**Backstreet Boys**

**Banking System**

**Software Design Document**

Glisca Maria-Coralia CEN 3.2A

Iofcea Albert Catalin CEN 3.2A

Mocioaca Gabriel CEN 3.2A

Petcu Alexandru-Gabriel CEN 3.2B

**TABLE OF CONTENTS**

1. INTRODUCTION 2

1.1 Purpose 2

1.2 Scope 2

1.3 Overview 2

1.4 Reference Material 2

1.5 Definitions and Acronyms 2

2. SYSTEM OVERVIEW 2

3. SYSTEM ARCHITECTURE 2

3.1 Architectural Design 2

3.2 Decomposition Description 3

3.3 Design Rationale 3

4. DATA DESIGN 3

4.1 Data Description 3

4.2 Data Dictionary 3

5. COMPONENT DESIGN 3

6. HUMAN INTERFACE DESIGN 4

6.1 Overview of User Interface 4

6.2 Screen Images 4

6.3 Screen Objects and Actions 4

7. REQUIREMENTS MATRIX 4

8. APPENDICES 4

**1. INTRODUCTION**

**1.1 Purpose**

This software design document describes the architecture and system design of our Bank Application.It also shows how the use cases detailed in the SRS will be implemented in the system using this design.

The primary audiences of this document are the software developers.

**1.2 Scope**

The purpose of the application is to reduce the need to go to the bank.

The system provides the access to the customer to create an account, deposit the cash from his account, also to view reports of all accounts present. The customers can access the banks website for viewing their Account details and perform the transactions on account as per their requirements.

**1.3 Overview**

The first part of this document(Part 1) is a short description of our Application, some references, acronyms and definitions.

The second part of this document(Part 2) is a short description of our application

The third part of this document(Part 3) are some diagrams about the functionality of our program(Architectural Design ,Decomposition Description Design Rationale) .

The fourth part of this document(Part 4) contains the data description and the dictionary of our app data.

The fifth part of this document(Part 5) is a closer look at what each design component does in a more systematic way.

The sixth part of this document(Part 6) is a description about the user experience.

The seventh part of this document(Part 7) is cross reference that traces components and data structures to the requirements in our document.

The eighth part of this document(Part 8) is providing supporting details that could aid in the understanding of the Software Design Document.

**1.4 Reference Material**

ASP.NET :

[ASP.NET | Open-source web framework for .NET](https://dotnet.microsoft.com/apps/aspnet)

[ASP.NET Documentation](https://docs.microsoft.com/en-us/aspnet/)

[ASP.NET MVC Pattern](https://dotnet.microsoft.com/apps/aspnet/mvc)

[C# docs - get started, tutorials, reference.](https://docs.microsoft.com/en-us/dotnet/csharp/)

[.NET Framework documentation](https://docs.microsoft.com/en-us/dotnet/framework/)

Databases :

<https://docs.microsoft.com/en-us/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>

<https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15>

Banking :

<https://en.wikipedia.org/wiki/Online_banking>

<https://en.wikipedia.org/wiki/Bank>

Framework CSS :

<https://devdocs.io/css/>

<https://getbootstrap.com/>

Other References :

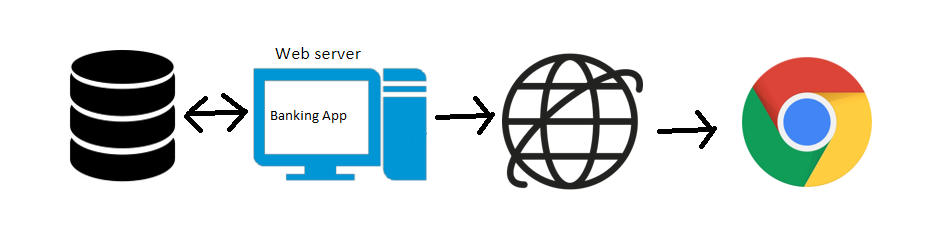
<https://docs.google.com/>

**1.5 Definitions and Acronyms**

|  |  |
| --- | --- |
| User | Someone who interacts with the application |
| Stakeholder | Any person who has interaction with the system who is not a developer |
| Database | A collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system |
| Performance requirement | A system/software system requirement specifying a performance characteristic that a system/software system or system/software component must possess; for example, speed, accuracy, and frequency |
| SRS | Software Requirement Specifications |
| BAMS | Bank Account Management System |
| SQL | Structured Query Language |
| ASP | ​Active Server Pages |
| Server | A central computer (server) which provides services such as file storage, printing, and communications in a network computing system |
| Software requirement | (1) A software capability needed by a user to solve a problem to achieve an objective; (2) A software capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document |
| System | A composite of equipment, skills, and techniques capable of performing or supporting an operational role or both. A complete system includes all equipment, related facilities, material, software, services and personnel required for its operation and support to the degree that it can be considered a self-sufficient item in its intended operational environment. |
| User class | A group of users for a system who have similar characteristics and requirements for the system. |
| RAT | Rational |

