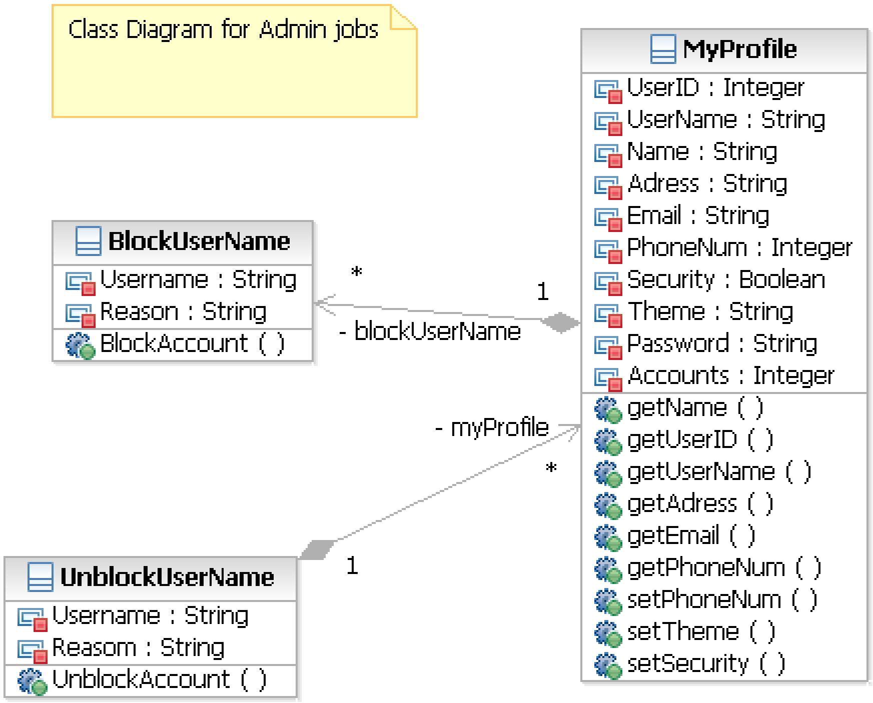


Figure 3: Class Diagram for Admin Controls

V. **Database Design:** This essentially consists of two sections:

i. Schema Design

ii. E-R Diagram

Schema Design:

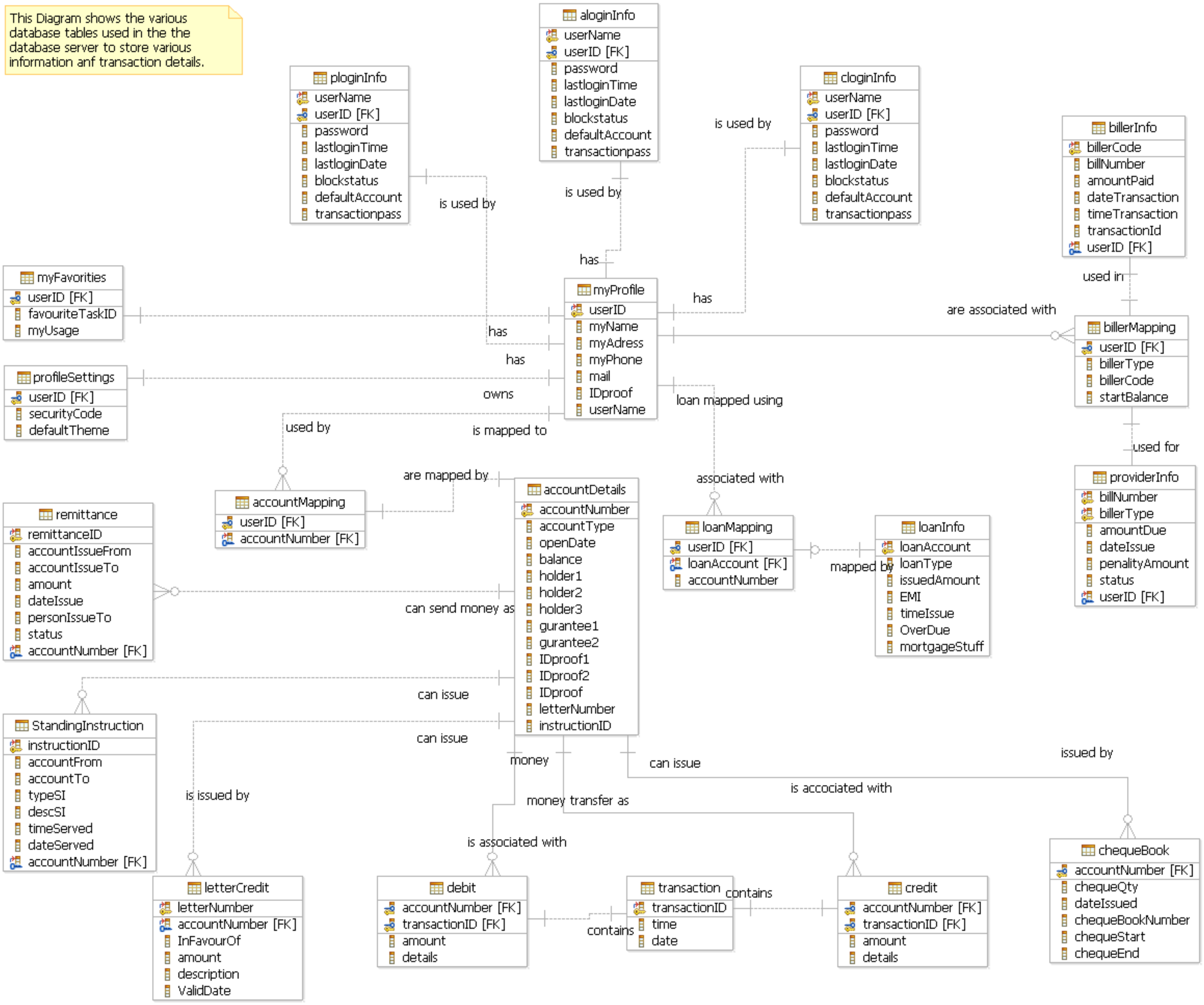


Figure 4: Schema Diagram for Internet Banking System

E-R Diagram: The Entity-Relationship model for various relations and entity sets is illustrated through the following diagram. The standard notations are followed in the diagram.

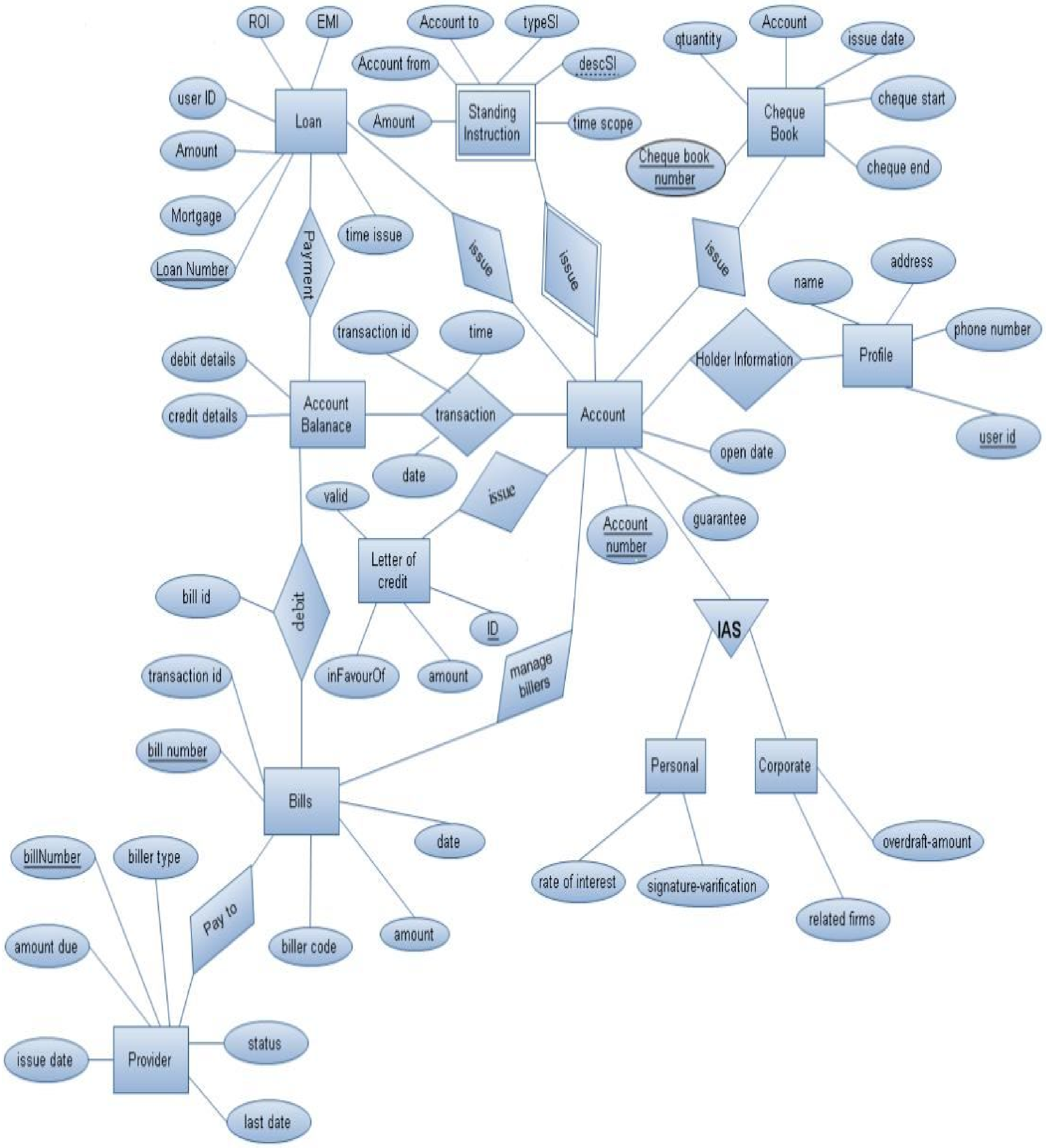


Figure 5: E-R Diagram for Internet Banking System

VI. Architecture Diagrams

Two types of architecture diagrams are proposed:

i. Deployment Model

ii. Multi-Layer Architecture

Deployment Model:

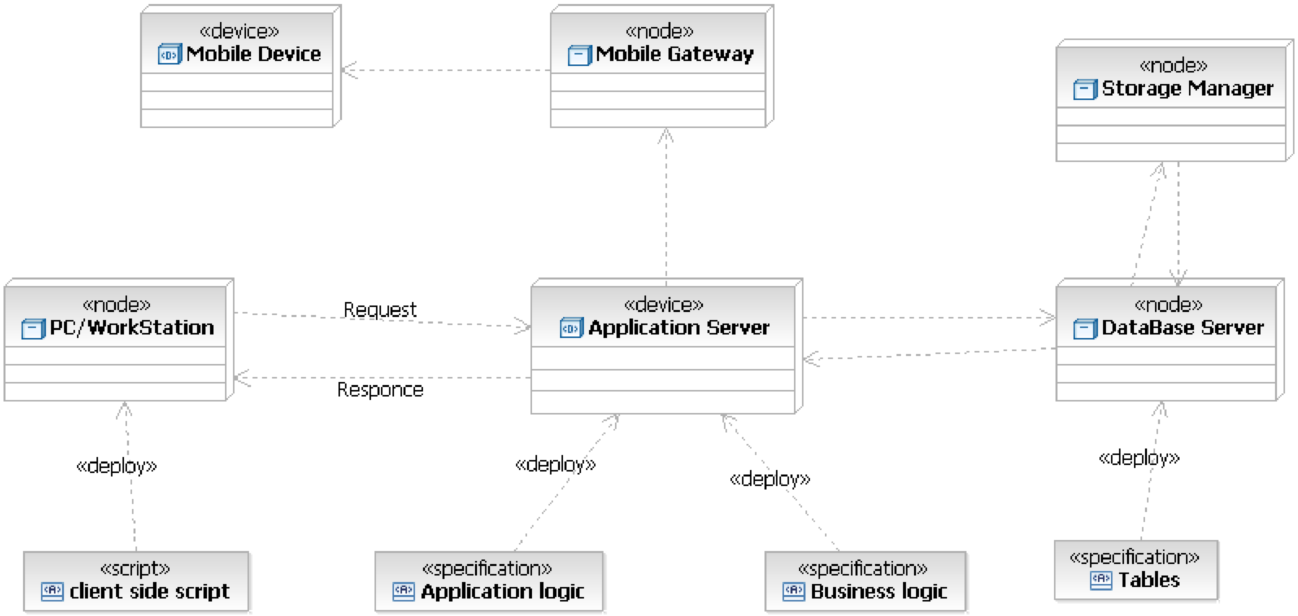
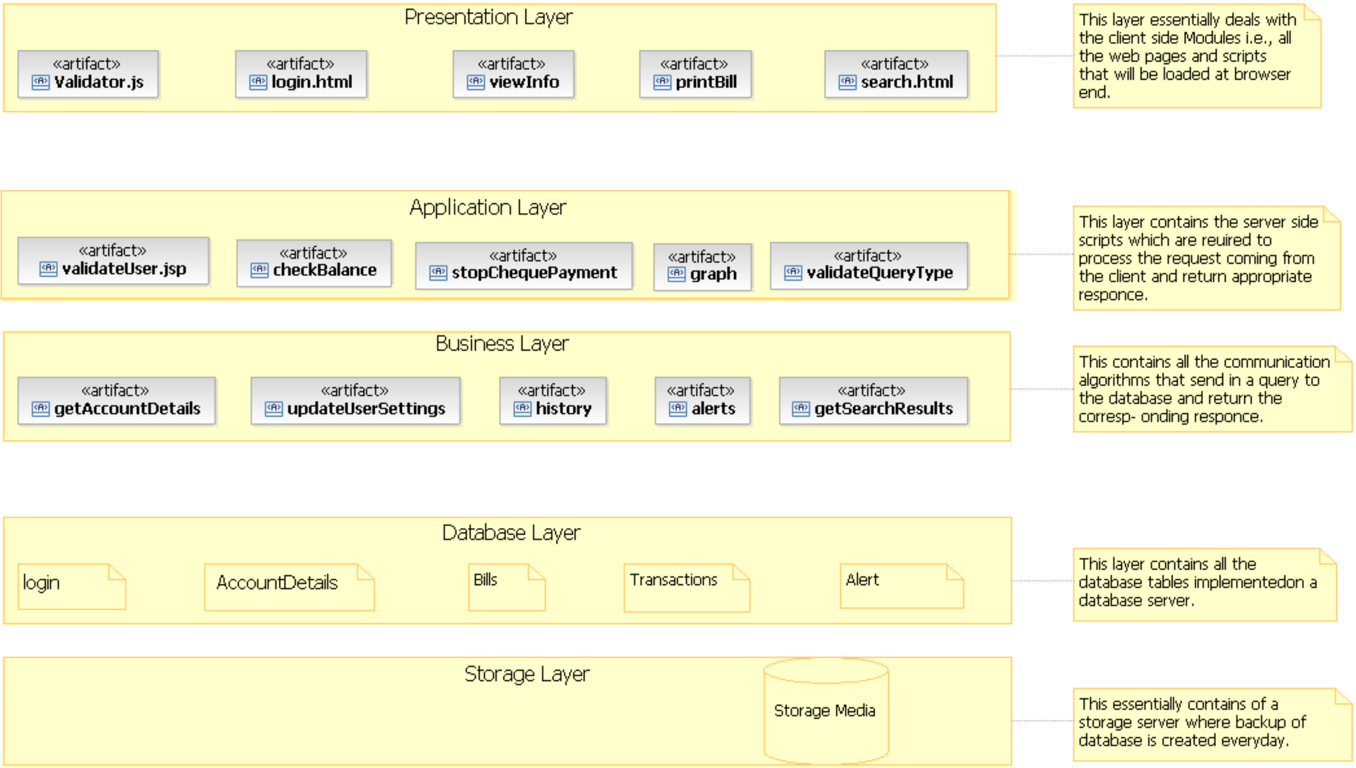


Figure 6: Deployment Diagram for Internet Banking System

Multi-Layer Architecture / 3-Tier Client Server Structure:



This layer essentially deals with the client side Modules i.e., all the web pages and scripts that will be loaded at browser end.

This layer contains the server side scripts which are reuired to process the request coming from the client and return appropriate responce.

This contains all the communication algorithms that send in a query to the database and return the corresp- onding responce.

This layer contains all the database tables implementedon a database server.

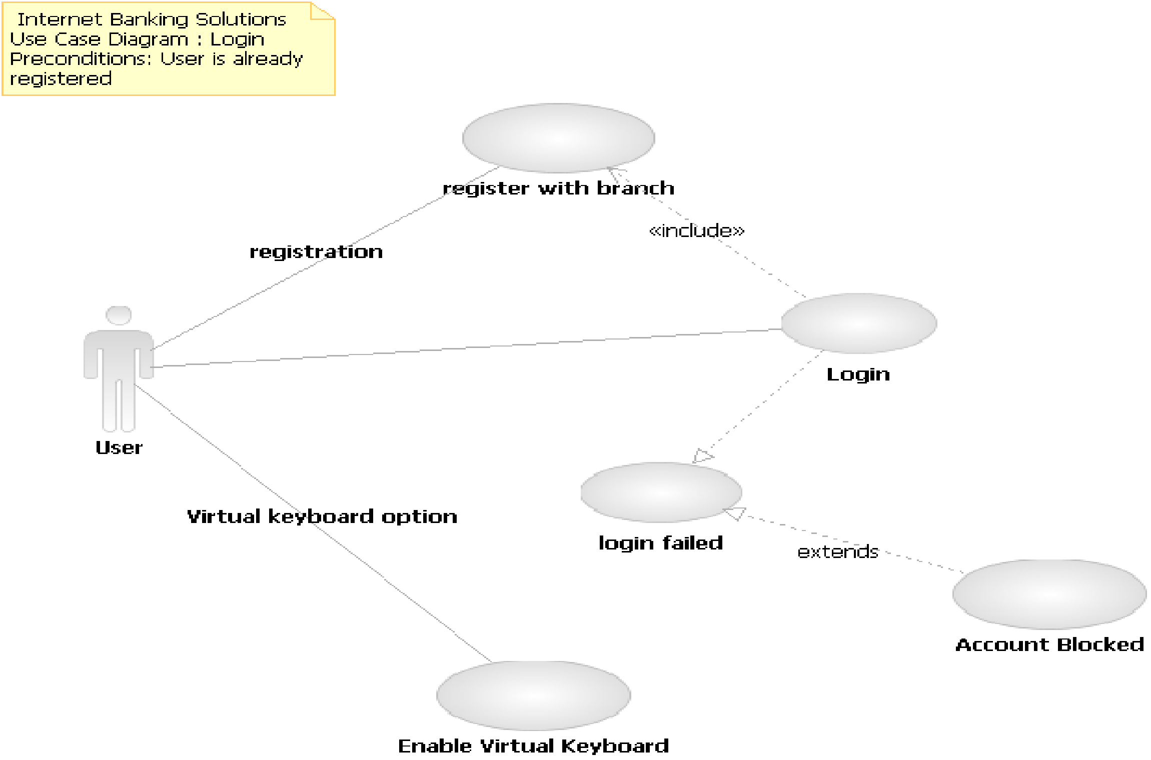
This essentially contains of a storage server where backup of database is created everyday.

Figure 6: Multi-Tier Architecture Diagram for Internet Banking System

Specific Requirements:

I. <u>Use-case Reports:</u>

a) <u>Name of Use Case</u> **: Login**



<u>Figure-7:</u> *Login Use Case*

specifications:

<u>Actors:</u> Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

<u>Pre-Conditions:</u> User is already registered with the bank.

<u>Post-Conditions:</u> User logged in or login failed.

Basic Flow:

- The user will be asked to provide the *User name* and *Password* to the browser.
- The user can enable **Virtual Keyboard** for security purpose..
- The user will provide the *user name* and *password.*
- The *user name* and *password* will be encrypted at client side.
- The encrypted data will be sent to server.
- The server will return the login session status to the client.
- User will be directed to the ***My Profile*** page

Alternate Flow : *Login failed*

- In case of wrong *user name* and *password* the server will return the error status.
- The user will be informed about login failure.
- A popup message will tell the user that on three wrong login attempts, the account will be blocked.
- If three wrong login attempts are made, the account will be blocked.

Alternate Flow *: Account Already blocked*

- The server will return the information that account is blocked
- A popup message will tell the user to contact the nearest branch of the bank to unblock the account.

Special Requirements: none

Use case relationships : none

b) Name of Use Case: **Map Account**

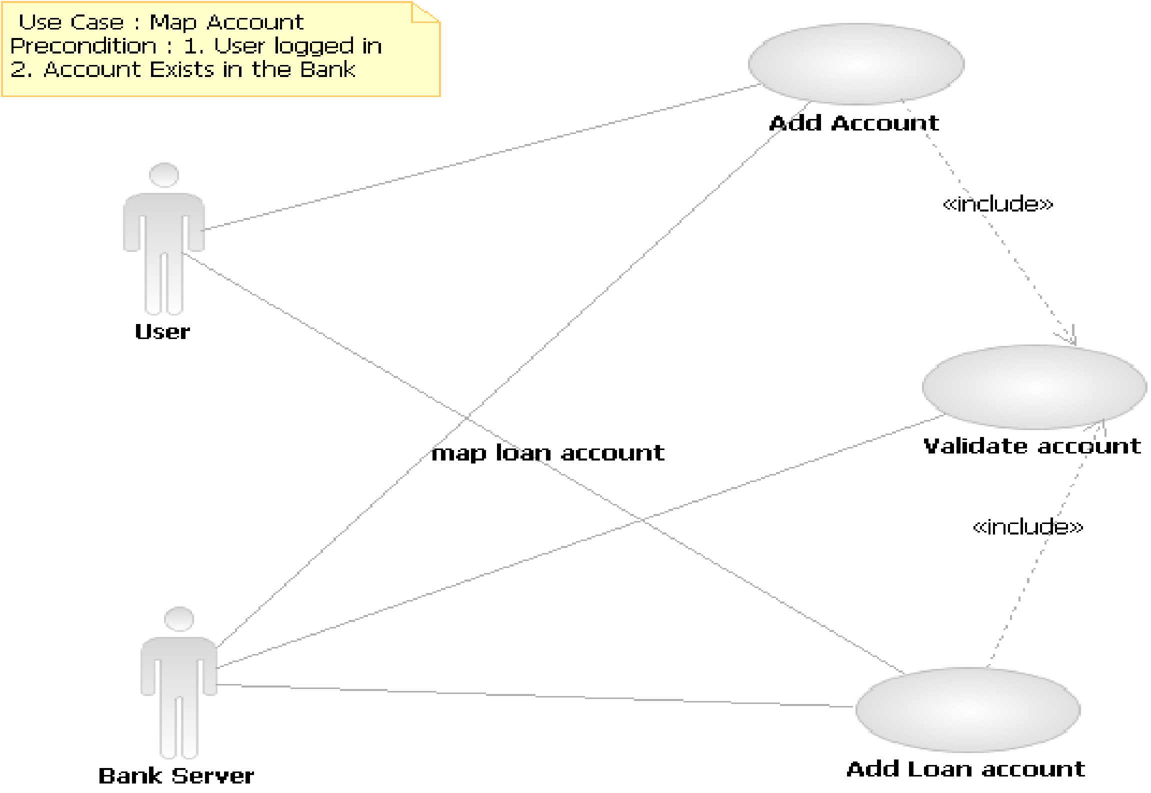


Figure-8 Map Account Use Case.

Description: One can map his loan/basic account to his profile.

Specifications:

Actors: Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions:

- User is already logged in.

Post-Conditions:

- New Account is mapped to the profile.

Basic Flow:

- The user will provide the account number.
- The account number will be validated by the server.
- The new account will be mapped to the profile.
- User will be informed.

