**Software Requirements**

**Specification**

Banking System

Glisca Maria Coralia CEN 3.2A

Iofcea Albert Cătălin CEN 3.2A

Mocioaca Marius Gabriel CEN3.2A

Petcu Alexandru Gabriel CEN 3.2B

**Table of Contents**

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

1.1Purpose ............................................................................................. 3

1.2 Scope ............................................................................................... 3

1.3 Definitions, acronyms, and abbreviations ......................................... 4

1.4 References ....................................................................................... 6 2. Overall description .............................................................................. 8 2.1 Product perspective .......................................................................... 8 2.2 Product functions .............................................................................. 8 2.3 User characteristics .......................................................................... 9 2.4 Constraints ....................................................................................... 9 2.5 Assumptions and dependencies ....................................................... 9 2.6 Apportioning of requirements ........................................................... 9 3. Specific requirements .......................................................................10 3.1.1 User interfaces ............................................................................ 10 3.1.2 Hardware interfaces .....................................................................13 3.1.3 Software interfaces ......................................................................13 3.1.4 Communications interfaces ..........................................................13 3.2 Functional requirements ................................................................. 13 3.2.1 User Class 1 - The Client ............................................................ 14 3.2.3 User Class 2 - Administrator ........................................................ 17 3.3 Performance requirements ............................................................. 18 3.4 Design constraints .......................................................................... 18 3.5 Software system attributes ............................................................. 18 4. Prioritization and Release Plan ........................................................ 21 4.1 Choice of prioritization method ....................................................... 21

**1. Introduction**

This section gives a scope description and overview of everything included in this SRS document. Also, the purpose for this document is described and a list of abbreviations and definitions is provided.

**1.1 Purpose**

The purpose of this document is to provide a detailed description of the functionality of a Banking system . It will explain the interface , the system constraints and how user friendly the application was made .

This SRS also serves as an input for the designing and modeling of the Banking system. The document also covers hardware, software and various other technical dependencies.

This document is intended to be used by the client to understand how the application was made.

**1.2 Scope**

The “Bank Account Management System” project is a model Internet Banking Site. This site enables the customers to perform the basic banking transactions by sitting at their office or at homes through PC or laptop. 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.

The primary aim of this “Bank Account Management System” is to provide an improved design methodology, which envisages the future expansion, and modification, which is necessary for a core sector like banking. This necessitates the design to be expandable and modifiable and so a modular approach is used in developing the application software. Anybody who is an Account holder in this bank can become a member of the Bank Account . Management System. He has to fill out a form with his personal details and Account Number.

* **Contex**

In the early 90's everybody had a bank account but they had no possibility to check they're account balance so they always had to go to a bank or an atm(Interminable queues and lazy staff).

The “Bank Account Management System” is a step further for a more modern society .

There are a number of clear reasons why we need this system :

* The astonishing increase of population in the world (so the queues are larger)
* The need to transfer money (you need to visit the bank for an transfer)
* The digitization of information (tons of paper)

The Banking system will provide major services like : Viewing the information of the banking account , change client address , change account password , making transfers to other accounts , making payments to other services , view transactions .

This project also covers various features like online registration of the users, reports generation, as well as managing all the data.

The Banking system is composed of two main components: a Client-side application which will run on web browsers and the Server-side application which contain Database, web server.

**1.3 Definitions, acronyms, and abbreviations**

|  |  |
| --- | --- |
| 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 |

**1.4 References**

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 Overview**

The remainder of this document includes three chapters and appendices. The second one provides an overview of the system functionality and system interaction with other systems. This chapter introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product.

The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences.

The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen.