**2. SYSTEM OVERVIEW**

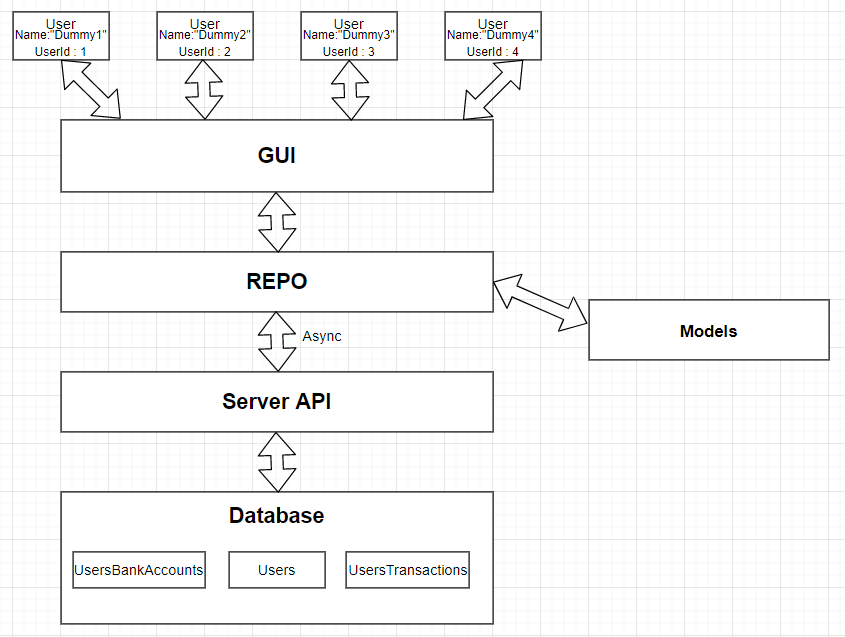
The Banking System project is a new, self-contained system intended for web application. This system is intended for bank clients to transfer funds from their bank account to another bank account with a smartphone just with the help of the internet, from anywhere to everywhere. With the help of the app, Banking users can transfer funds, pay bills, check account balance and view your recent transactions. Below is a diagram of the Banking System which illustrates the interactions between the server and clients



**3. SYSTEM ARCHITECTURE**

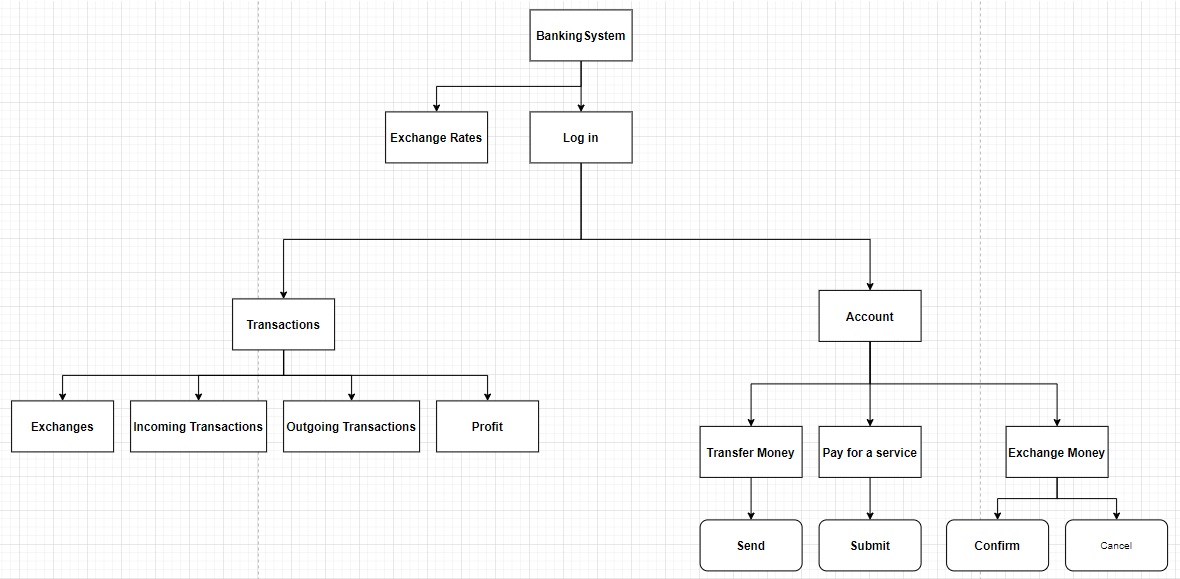
**3.1 Architectural Design**

The block diagram below shows the principal parts of the system and their interactions.