Alternate Flow: ***Invalid account number***

- The User will be informed that account number is invalid.
- User will be asked to try again.
- Control will be transferred to the initial page.

c) Name of Use Case: **Edit Profile**

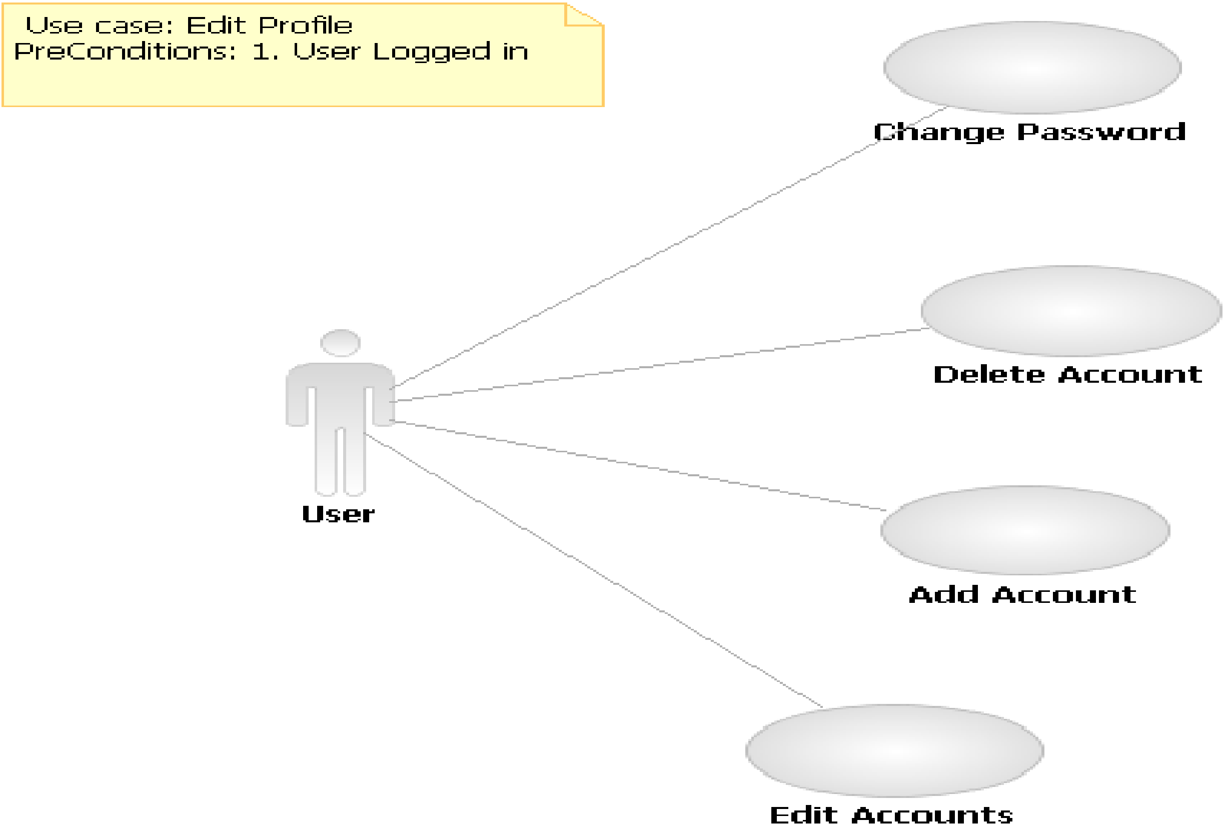


Figure-9 Edit Profile Use Case.

Description: Through this a user can edit his user profile.

Specifications:

Actors: Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions:

- User is already logged in

Post-Conditions:

- Profile is saved·
- Profile settings are applied.

Basic Flow:

- The user will be redirected to Edit profile page.
- The user will be required to enter the new information .
- User will submit the data.
- User will be asked for confirmation.
- Profile data will be saved.
- Profile settings will be applied.

d) Name of Use Case: **Edit Settings**

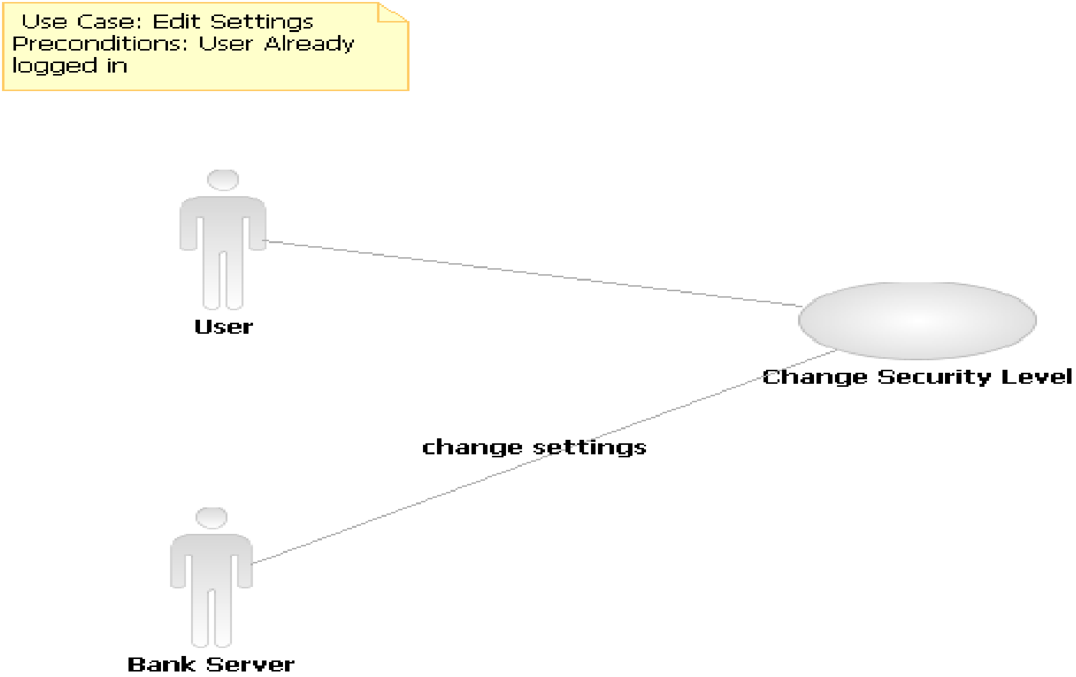


Figure-10 Edit Settings Use Case.

Description: Through this, a user can modify his account's security level.

Specifications:

Pre-Conditions:

i. User is already logged in

Flow of Events:

i. User selects the option which he wants to edit and gives the corresponding new information.

ii. For the option selected, the pre-conditions must hold in current system state.

iii. On clicking update, the system settings are updated.

e) Name of Use Case: **Account to Account money transfer.**

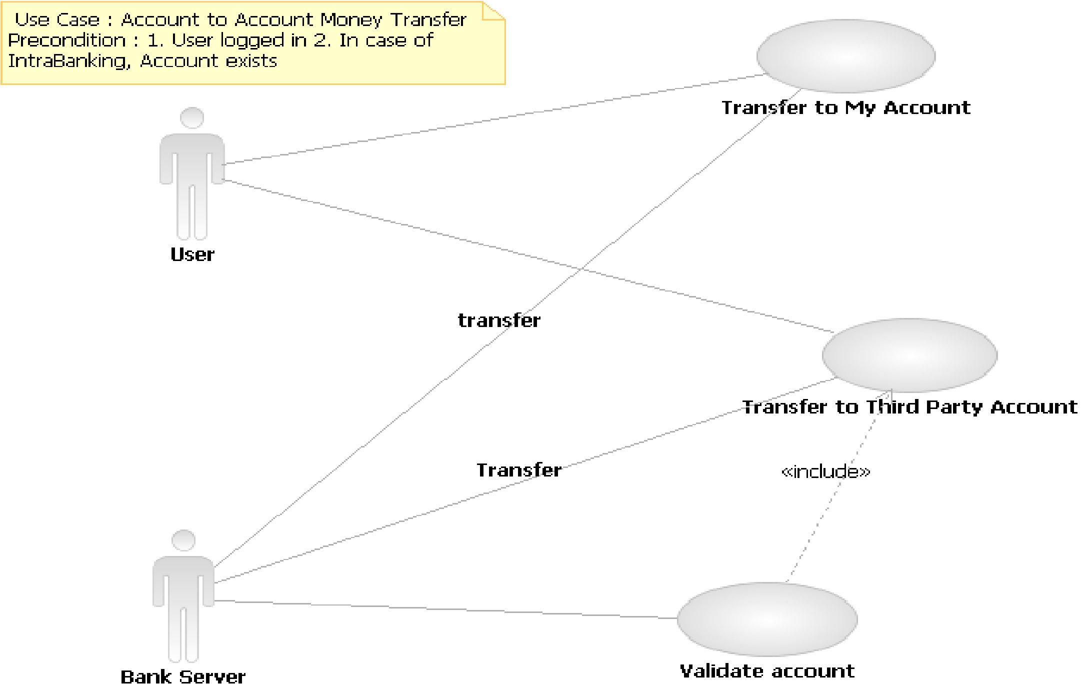

Use Case : Account to Account Money Transfer
Precondition : 1. User logged in 2. In case of IntraBanking, Account exists
Transfer to My Account
User
transfer
Transfer to Third Party Account
Transfer
«include»
Bank Server
Validate account

Figure-10 Account to Account money transfer Use Case.

Description: This module will let the user transfer money from one account to another.

Specifications:

Actors: Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions:

- User is already logged in.

Post-Conditions:

- Account balance is in consistent state.

Basic Flow:

- The user will select an account from existing accounts to make the transfer of money.
- In case of 3rd party account, the user will provide the account number.
- In this case, the account number will be validated by the server.
- User will input the amount to be transferred.
- The user will be asked for Transaction password.
- Transaction will be committed.
- User will be informed.

Alternate Flow : *Invalid Third party account number*

- The User will be informed that account number is invalid.
- Control will be returned to the first page.

f) Name of Use Case: **Print Bills, Receipt etc.**

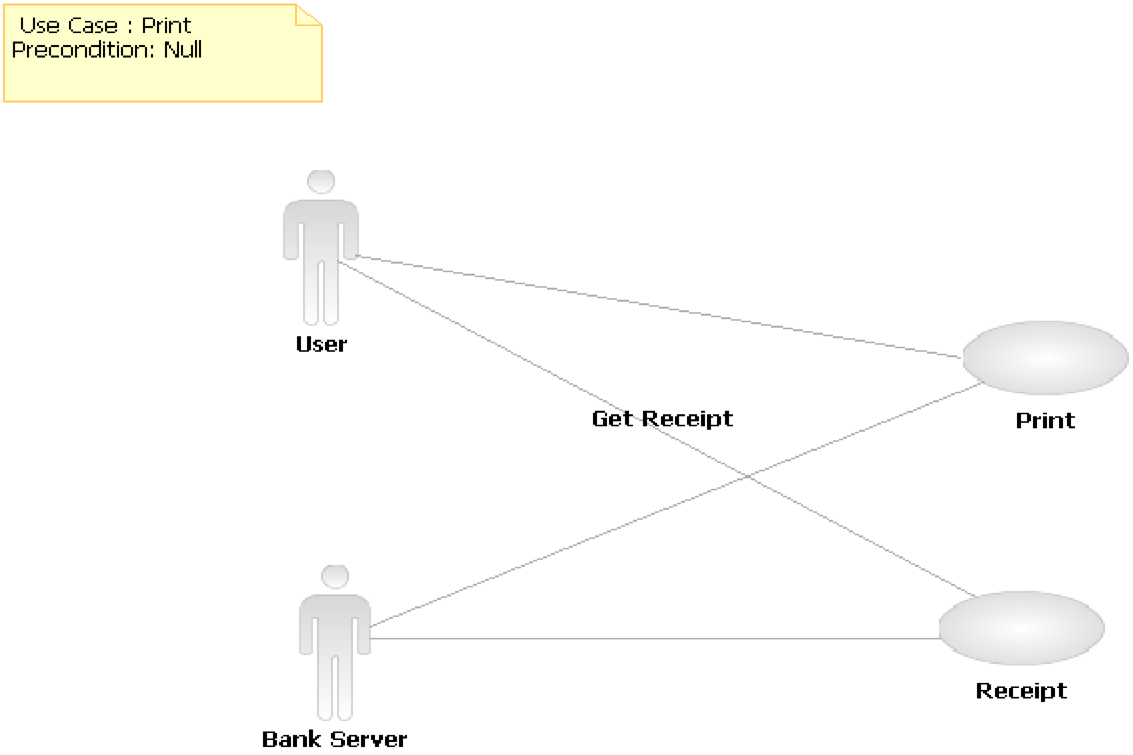


Figure-11: Print document Use Case.