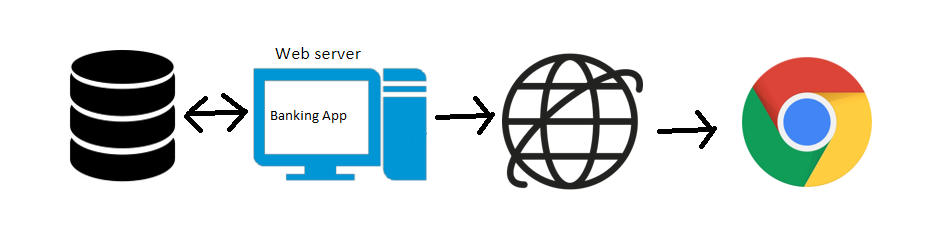
The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

**2. Overall description**

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

**2.1 Product perspective**

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

**2.2 Product functions**

The data system consists of two main parts: a central database management system for the uploading, storage, and management of data, and a client application to allow users access and interact with the data.

The Product functions are more or less the same as described in the product perspective. The functions of the system include the system providing different types of services based on the type of users. All users have the following functions:

* View account details
* Change client address
* Change password
* Transfers to other accounts
* Making pays to other services
* View transactions
* Each client can have different types of accounts , depending of the money currency he choses
* Exchange money from one currency to another

**2.3 User characteristics**

The users of the app will be the clients of the bank who want to manage their bank accounts. They will be able to register an account , login to that account, edit their profile. Users can transfer money to another bank account, can pay service providers and view all their transactions.

**2.4 Constraints**

The Internet connection is a constraint for the application. Since the application fetches data from the database over the Internet, it is crucial that there is an Internet connection for the application to function.

**2.5 Assumptions and dependencies**

* + The users have sufficient knowledge of computers
  + The users know the English language, as the user interface will be provided in English
  + The users should have a computer or phone with Internet connection
  + The App can access the Bank database

**2.6 Apportioning of requirements**

**Operating Environment**

**Software specs:**

* Operating System : Windows Server 2019
* Database : SQL Server
* Programing Languages : C# , HTML , CSS
* Server Web : Microsoft IIS

**Hardware Specs:**

Minimum requirements:

* RAM: 1GB
* CPU: Intel Pentium 4
* Internet connection

**Recommended specifications**

* RAM: 4GB
* CPU: Intel Core i3
* Internet connection

The database should allow a big amount of data. The application will be designed in such a way that it can be run on the latest versions of the most used operating systems and browsers. The .NET technology shall be used to implement the web application and SQL Server shall be used to manage the database. The user should have a browser installed on his system.

**3. Specific requirements**

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

**3.1 External interface Requirements**

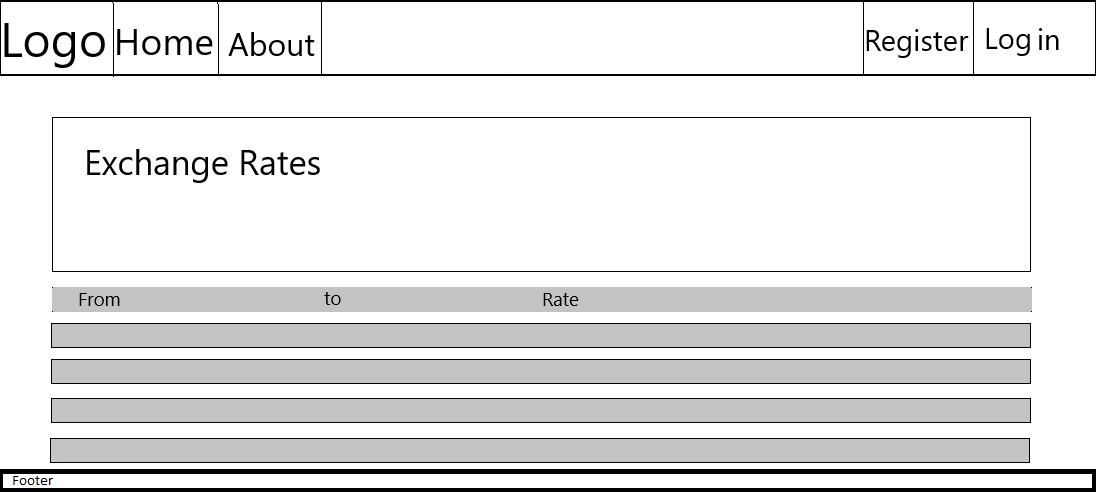
This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