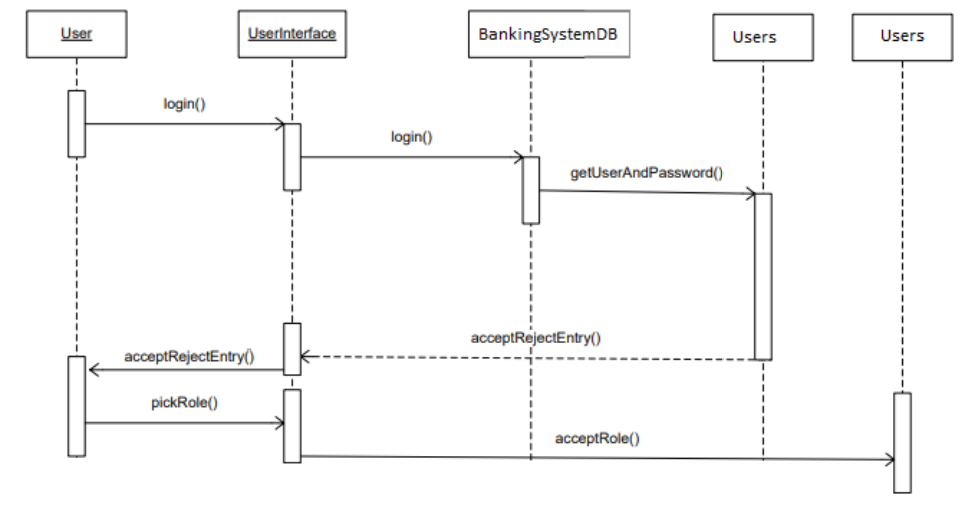
****

**3.2 Decomposition Description**

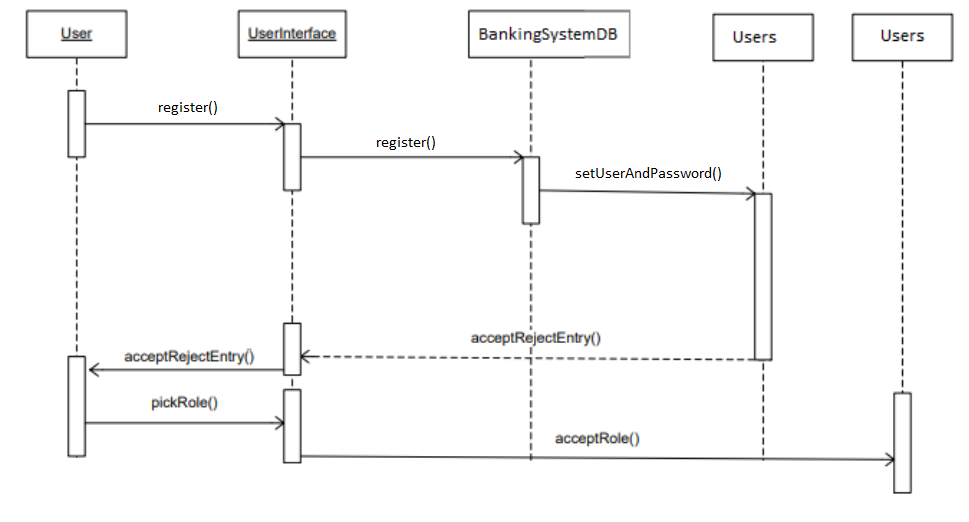
**Decompositional Tree**



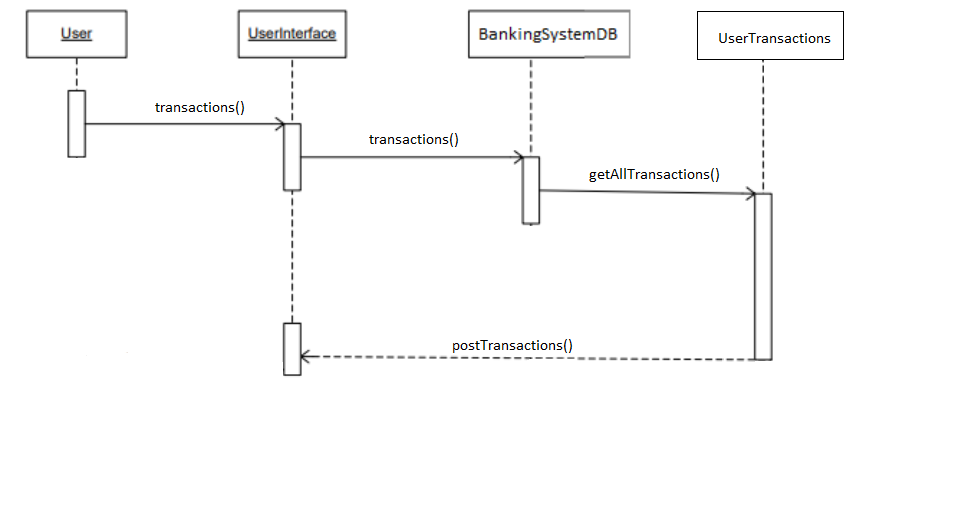
**3.2.1 Log In (C1) Data Flow**



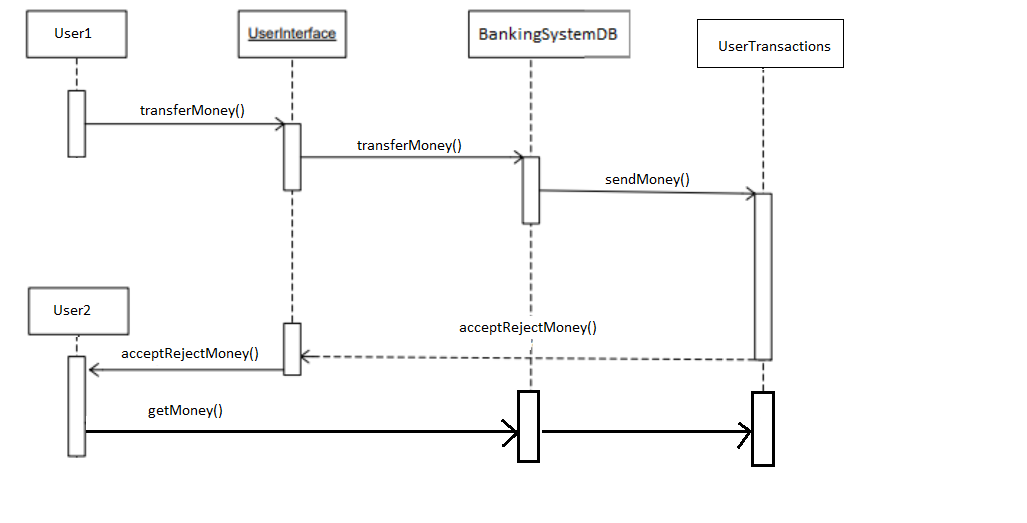
3.2.2 Register **(C2)** DataFlow



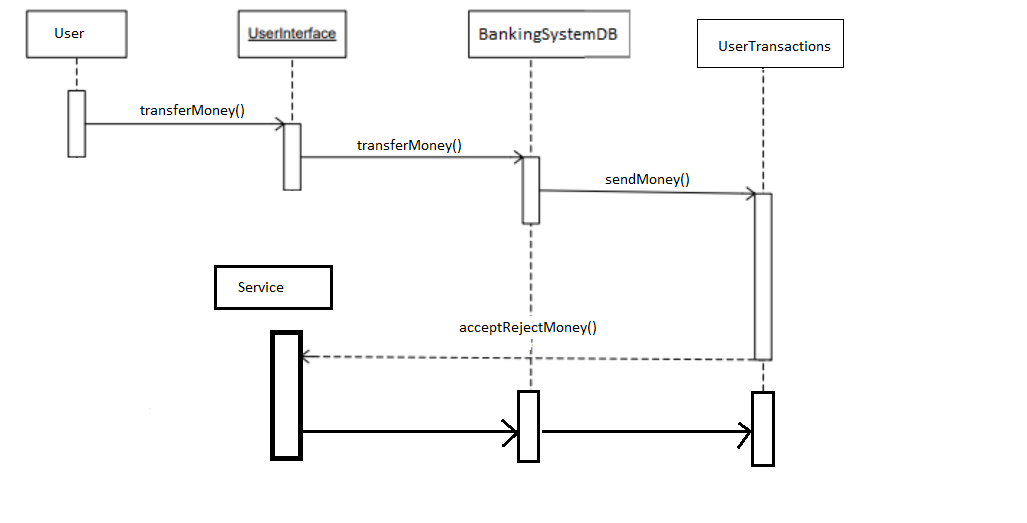
3.2.3 Transactions **(C3)** DataFlow



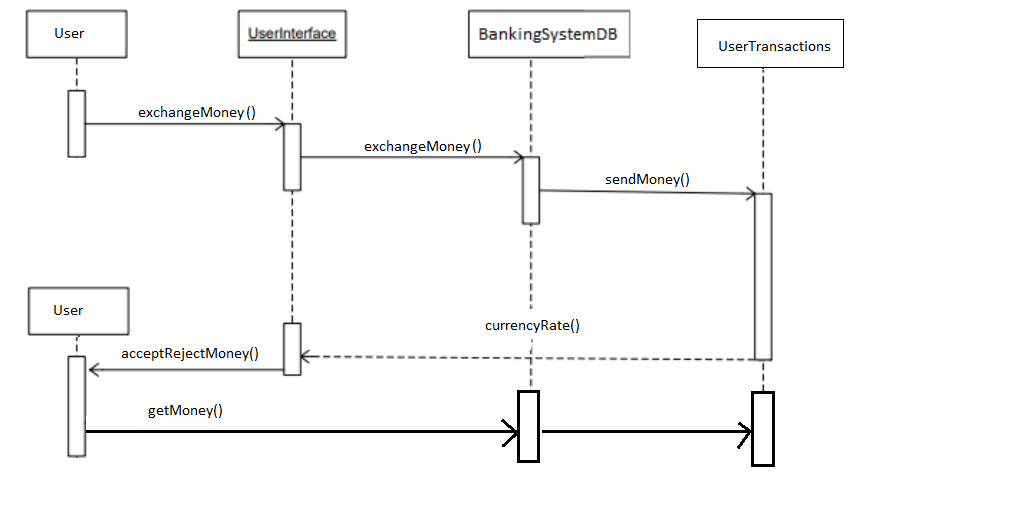
3.2.4 Transfer Money **(C4)** DataFlow



3.2.5 Pay for a Service **(C5)** DataFlow



3.2.6 Exchange Money **(C6)** DataFlow



**3.3 Design Rationale**

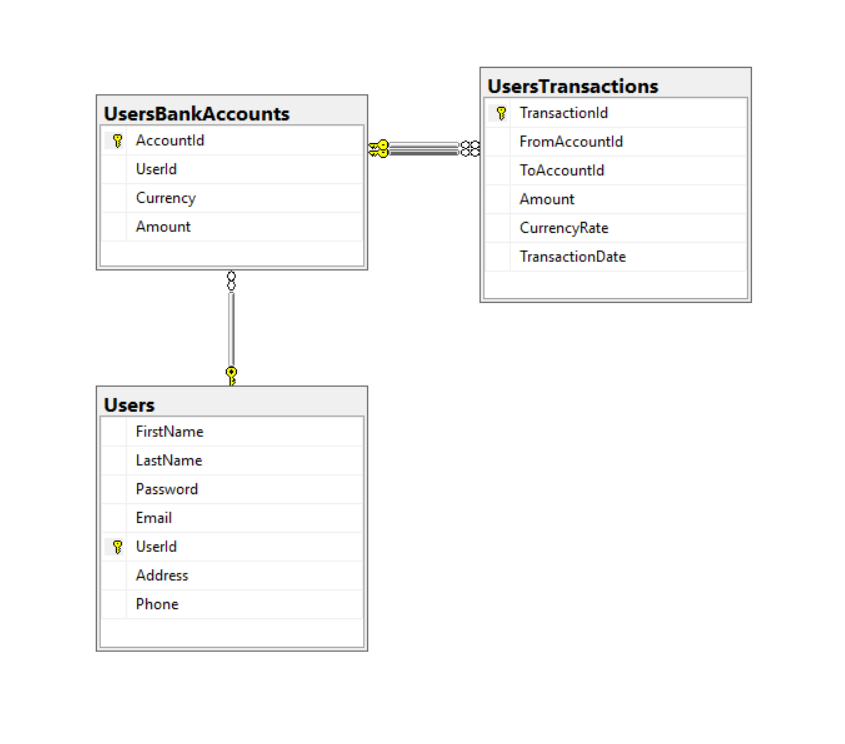