Description: This module is made use of atmany places to offer the service of printing in a particular format.

Specifications:

Actors:

- Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions

- User logged in.

Post-Conditions:

- None.

Basic Flow:

- The user will select a document to print.
- User will be asked to print to printer/file or save as a file.
- The control will be transferred to the server.
- Server will send the document as a specific file format or to print.
- The document will be printed/saved.
- User will get a receipt.

g) Name of Use Case: **Requests**

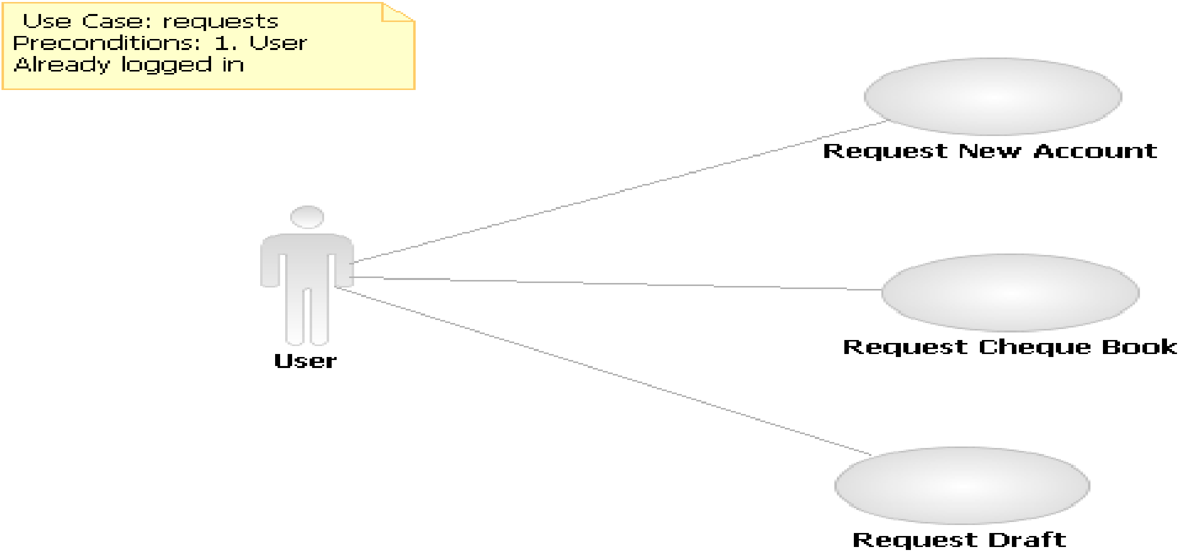


Figure-12: Requests Use Case.

Description: Through this module, a user can request bank to issue a cheque book to his account or can also request bank to issue a draft to him.

Specifications:

Actors:

- Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions:

- User is already logged in.

Post-Conditions

- Request is added to be processed.

Basic Flow:

- The user will add his request.
- The request will be queued for processing.
- User will be informed.

Name of Use Case: **Search Utility**

Description: Using this module, a user can search for a desired set of outputs such as his previous transactions, etc.

Specifications:

Actors: Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions:

- User is already logged in

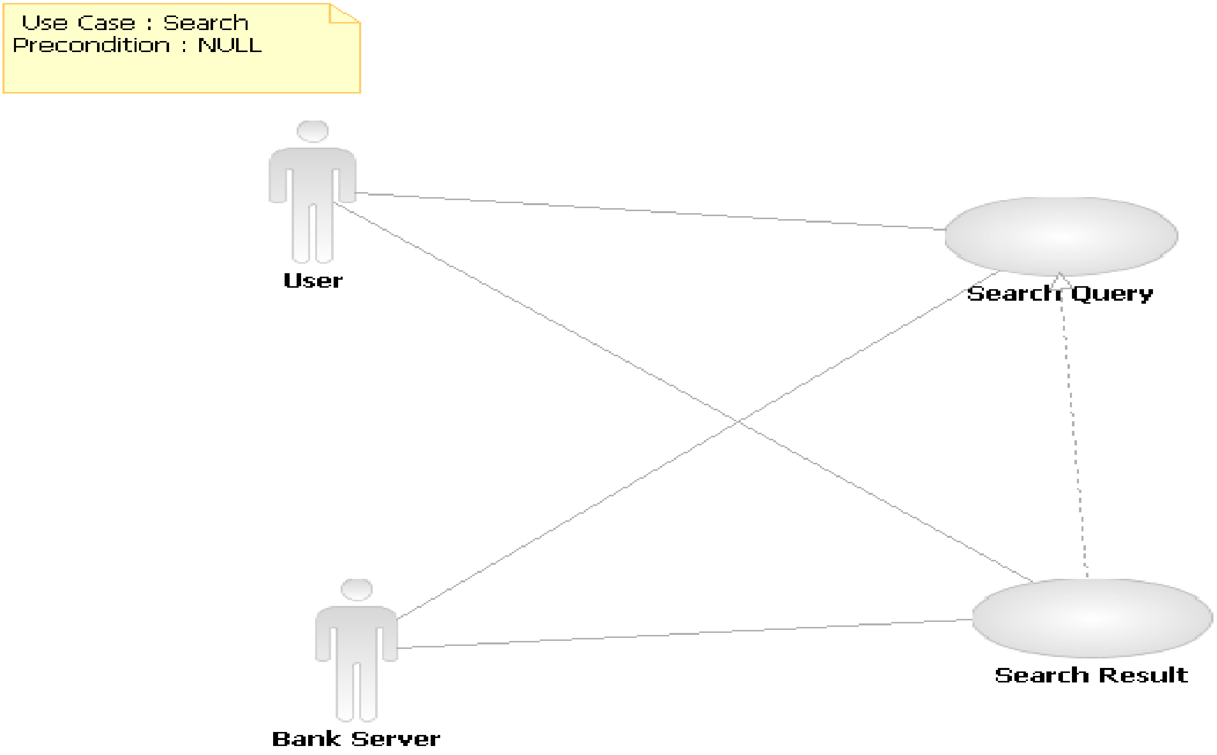


Figure-13: Search Utility Use Case.

<u>Post-Conditions:</u>

- None

Basic Flow:

- The user will provide the search string and domain to be searched.
- The request will be forwarded to the server.
- The server will return the result.
- Result will be displayed.

h) Name of Use Case: **Issue Standing Instruction**

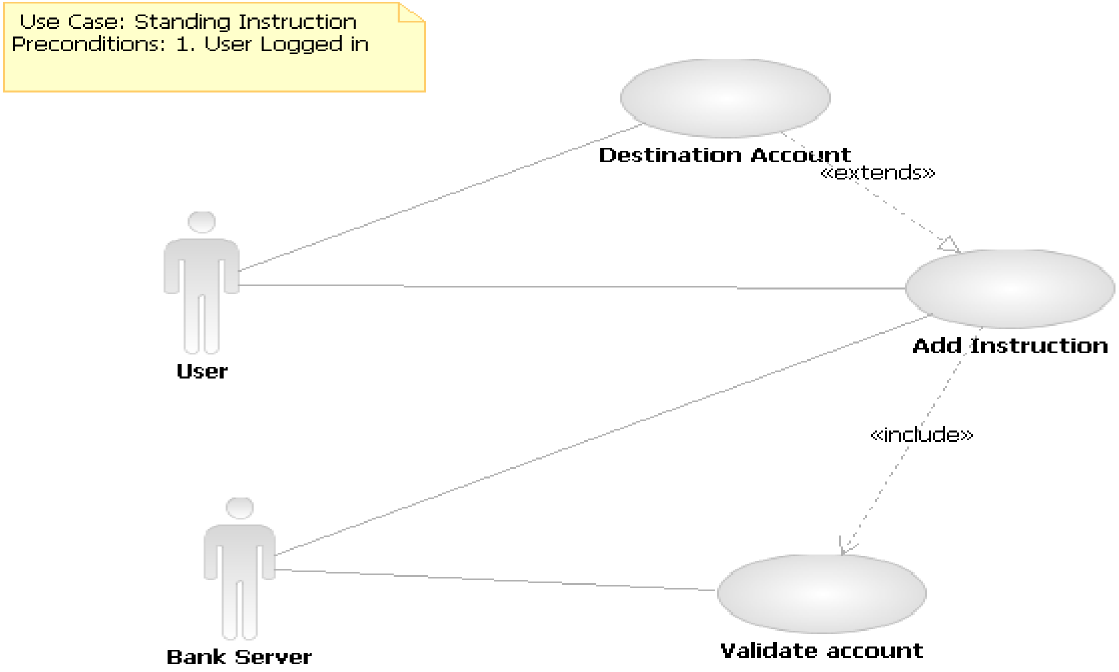


Figure-14: Issue Standing Instruction Use Case.

Description: A user can issue an instruction to the bank wherein he can instruct the bank to transfer money from his account to some other account periodically.

Specifications:

Actors: Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions:

- User is already logged in.

Post-Conditions:

- Instruction is added to the account.

Basic Flow:

- The user will choose/provide the source and destination Account
- The destination account will be validated by the server.
- The user will be asked for his *Transaction password.*
- On confirmation, the instruction will be added to the account.

Alternate Flow : Invalid destination account number

- The User will be informed that account number is invalid.
- Control will be returned to the first page.

I) Name of Use Case: **View Profile Information**

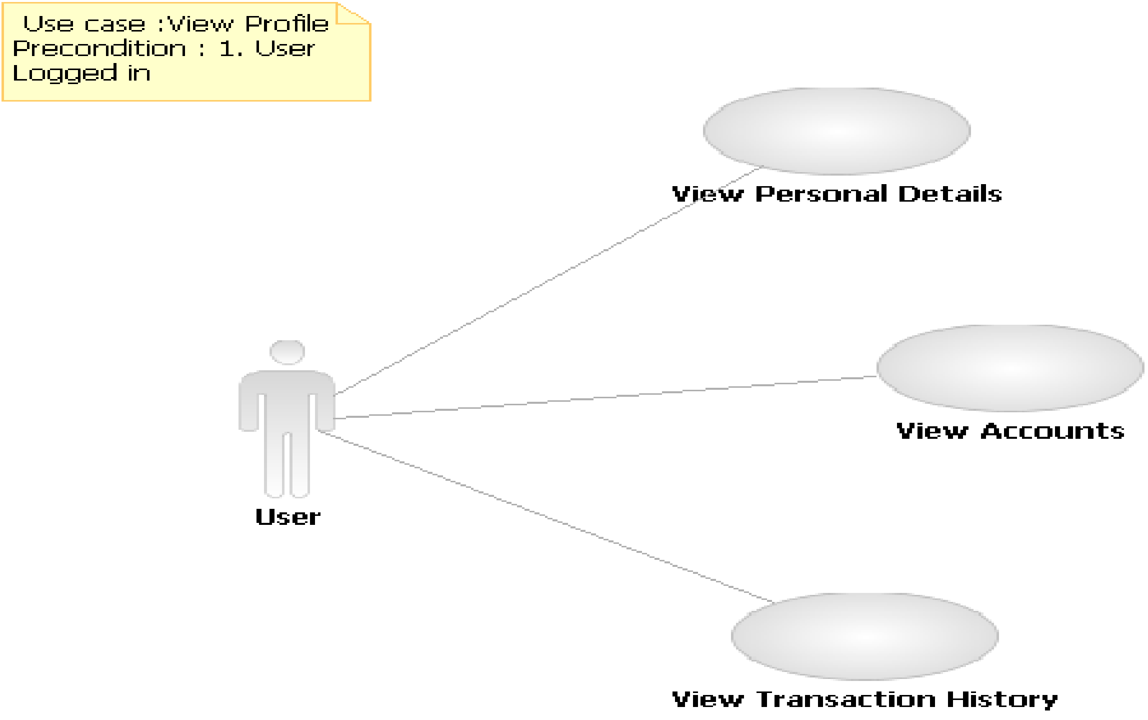


Figure-15: View Profile Information Use Case.