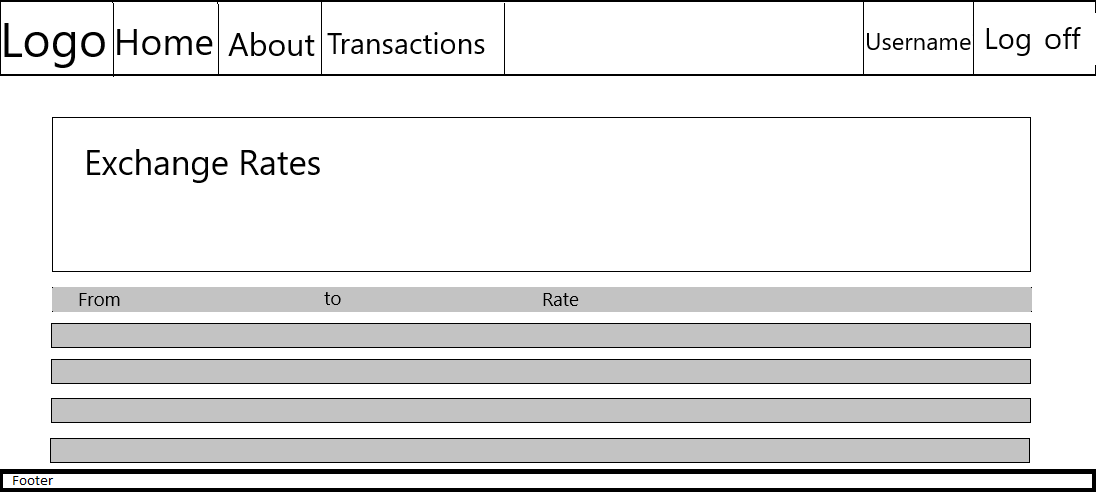
**3.1.1 User interfaces**

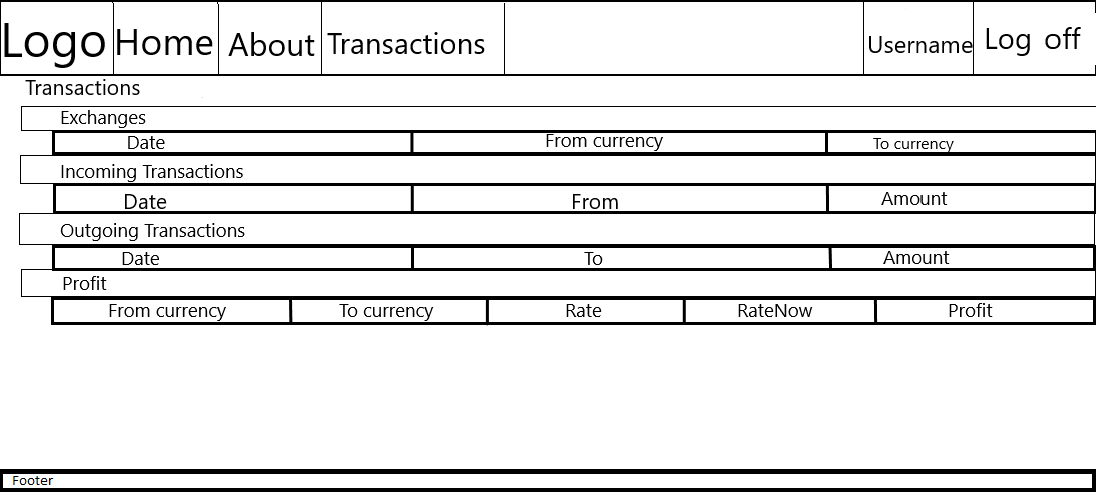
A first-time user of the web application should see a home landing page where random information is displayed as well a top navigation bar. The user is then able to go to a register/login page by clicking on the items in the left part of the navigation bar.