We decided upon the aforedescribed architecture because it allows our user interface to only check a single source of truth, which is the repository. The repository is the one who then can choose to get data: either API calls to the server-side application or local models.

**4. DATA DESIGN**

**4.1 Data Description**

There is one database in this application. The database consists of 3 tables: “Users”, “UsersBankAccounts” and “UsersTransactions”.

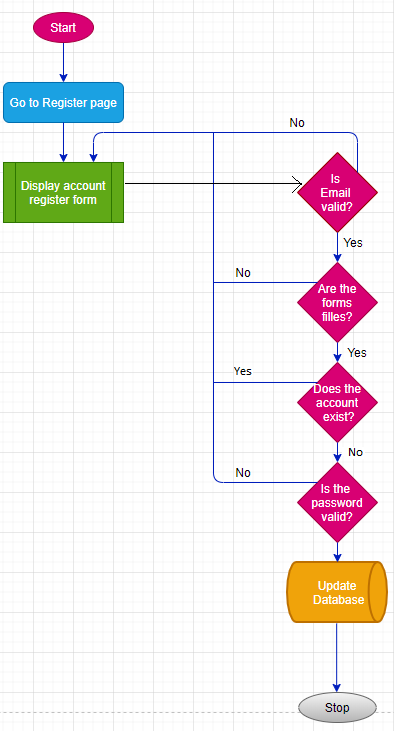
* The “Users” table stores the clients’ personal data;
* The “UsersBankAccounts” table stores the amount of money a user has in a certain currency he has an account in;
* The “UsersTransactions” table stores clients transactions.