Description: Through this, one can view his profile information.

Specifications:

Actors: Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions:

- User is already logged in.

Post-Conditions:

- None

Basic Flow

- The user will login into his account.
- He/She will view his/her profile.

J) Name of Use Case: **Bill**

Description: Through this, one can pay his utility bills from his account.

Specifications:

Actors: Customers, Entrepreneurs, Industrialists, Organizations, Administrators, etc.

Pre-Conditions:

- User is already logged in.

Post-Conditions:

- None

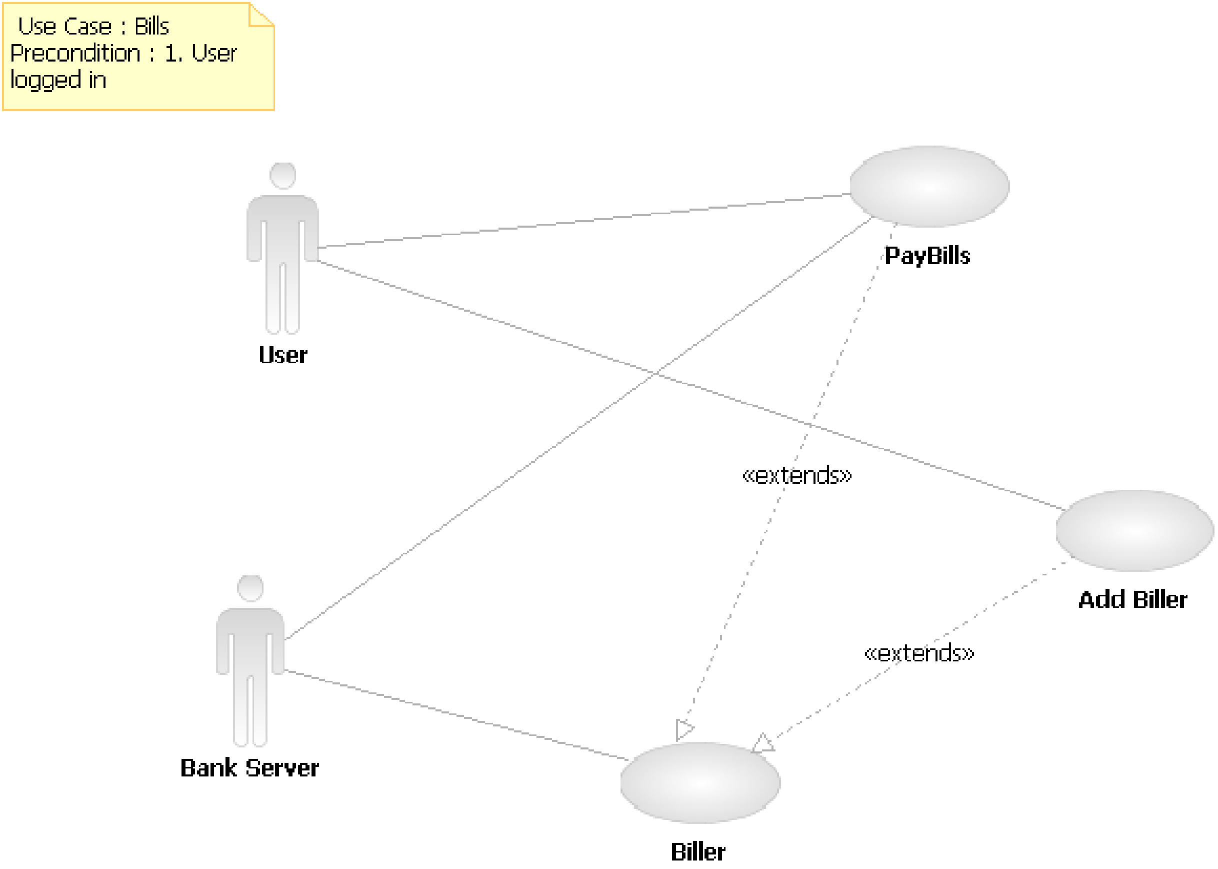


Figure-15: View Profile Information Use Case.

Basic Flow:

- The user will select a biller to pay a bill.
- If biller is not added to user's list, it will be added to the list.
- The user will be asked for transaction password.
- The bill will be paid.
- User will be informed and a receipt will be provided to the user.

K) Name of Use Case: **Block User Account**

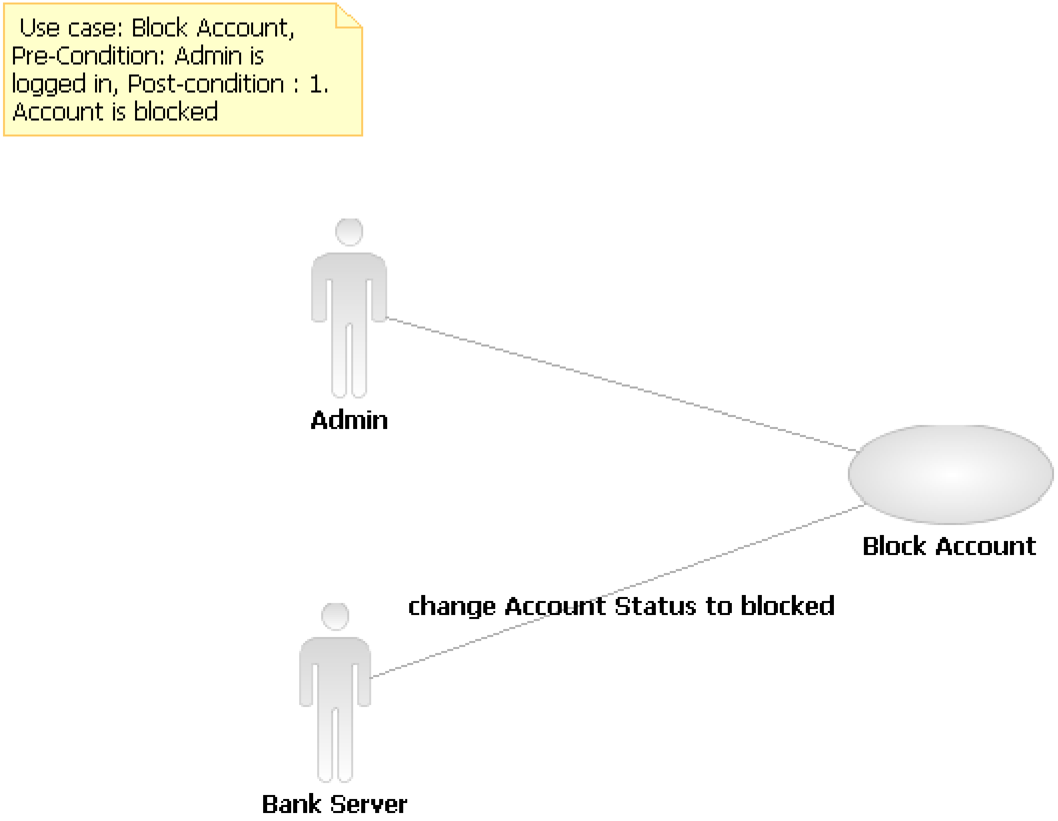

Use case: Block Account, Pre-Condition: Admin is logged in, Post-condition : 1. Account is blocked
Admin
Block Account
change Account Status to blocked
Bank Server

Figure-16: Block user account Use Case.

Description: Through this, Administrator can block one’s user account.

Specifications:

Actors: Administrator of the IBS system.

Pre-Conditions:

- Administrator logged in.

Post-Conditions:

- User Account is blocked.

Basic Flow:

- The Administrator will provide a *username* to be blocked.
- Server will block the *user account.*
- Administrator will be informed.

K) Name of Use Case: **Unblock User Account**

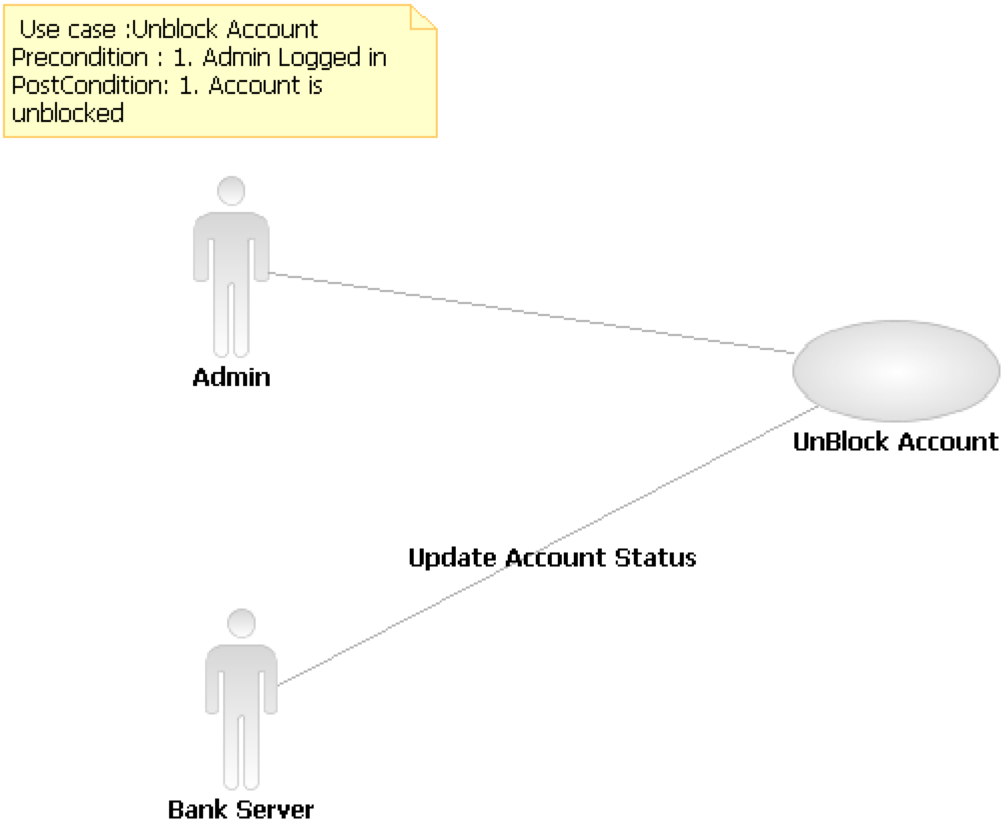

Use case :Unblock Account
Precondition : 1. Admin Logged in
PostCondition: 1. Account is unblocked
Admin
UnBlock Account
Update Account Status
Bank Server

Figure-17: UnBlock user account Use Case.

Description: Through this, Administrator can unblock one's blocked user account.

Specifications:

Actors: Administrator of the IBS system.

Pre-Conditions:

- Administrator logged in.

Post-Conditions:

- User Account is blocked.

Basic Flow:

- The Administrator will provide a *username* to be blocked.
- Server will block the *user account.*
- Administrator will be informed.

VII. Sequence Diagrams:

A sequence diagram (also called interaction diagram) is a UML construct of a Message Sequence Chart. It shows how processes operate one with another and in what order.