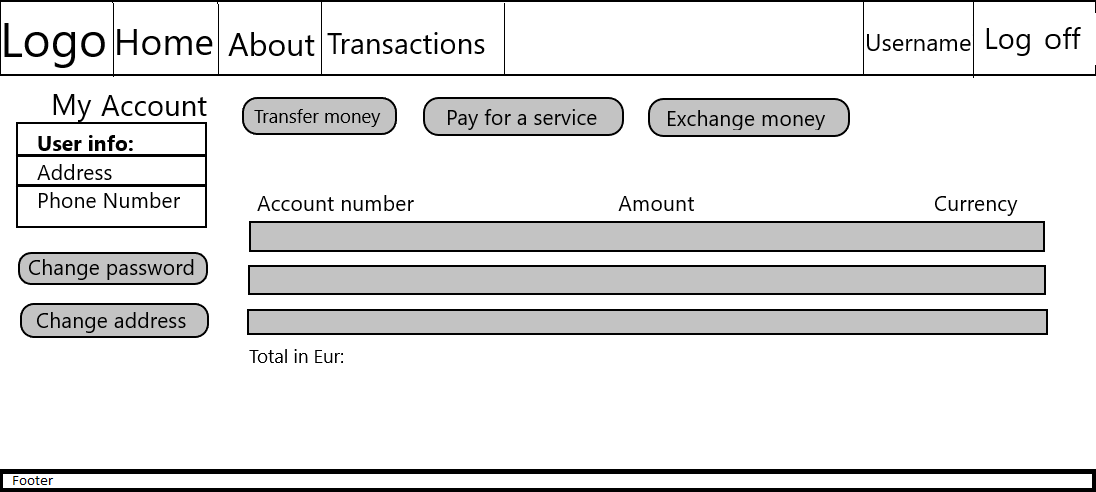
Once registered/logged in, a user can view their transaction history by clicking on the Transactions item on the navigation bar or they can access their account page by clicking on their username.

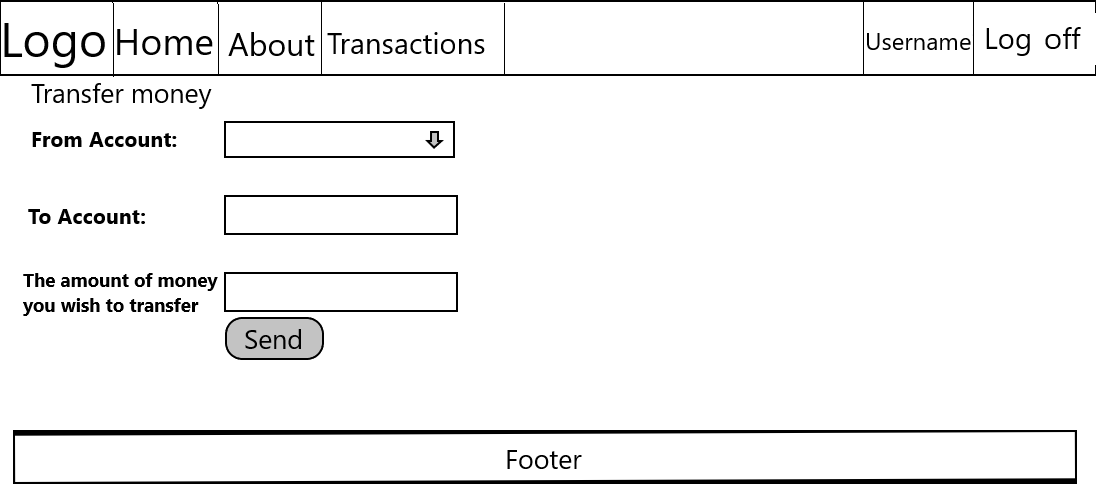
On their account page they can view/change account information like password, address or they can use the banking features like transferring money, pay for a service or exchange money. Each of these have their own page.