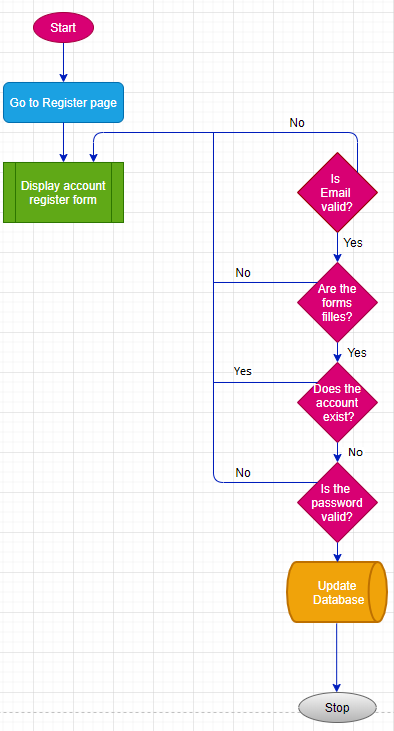


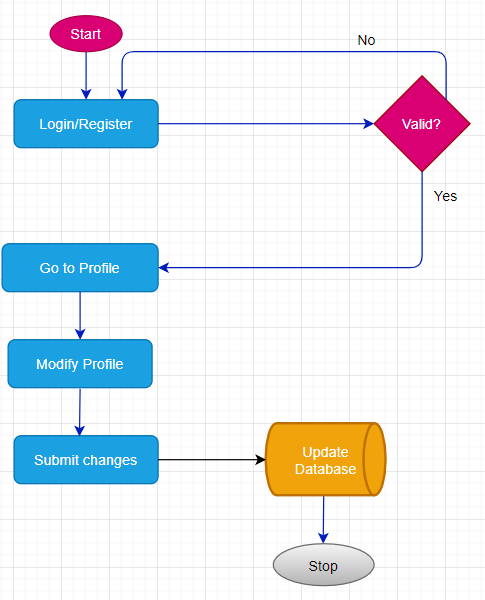
**4.2 Data Dictionary**

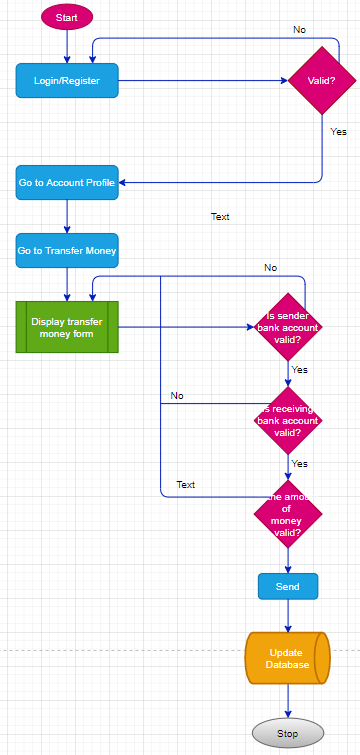
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Field** | **Type** | **Null** | **Default** |
| **Users** | UserId | int | No | 1 |
|  | FirstName | nvarchar(32) | No |  |
|  | LastName | nvarchar(32) | No |  |
|  | Password | nvarchar(32) | No |  |
|  | Email | nvarchar(64) | No |  |
|  | Address | nvarchar(32) | No |  |
|  | Phone | int | NULL | NULL |
|  |  |  |  |  |
| **UserBankAccounts** | AccountId | int | No | 1 |
|  | UserId | int | No |  |
|  | Currency | nvarchar(3) | No |  |
|  | Amount | decimal(10,2) | No |  |
|  |  |  |  |  |
| **UserTransactions** | TransactionId | int | No | 1 |
|  | FromAccountId | int | No |  |
|  | ToAccountId | int | No |  |
|  | Amount | decimal(10,2) | No |  |
|  | CurrencyRate | decimal(10,2) | No |  |
|  | TransactionDate | datetime | No |  |