(i) Client Login to System

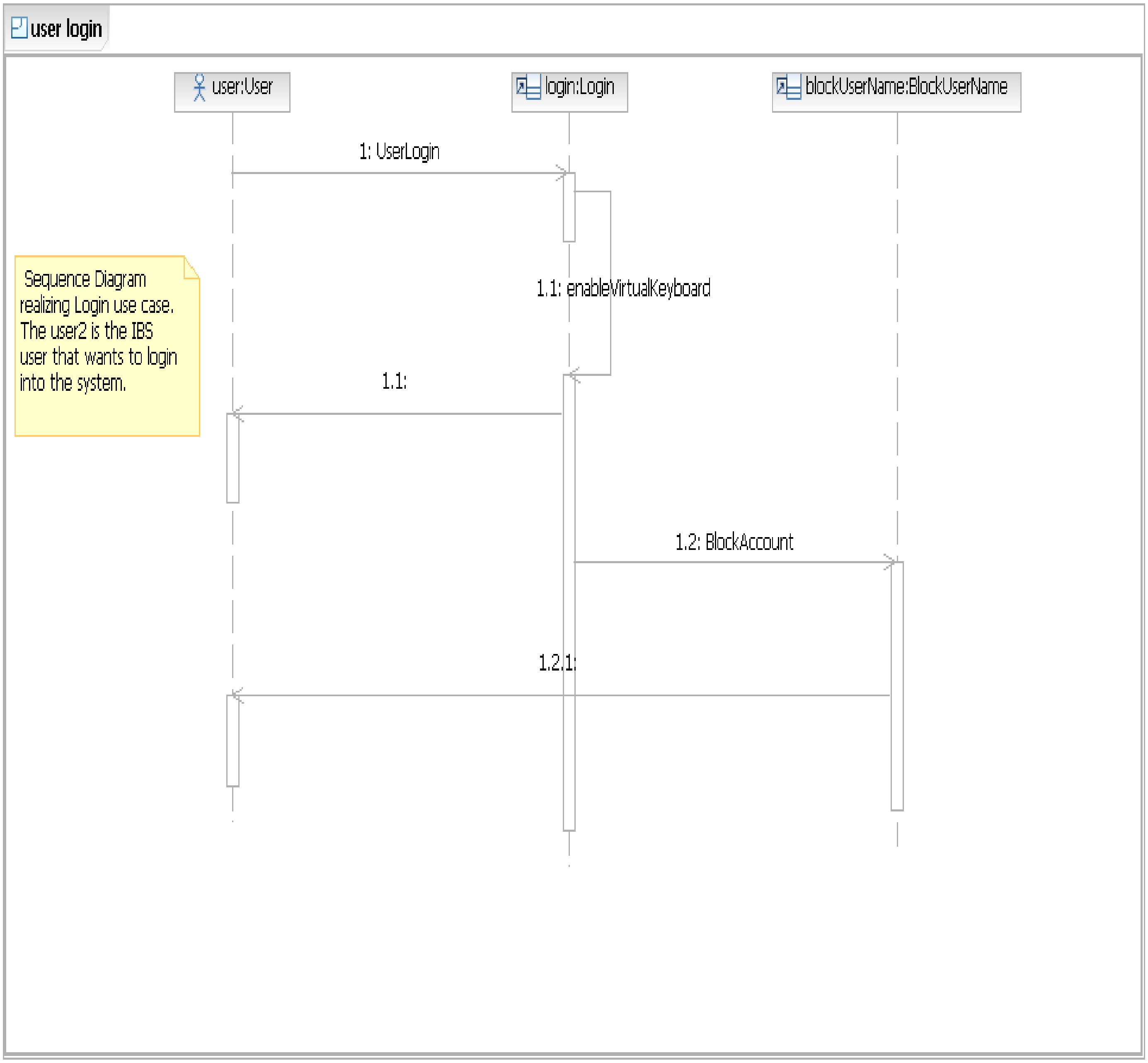

user login
user:User
login:Login
blockUserName:BlockUserName
1: UserLogin
1.1: enableVirtualKeyboard
1.1:
1.2: BlockAccount
1.2.1:
Sequence Diagram realizing Login use case. The user2 is the IBS user that wants to login into the system.

Figure-18: Client Login to System.

(ii) Account to Account money transfer

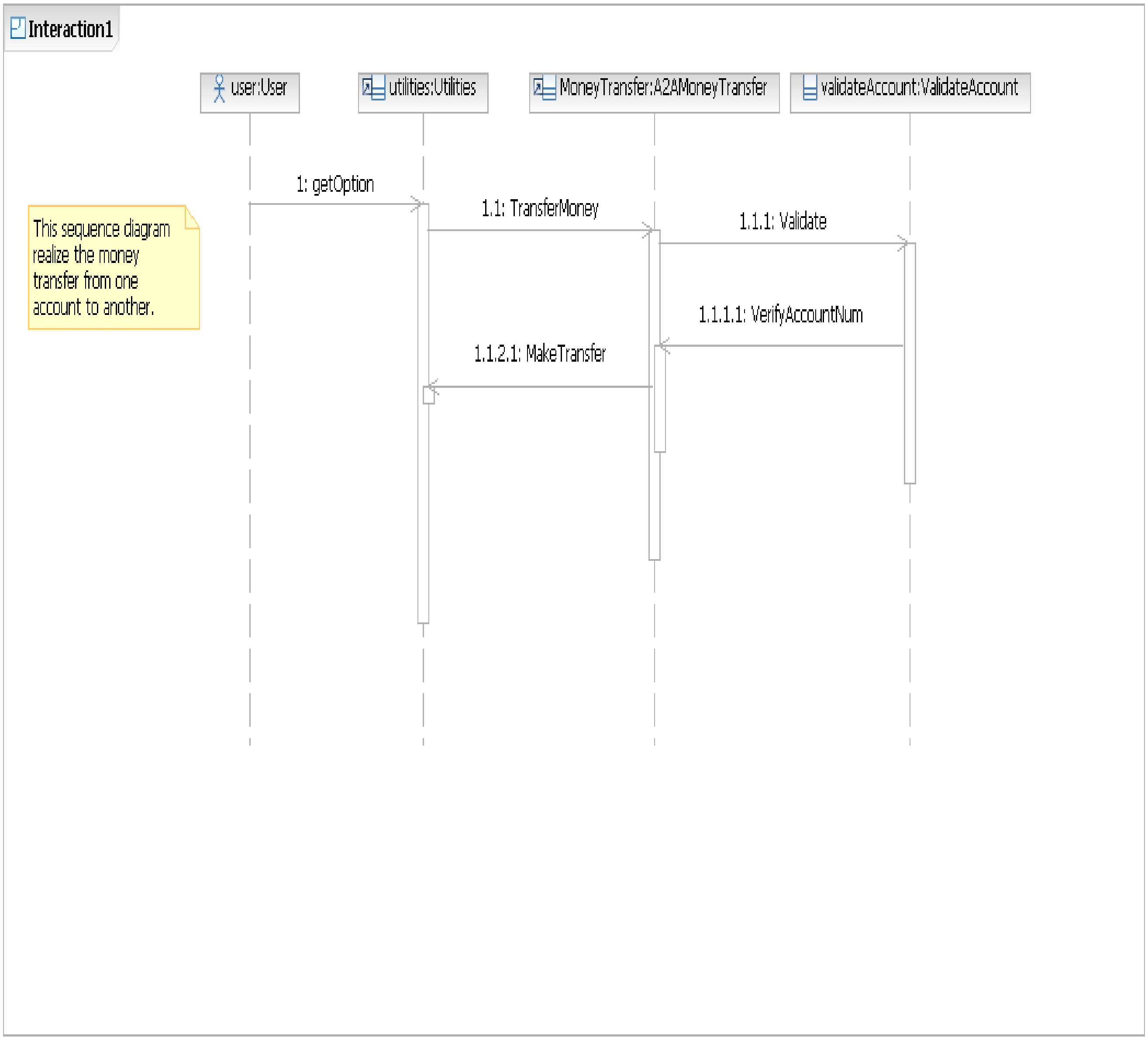

Interaction1
user:User
utilities:Utilities
MoneyTransfer:A2AMoneyTransfer
validateAccount:ValidateAccount
1: getOption
1.1: TransferMoney
1.1.1: Validate
1.1.1.1: VerifyAccountNum
1.1.2.1: MakeTransfer
This sequence diagram realize the money transfer from one account to another.

Figure-19: Account to Account money transfer

(iii) Map Account to UserID

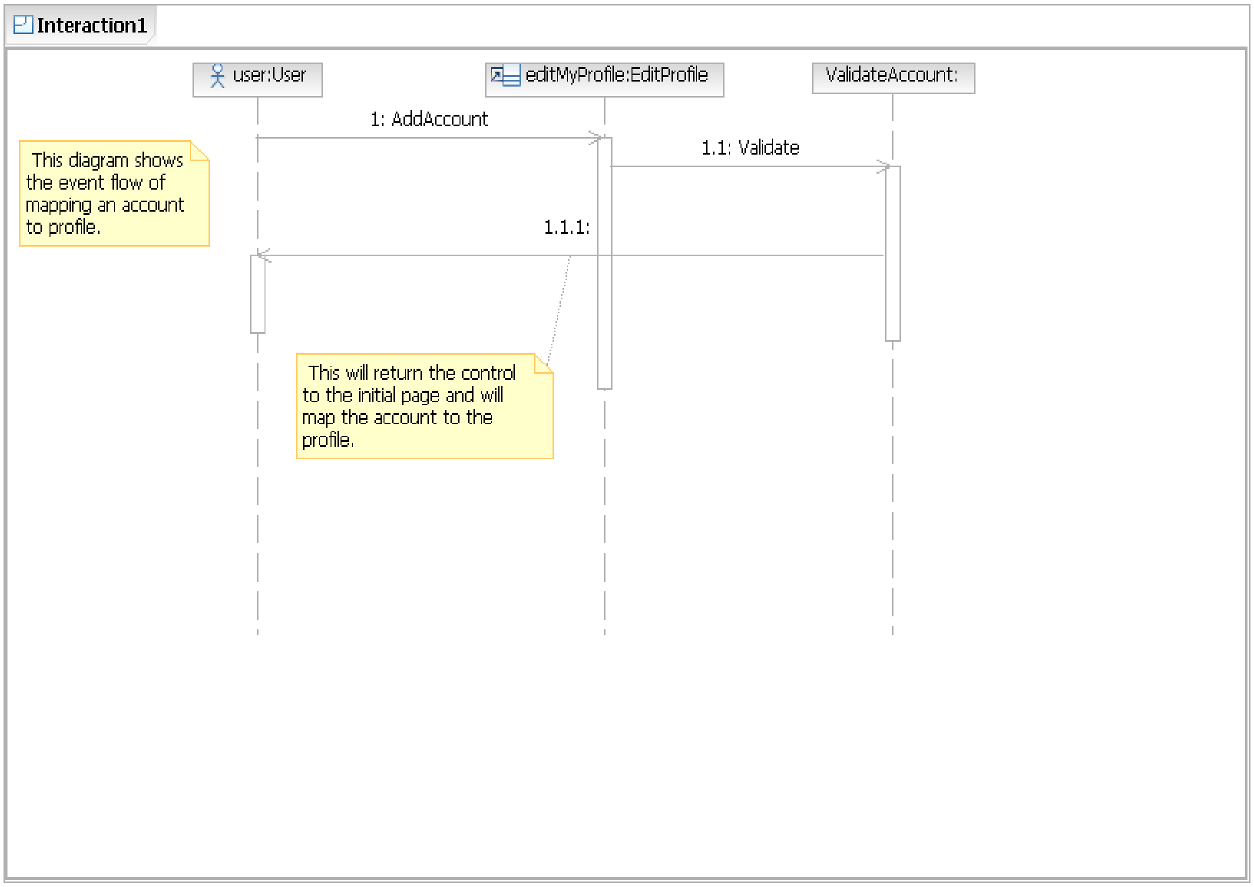

Interaction1
user:User
editMyProfile:EditProfile
ValidateAccount:
1: AddAccount
1.1: Validate
1.1.1:
This diagram shows the event flow of mapping an account to profile.
This will return the control to the initial page and will map the account to the profile.

Figure-20: Map account to userID.

(iv) Issue Standing Instruction to the system.