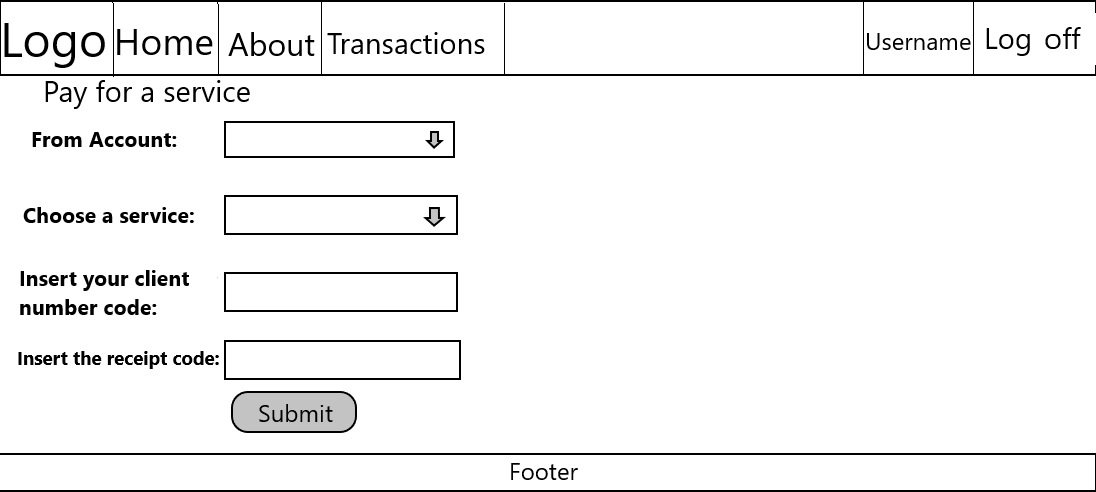


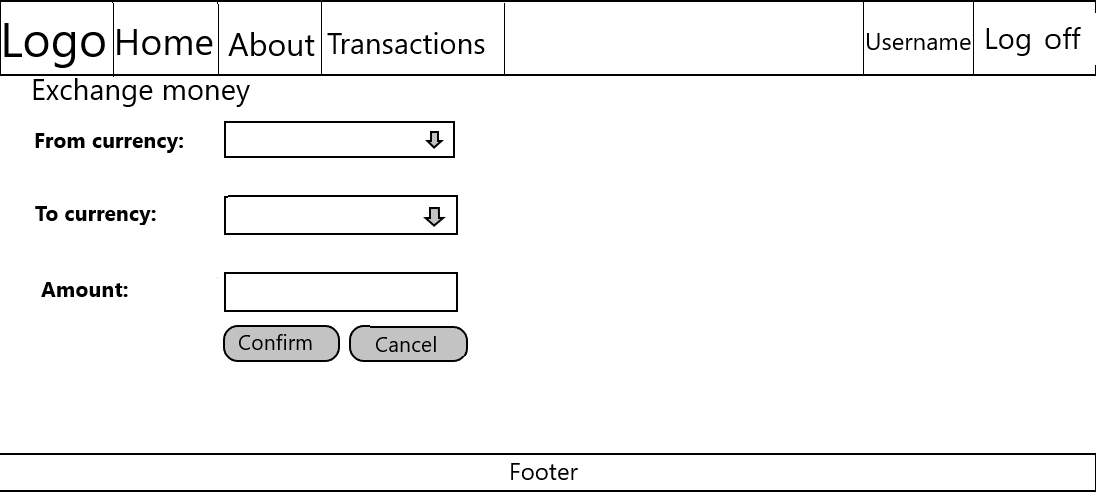












**3.1.2 Hardware interfaces**

Since it is a web application, it does not have any direct hardware interfaces. The hardware connection to the server is managed by the browser and the connection to the database is managed by the operating system on the server.

**3.1.3 Software interfaces**

The application communicates with the database in order to get the information about the user’s wallet, it is able to both modify and read from the database.

**3.1.4 Communications interfaces**

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for the web portal.