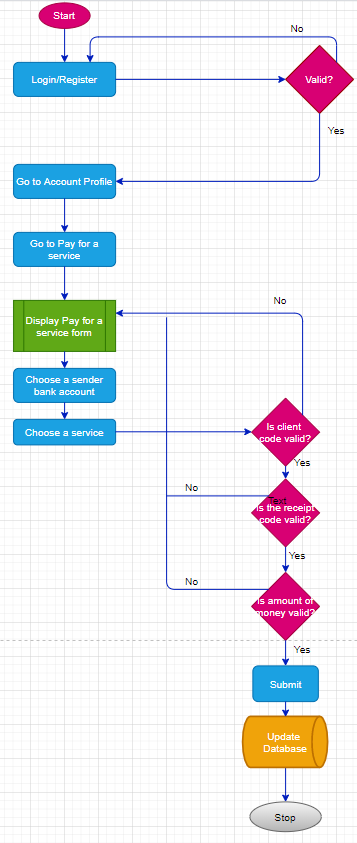
**5. COMPONENT DESIGN**

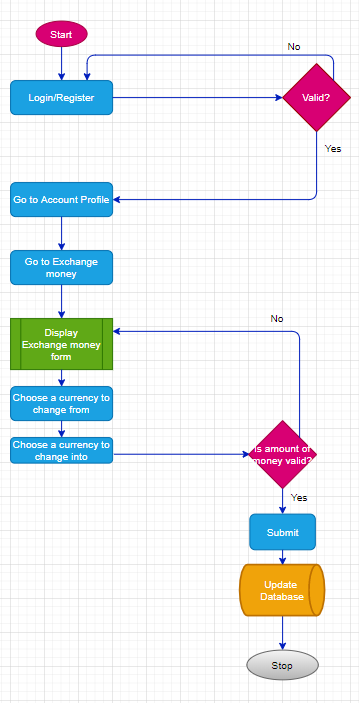








****

****

**6. HUMAN INTERFACE DESIGN**

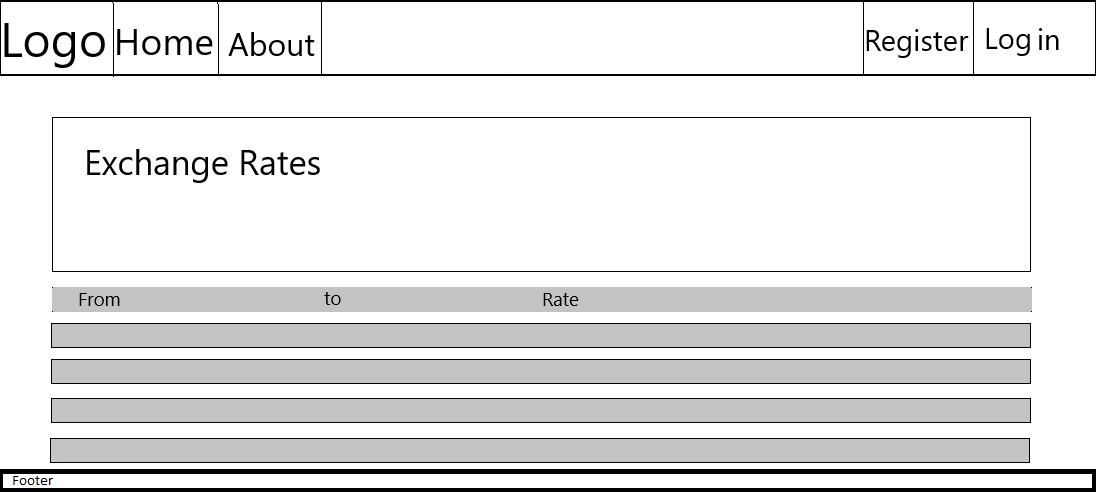
**6.1 Overview of User Interface**

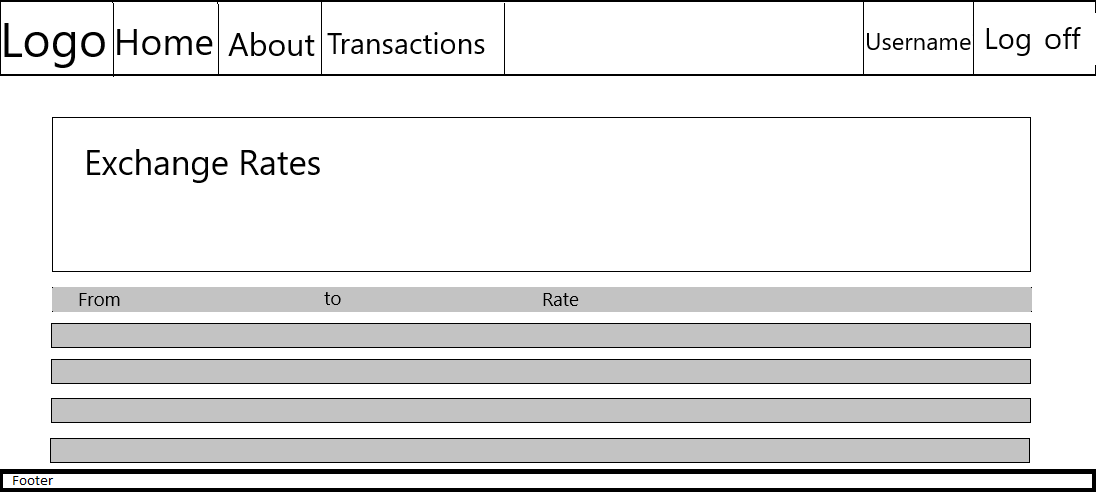
A first-time user of the web application should see a home landing page where random information is displayed as well a top navigation bar. The user is then able to go to a register/login page by clicking on the items in the left part of the navigation bar.

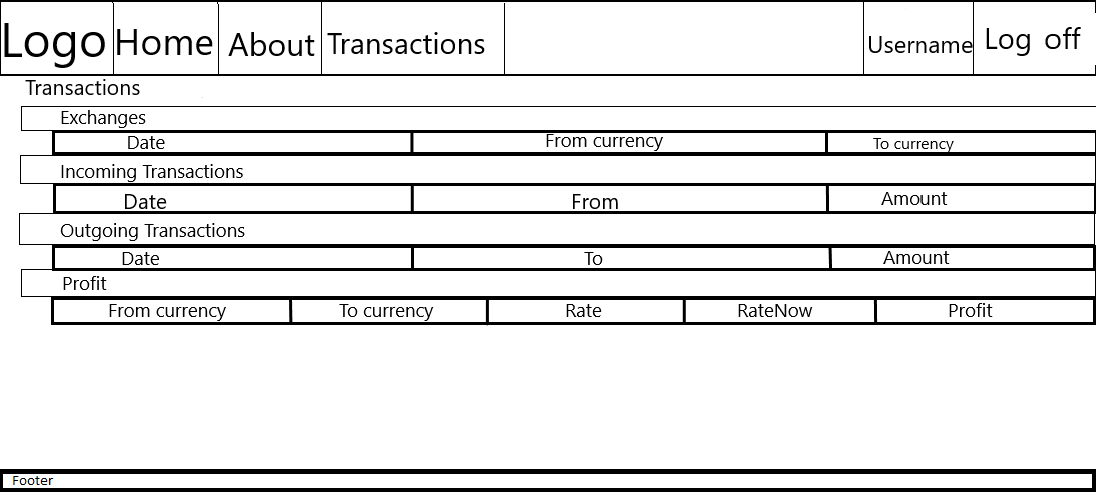
Once registered/logged in, a user can view their transaction history by clicking on the Transactions item on the navigation bar or they can access their account page by clicking on their username.