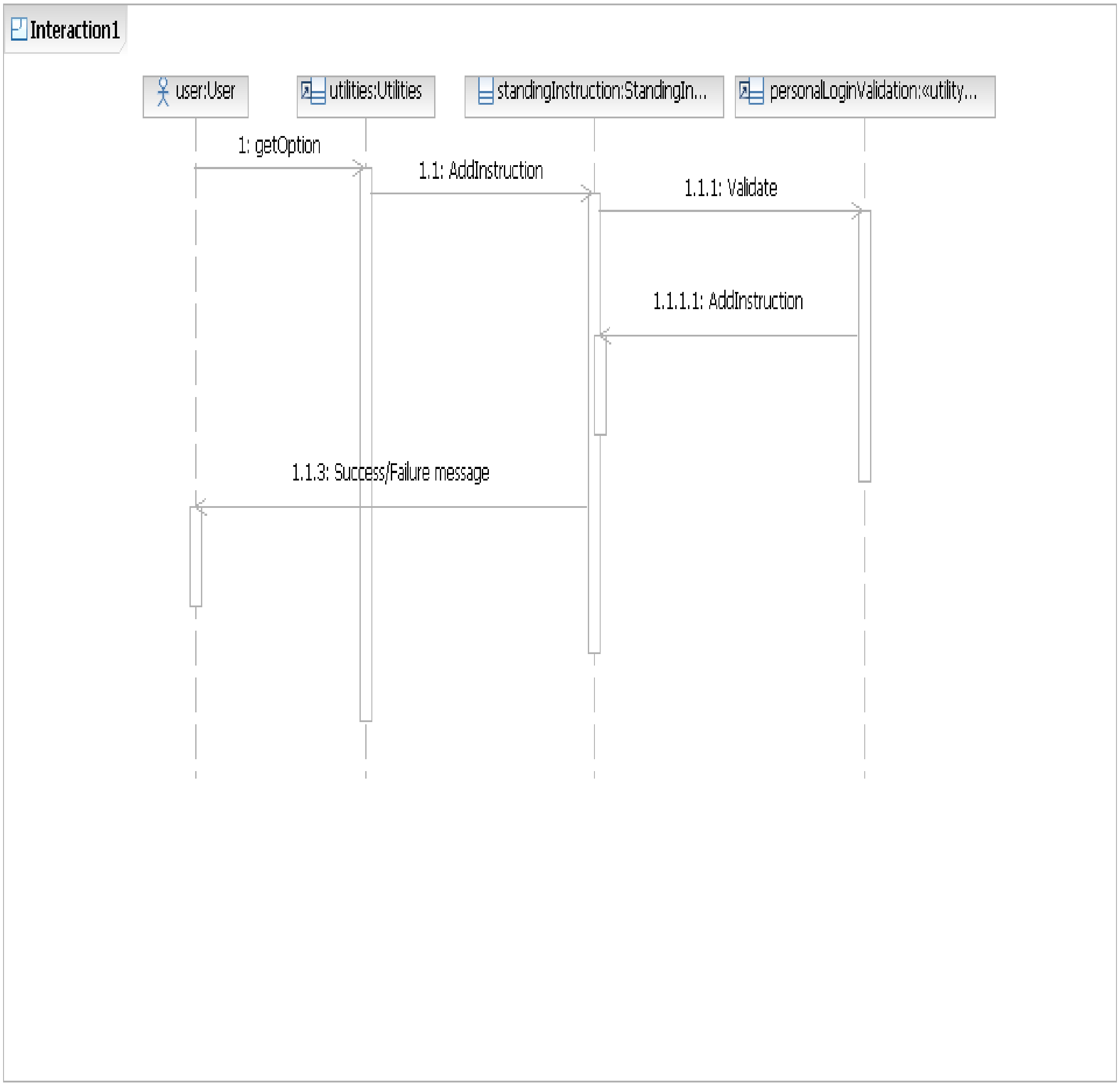


Figure-21: Issue Standing Instruction to the system.

(v) Payment of Bills

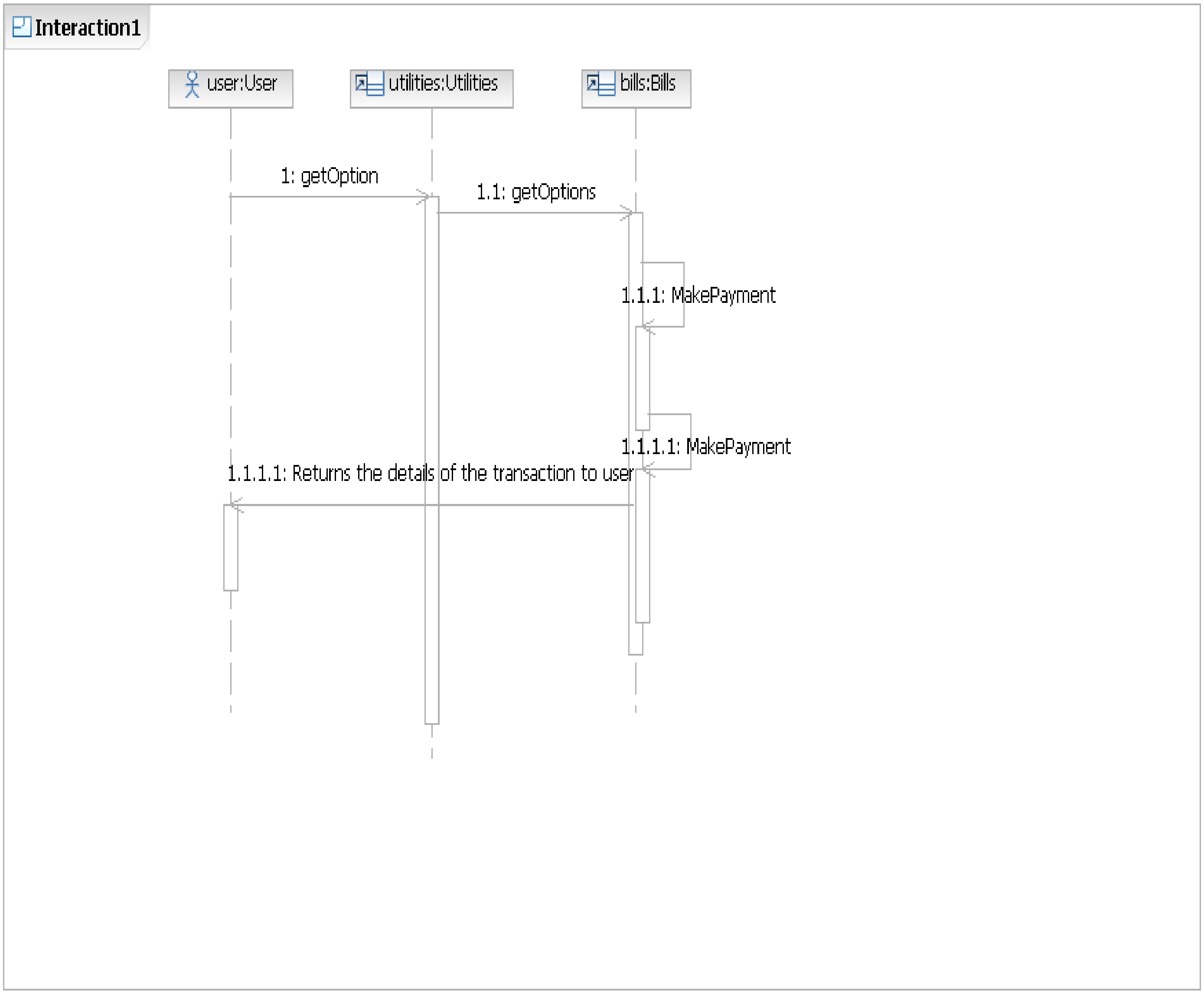


Figure-22: Payment of bills.

(vi) Edit User Profile

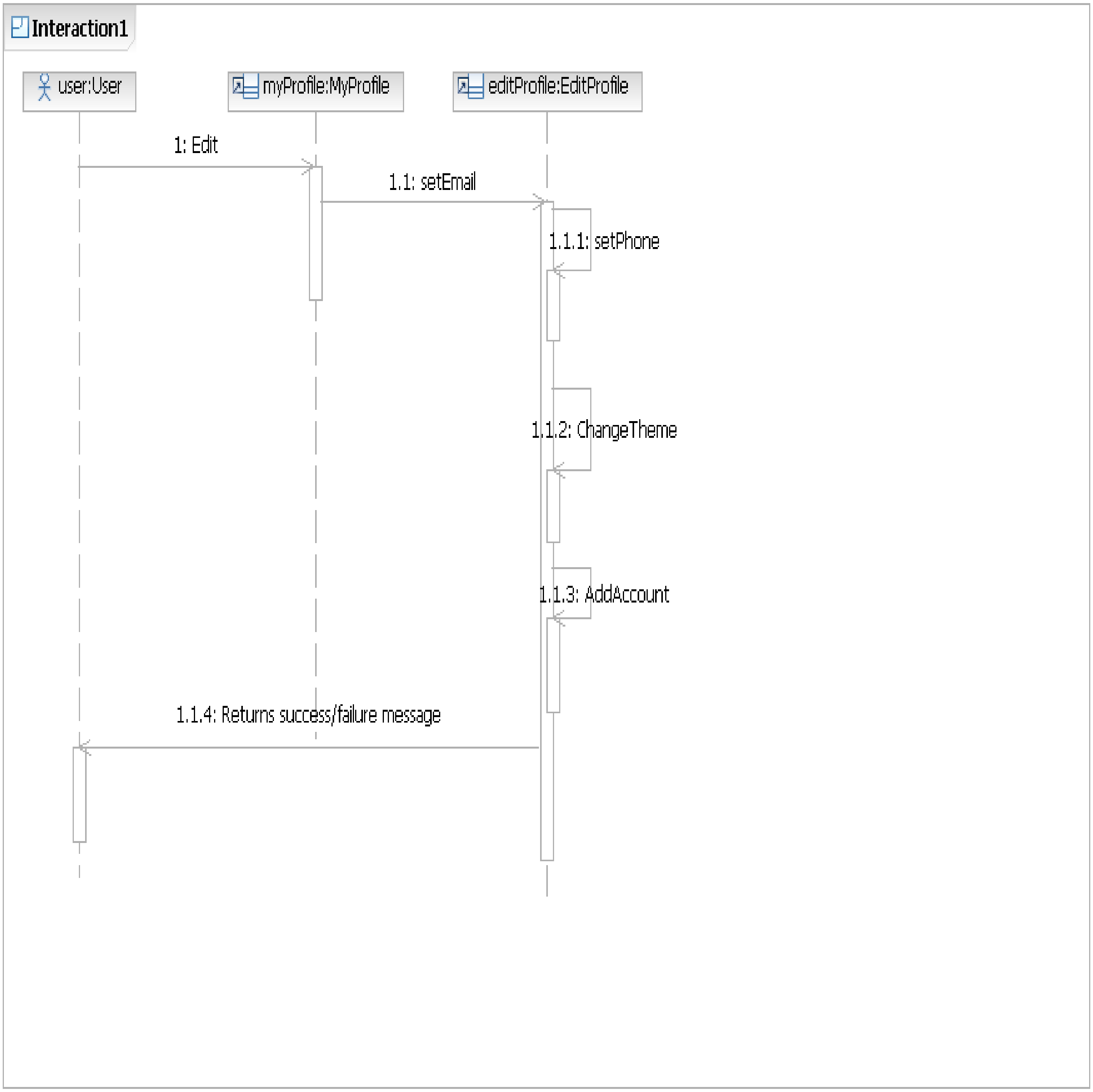


Figure-23: Edit user profile.