**3.2 Functional requirements**

## **3.2.1 User class - The client:**

3.2.1.1 **Functional requirement 1.1**

**ID:FR1**TITLE: User Registration  
DESC: To create an account, the client must press the ‘Register’ button from the navigation bar or from the ‘Login’ page. The First and Last Name fields are checked to see if the characters are letters(these fields must be written). The Email field must be completed with an unused Email. The ‘password’ field cannot be left empty; the password must respect the required form imposed by the site. The ‘confirm password’ field that checks to see if the password was entered correctly. The ‘Address’ field where the client will write his home address. The ‘Phone’ field where the client will write his phone number. The ‘Sign up’ button to finish the registration.  
RAT: In order for a user to register on the application.

3.2.1.2 **Functional requirement 1.2**

**ID:FR2**TITLE: User Login  
DESC: The ‘Enter email’ field that gets checked to see if the email is correct(CANNOT BE LEFT EMPTY). The ‘Enter password’ field that gets checked to see if the password is correct(CANNOT BE LEFT EMPTY). ‘Remember me’ option to keep the user logged for a future visit. ‘Login’ button to finish the login. ‘Sign up’ button for clients that are not yet registered.  
RAT: In order for the user to use his account.

3.2.1.3 **Functional requirement 1.3**

**ID:FR3**  
TITLE: Change password  
DESC: The ‘Current Password’ field that checks if the password is correct(CANNOT BE LEFT EMPTY). The ‘New Password’ field where the user writes his new password. The field must be completed and it is checked to see if it respects the required conditions. The ‘Confirm New Password’ field that checks to see if the password was entered correctly(CANNOT BE LEFT EMPTY). The ‘Update password’ button to finish changing the password.  
RAT: In order for the user to change his password.

3.2.1.4 **Functional requirement 1.4**

**ID:FR4**TITLE: Change addressDESC: The ‘Current address’ field that shows the user’s current address(CANNOT BE LEFT EMPTY). The ‘Change address’ field where the user can change their address.(CANNOT BE LEFT EMPTY). ‘Update address’ button to finish changing the address.  
RAT: In order for the user to change his address.

3.2.1.5 **Functional requirement 1.5**

**ID:FR5**TITLE: Account Info  
DESC: The ‘User info’ table where the user’s personal details such as his address/phone number will be displayed. The ‘Account number’ field where the user can see the identifier of his account. The ‘Account Balance’ field where you can see your current account’s balance in the account’s specific currency. The ‘Currency’ field where the currency of the current account will be displayed. There will be a field called ‘Total amount; that will display the total amount of money(Multiple choices) we have in all of our accounts. ‘Transfer Money’ button that redirects the user to a page where they can transfer money to someone. The ‘Pay For a Service’ button that redirects the user to a page where they can pay for a certain service. The ‘Exchange Money’ button that redirects the user to a new page where he will be able to change his current password. The ‘Change address’ button which redirects the user to a new page where he will be able to change his current address.  
RAT: To see the account information

3.2.1.6 **Functional requirement 1.6**

**ID:FR6**

TITLE: Money transfer  
DESC: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 inserted 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.  
RAT: So that the user can transfer money to other users.

3.2.1.7 **Functional requirement 1.7**

**ID: FR7**TITLE: Payment for a service

DESC: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.  
RAT: So that the user can transfer money between other of his accounts.

3.2.1.8 **Functional requirement 1.8**

**ID: FR8**TITLE: Currency exchangeDESC:The user will choose which currency he will exchange into another with the ‘From Currency’ dropdown list. The user will choose which currency he will exchange money with the ‘To Currency’ dropdown list.42. 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. After pressing cancel there will appear the message ‘Are you sure about that?’ and the buttons ‘Yes’ and ‘No’ to confirm the action. ‘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. If the account from which the user wants to exchange the money does not have the wanted amount, the user will receive the message ‘Not enough money to proceed’.  
RAT: So that the user can exchange his money to another currency

3.2.1.9 **Functional requirement 1.9**

**ID: FR9**

TITLE: Transactions view

DESC: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.