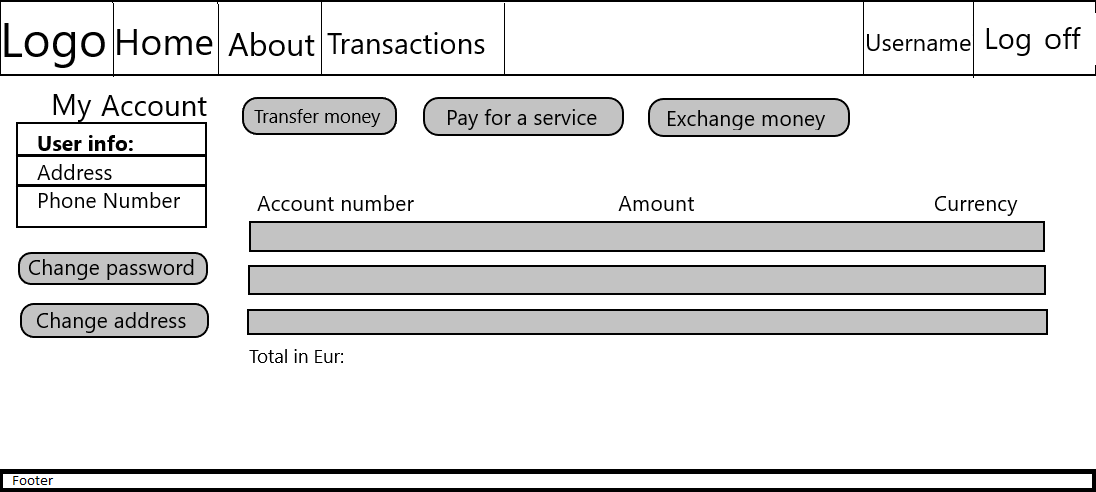
On their account page they can view/change account information like password, address or they can use the banking features like transferring money, pay for a service or exchange money. Each of these have their own page.