(vii) Block User Account

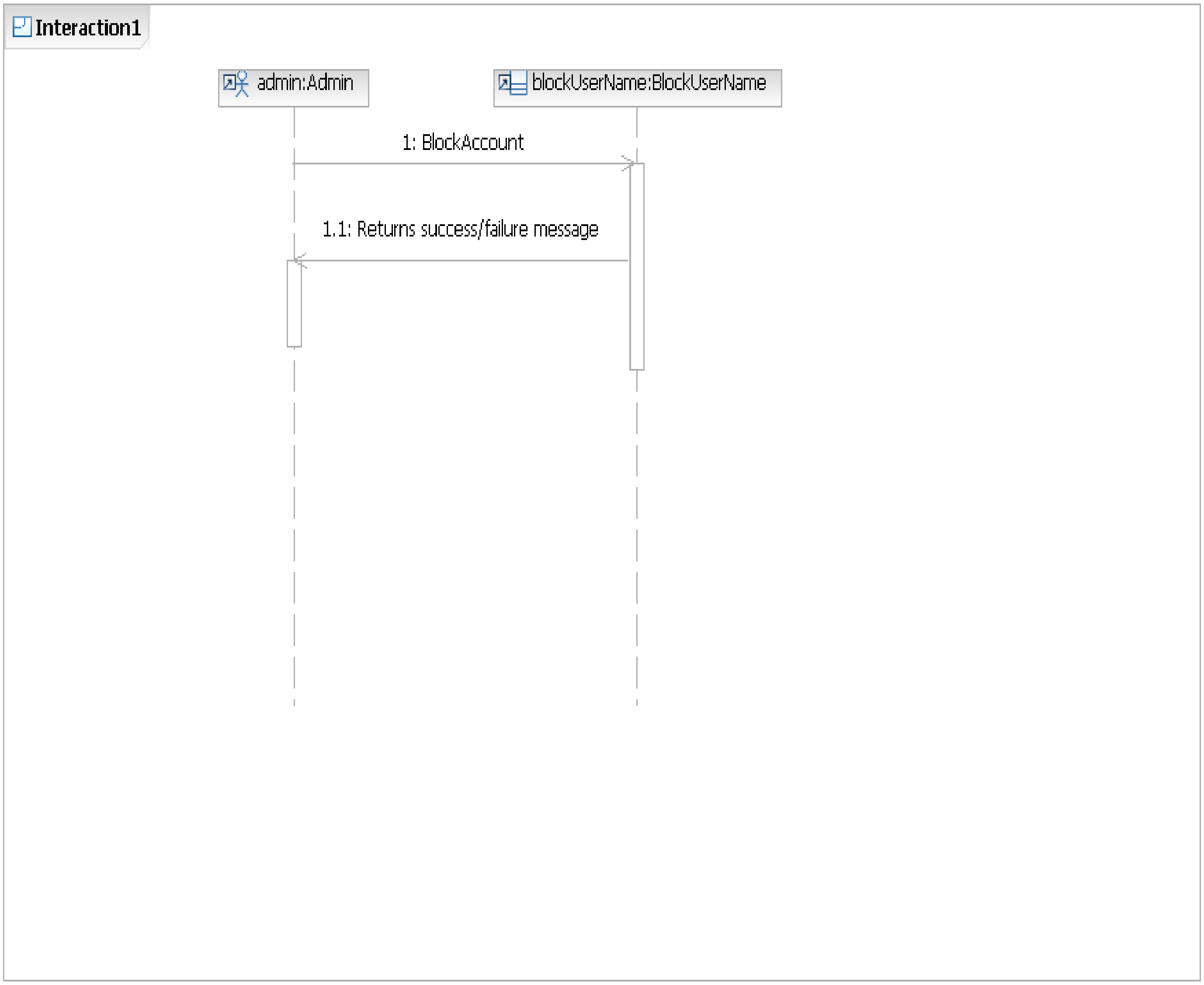

Interaction1
admin:Admin
blockUserName:BlockUserName
1: BlockAccount
1.1: Returns success/failure message

Figure-24: Block user account.

(viii) Unblock User Account

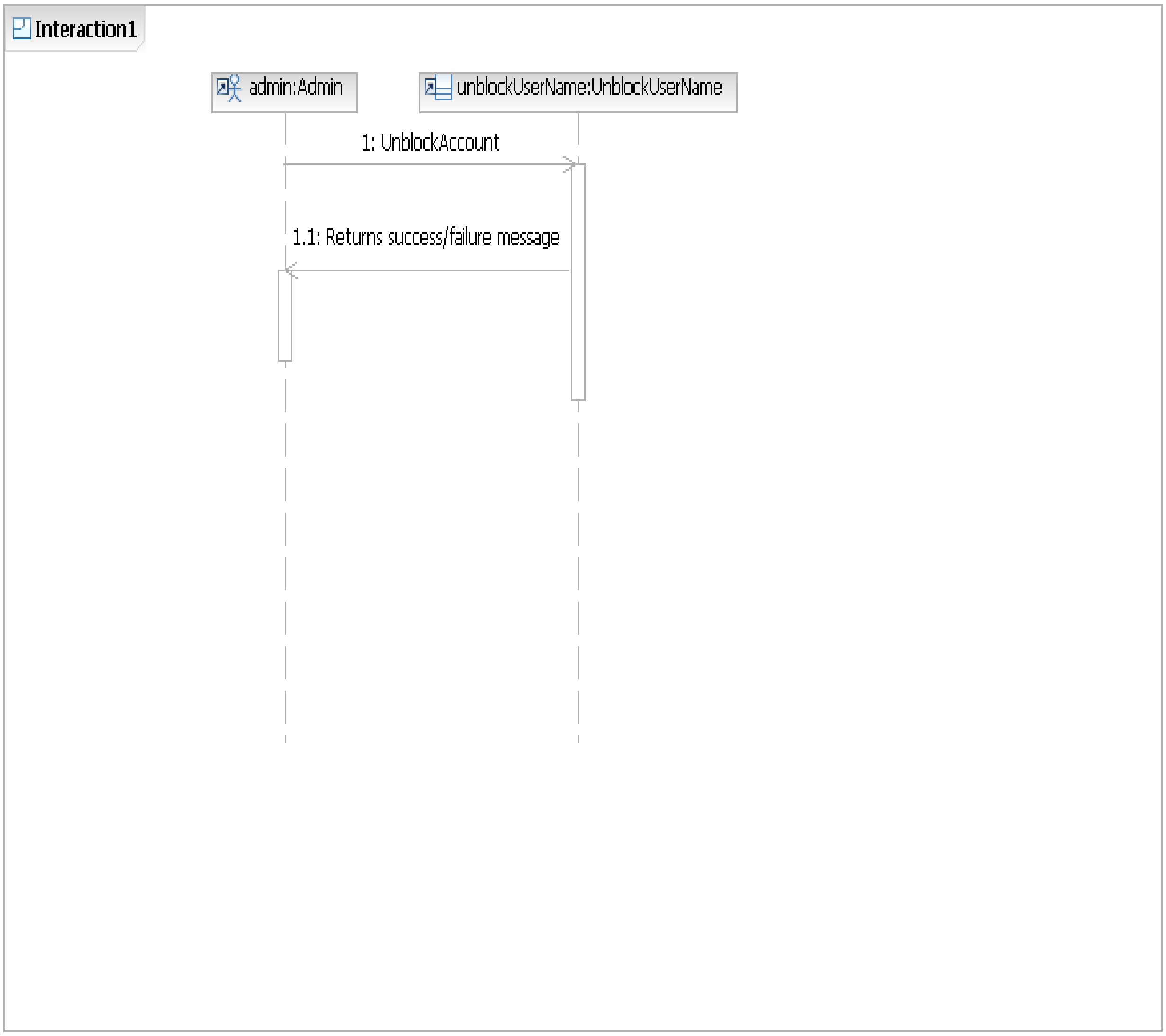

Interaction1
admin:Admin
unblockUserName:UnblockUserName
1: UnblockAccount
1.1: Returns success/failure message

Figure-25: Unblock user account

(ix) Requests

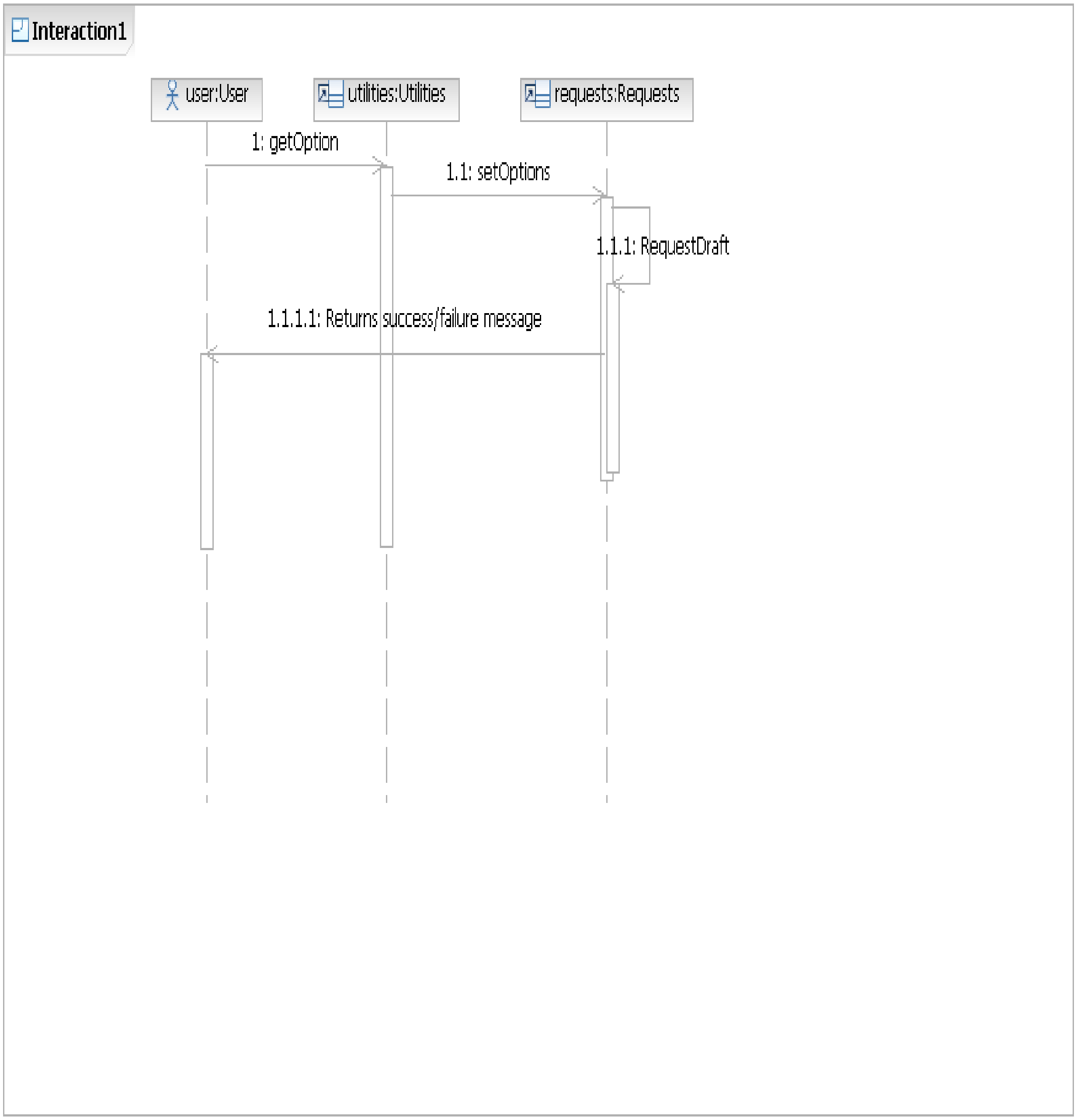


Figure-22: Requests.

II. Supplementary Requirements:

- **Have hours of operation that are 24 x 7 -** Because system can be an automated process, so it can stay open for 24 hours a day. If the base is now the entire world, staying open 24 hours a day becomes critical. System is required to be available 24x7 so UPS support must be on server site for at least 8 hours in case of power failure. System will remain inaccessible to users at 2:00 to 4:00 am for backup and maintenance purpose.

- **Make the existing Web site more dynamic in nature -** Many early Web implementations consisted of static HTML pages. This becomes very difficult to manage if the number of pages gets too large. An effective system should be largely dynamic taking advantage of technology that automates this process rather than relying on manual processes. Application should serve dynamic user based customized web pages to its clients from server.

- **Tie the existing Web site into existing enterprise systems –** Any existing Web site that relies on the manual duplication of data from another system is one that can be improved. Most of the business data in the world today exists in enterprise servers that can be connected to the Web servers to make this process far more effective.

- **Provide good performance and the ability to scale the server –** The Web
  Application Server should provide good performance and the ability to manage performance with techniques, such as support for caching, clustering, and load balancing.

- **Providing session management capability -** Web application developers should not spend valuable time worrying about how to maintain sessions within the application. The Web Application Server should provide these services.