RAT: So that the user can view his transaction.

## **3.2.2 Admin class - The Admin:**

3.2.2.1 **Functional requirement 2.1**

**ID:FR10**

**Feature: Administrator login**

In order to administer the application and its users, an administrator should be logged into the site.

**Scenario: Successful log-in**

When the administrator logs in with an administrator account, the admin page will appear on the navigation.

3.2.2.2 **Functional requirement 2.2**

**ID: FR11**

**Feature: Check user activity**

**Scenario: Verify the transactions of an user**

Given the administrator is logged in

The administrator should be able to see all of the transactions made by

said user

3.2.2.3 **Functional requirement 2.3**

**ID: FR12**

**Feature: Manage users**

**Scenario: Delete an user**

Given the administrator is logged in

When the administrator deletes an user

Then the deleted user should be removed from the list of users

**3.3 Performance requirements**

**ID: PRQ1**  
TAG: ApplicationLoadingTIme   
GIST: The amount of time it takes to load the application.   
SCALE: Seconds.   
METER: Observations done from the performance log during testing   
MUST: No more than 3s.   
PLAN: No more than 2s.   
WISH: No more than 1s.

**3.4 Design constraints**

**ID: DC1**   
TAG: ApplicationMemoryUsage   
GIST: The amount of memory occupied by the application.   
SCALE: MB.   
METER: Observations done from the performance log during testing   
MUST: No more than 10 MB.   
PLAN: No more than 4 MB.   
WISH: No more than 3 MB.  
MB: DEFINED: Megabyte

**3.5 Software system attributes**

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

3.5.1 Reliability

**ID: SSA1**  
TAG: SystemReliability 24   
GIST: The reliability of the system.   
SCALE: The reliability that the system gives keeps all of the transaction history.   
METER: Measurements obtained from 1000 transactions.   
MUST: More than 98% of the transaction history.   
PLAN: More than 99% of the transaction history.   
WISH: 100% of the transaction history.

3.5.2 Availability

**ID: SSA2**   
TAG: SystemAvailability   
GIST: The availability of the system when it is used.   
SCALE: The average system availability (not considering network failure).  
 METER: Measurements obtained from 1000 hours of usage during testing.   
MUST: More than 98% of the time.   
PLAN: More than 99% of the time.   
WISH: 100% of the time.

**ID: SSA3**   
TITLE: Internet Connection   
DESC: The application should be connected to the Internet.   
RAT: In order for the application to communicate with the database.   
DEP: none

3.5.3 Security

**ID: SSA3**   
TAG: UserCreateAccountSecurity   
GIST: The security of creating accounts for users of the system.   
SCALE: If a user wants to create an account and the desired username is occupied, the user should be asked to choose a different user name.   
METER: Measurements obtained on 1000 hours of usage during testing.   
MUST: 100% of the time.

3.5.4 Maintainability

**ID: SSA4**   
TITLE: Application extensibility   
DESC: The application should be easy to extend. The code should be written in a way that it favors implementation of new functions.   
RAT: In order for future functions to be implemented easily to the application.   
DEP: none

**ID: SSA5**   
TITLE: Application testability   
DESC: Test environments should be built for the application to allow testing of the applications' different functions.   
RAT: In order to test the application.   
DEP: none

3.5.5 Portability

**ID: SSA6**   
TITLE: Application portability   
DESC: The application should be portable with iOS and Android.   
RAT: The adaptable platform for the application to run on.   
DEP: none

**4. Prioritization and Release Plan**

In order to get a view of how to divide the requirements into different releases and what requirements should be included in which release, a prioritization of the requirements is needed. This section discusses the choice of prioritization methods and gives a suggestion of how the release plan for these requirements could look like.

**4.1 Choice of prioritization method**

Since the requirement list and the team size is not that large, we stuck to a simple yes or no vote.

**4.2 Release Plan**

We decided on a single full featured release.