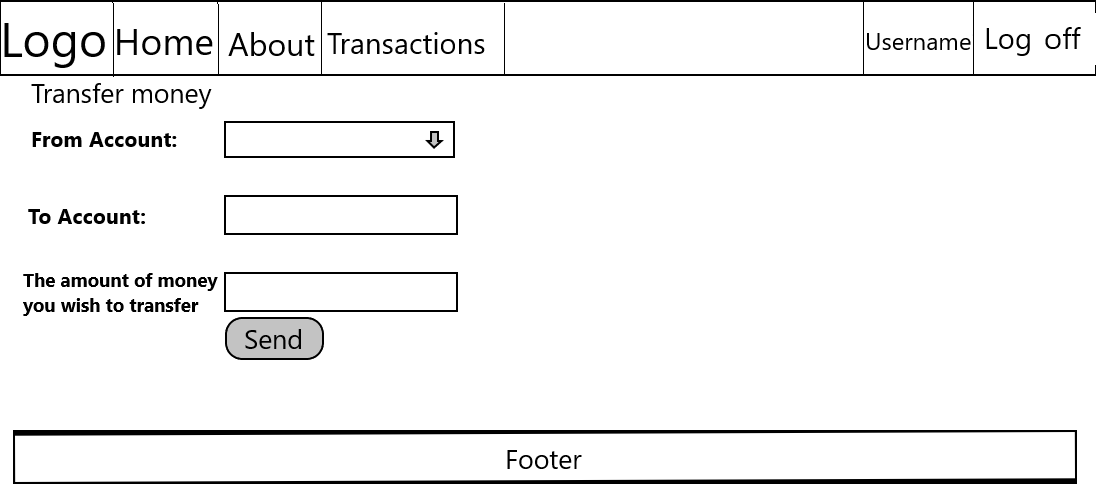
**6.2 Screen Images**

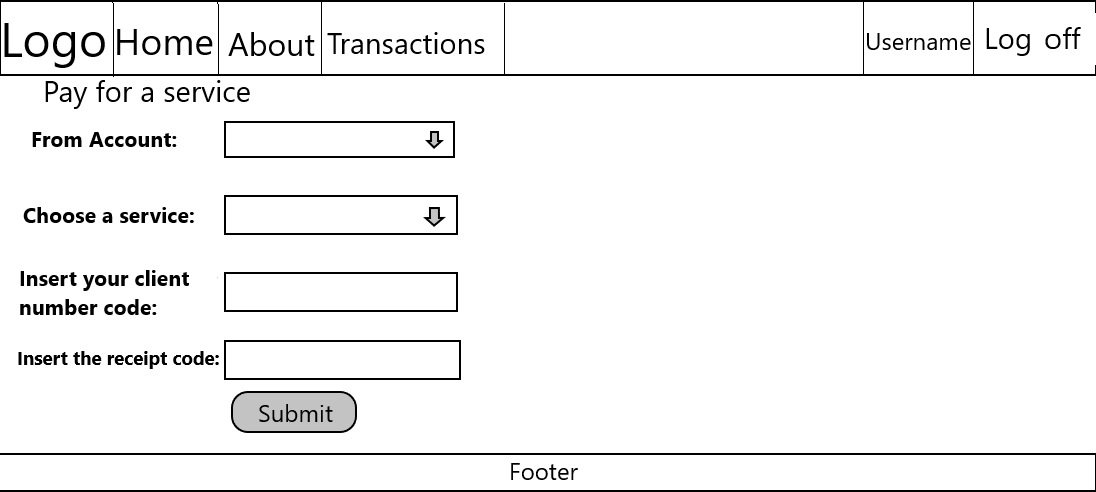
****

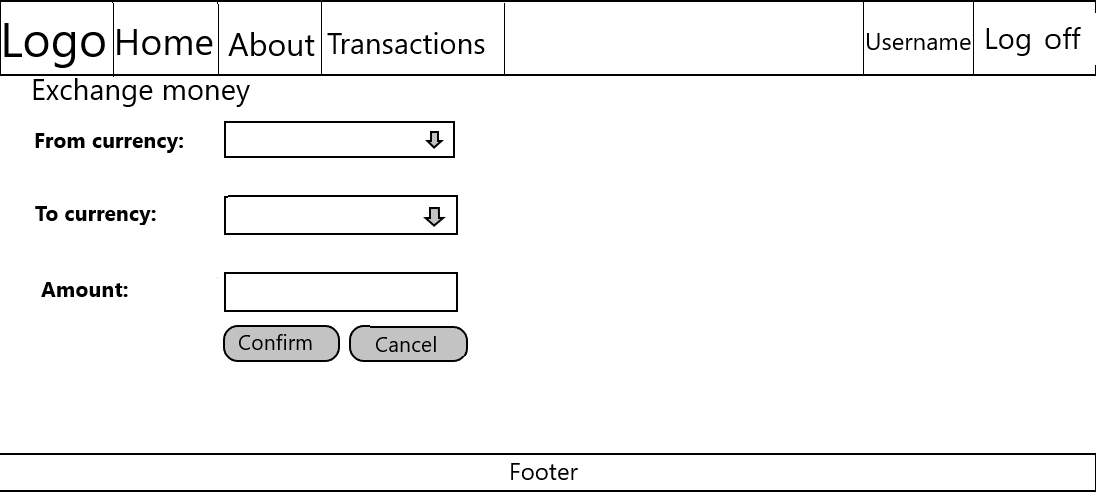
****

****

****

****

****

****

**6.3 Screen Objects and Actions**

1. Navbar :
   * ***“Logo”:*** Image of the logo
   * ***“Home”:*** button will redirect to the Home page
   * ***“About”:*** button will redirect to About page
   * ***“Register”:*** To create a new account, the client must press the ‘Register’ button from the navigation bar or from the *‘Login’* page.
   * ***“Login”:*** To log in, the client must press the Login Button from the navigation bar.
   * ***“Transactions”:*** To see all of their transactions, the client must press the Transactions button. This will redirect to a new page where their transactions will be shown.
   * ***“Username”:*** In this field, the username of the client will be shown.
   * ***“Log off”:*** This button will log the client out of the application.
2. Home page :
   * ***“Exchange Rates” :*** In this page all exchange rates will be presented. Exchange rates will be updated every hour.
3. Transactions Page :

The Transactions page will consist of a collapsible composed of more sections. The ***‘Exchanges’*** section where the user will be able to see all of the money exchanges he made from one currency to another. The ***‘Incoming Transactions’*** section where the user will see all of the transactions made to his accounts.The ***‘Outgoing Transactions’*** sections where the user will see all of the transactions made from one of his accounts to other users/services. The ***‘Profit’*** section where the user will see if the exchanges they made were profitable or not.

1. MyAccount page :
   * ***“Transfer Money”*** button will redirect to Transfer Money page
   * ***“Pay for a service”*** button will redirect to Pay for a service page
   * ***“Exchange Money”*** button will redirect to
   * ***“Change password”*** User can change his password
   * ***“Change address”***  User can change his address
   * ***“User info”*** will show info about the client like address, phone number.
   * ***Accounts list*** will show a list of different accounts that this user has.
2. Transfer Money page: The user will choose which one of his existing accounts he will use to transfer money from with the ‘From Account’ dropdown list. The user will choose which account he will transfer money to within the ‘To Account’ field. It checks to see if the said account exists. ‘The amount of money you wish to transfer’ field where the user enters the sum he wishes to transfer. It is required to have only numerical values. The ‘Send’ button to finish the transfer.
3. Pay for a service page : The user will choose which one of his existing accounts he will use to transfer money from with the ‘From Account’ dropdown list. The user will choose which service provider he will send money to within the ‘Choose a service’ dropdown list.The user will have to provide his client number of the chosen service into the “Insert your client number code:” field. This field cannot be left empty.The user will have to provide the number of the receipt of the chosen service into the “Insert the receipt code:” field. This field cannot be left empty and it will be checked to only have numerical elements.The ‘Submit’ button which will submit the payment the user makes towards the specified service.
4. Exchange Money page : User can exchange money from a currency to another .The user will choose which currency he will exchange into another with the ‘From Currency’ dropdown list.The user will choose to which currency he will exchange money with the ‘To Currency’ dropdown list.The ‘Amount of money’ field where the user enters the sum he wishes to exchange.The ‘Confirm’ button to finish the operation.‘Cancel’ button to cancel the exchange. ‘See your profit’ button that redirects the user to the ‘Transactions’ page that shows them if they made the right choice by exchanging at that time, or not, by comparing the rise and fall of a certain currency’s value.

**7. REQUIREMENTS MATRIX**

|  |  |  |  |
| --- | --- | --- | --- |
| **REQUIREMENTS TRACEABILITY MATRIX** | | | |
| **Traceability #** | **Requirement ID** | **Components ID** | **Data structures** |
| **1** | **FR1** | **C2** | **Users** |
| **2** | **FR2** | **C1** | **Users** |
| **3** | **FR3** | **C2** | **Users** |
| **4** | **FR4** | **C2** | **Users** |
| **5** | **FR5** | **C2** | **Users** |
| **6** | **FR6** | **C4** | **UserTransactions** |
| **6** | **FR7** | **C5** | **UserTransactions** |
| **8** | **FR8** | **C6** | **UserTransactions** |
| **9** | **FR9** | **C3** | **UserTransactions** |
| **10** | **FR10** | **C1** | **Users** |
| **11** | **FR11** | **C1** | **Users** |
| **12** | **FR12** | **C1** | **Users** |
| **13** | **PRQ1** | - | - |
| **14** | **DC1** | - | - |
| **15** | **SSA1** | - | - |
| **16** | **SSA2** | - | - |
| **17** | **SSA3** | - | - |
| **18** | **SSA4** | - | - |
| **19** | **SSA5** | - | - |
| **20** | **SSA6** | - | - |