
Software Requirements Specification

for

Bank Account Management

**Prepared by: Diaconu Alexandru
Florescu Emilia
Hinoveanu Elena
Ica Alexandra**

Table of Contents

Revision History	1
1. Introduction.....	2
1.1 Purpose.....	2
1.2 Product Scope.....	2
1.2 Intended Audience and Reading Suggestions	2
1.4 Definitions, Acronyms, and Abbreviations	2
1.5 References.....	2
2. Overall Description	3
2.1 Product Perspective	3
2.2 Product Functions	3
2.3 User Classes and Characteristics	4
2.4 Operating Environment	4
2.5 Design and Implementation Constraints	4
2.6 User Documentation	4
2.7 Assumptions and Dependencies.....	4
3. External Interface Requirements	5
3.1 User Interfaces.....	5
3.2 Hardware Interfaces.....	5
3.3 Software Interfaces	5
3.4 Communications Interfaces.....	6
4. System Features	7
5. Other Nonfunctional Requirements	14
5.1 Performance Requirements	14
5.2 Safety Requirements.....	14
5.3 Security Requirements.....	14
5.4 Software Quality Attributes.....	14
5.5 Business Rules	15
6. Other Requirements	16
Appendix A: Glossary.....	16
Appendix B: Analysis Models	16
Appendix C: To Be Determined List	16

Revision History

Name	Date	Reason For Changes	Version
Diaconu Alexandru Florescu Emilia Hinoveanu Elena Ica Alexandra	15.03.2017	Initial Version	1
Diaconu Alexandru Florescu Emilia Hinoveanu Elena Ica Alexandra	22.03.2017	Initial Version	1.1

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Bank Account Management application. It will explain the features of the system, the interfaces of the system and what the system will do.

1.2 Product Scope

This software system will be a bank account management system. This system will be designed to automate the entire management process. It will use the client-server architecture.

More specifically, this system is designed to allow a user to create an account, manage it, make transactions and domestic payments. Besides this, an administrator can create and delete user accounts, change details and review transactions.

The benefits of this application include: ease of access (the user is able to manage his account from any computer with Internet access).

The strategy adopted for this project is the Agile approach.

1.2 Intended Audience and Reading Suggestions

This document is intended for both the stakeholders and the developers of the system.

1.4 Definitions, Acronyms, and Abbreviations

1.5 References

<https://github.com/ica-alexandra/Bank-Application.git>

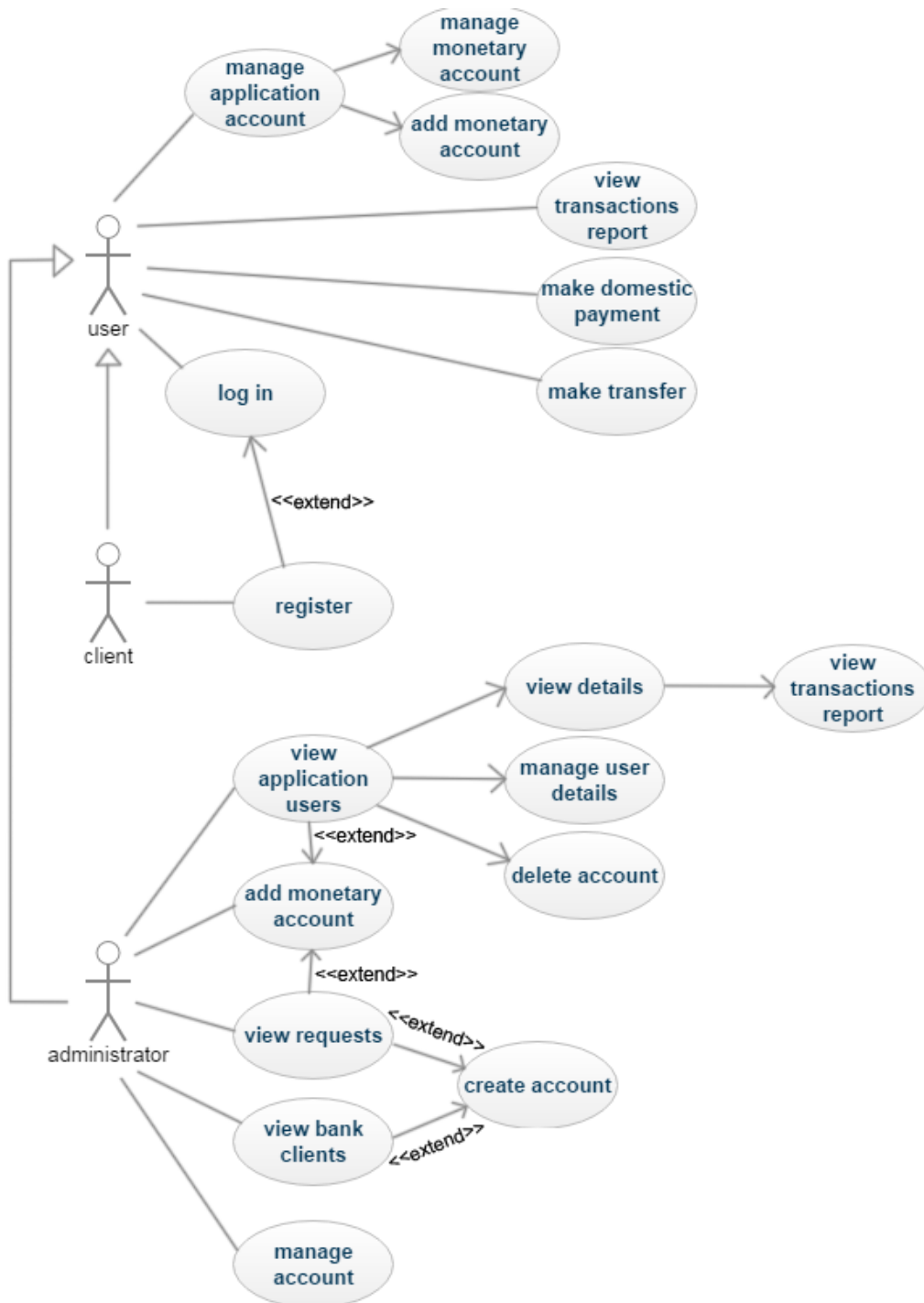
2. Overall Description

2.1 Product Perspective

This application is a new, self-contained product which will allow the users to manage their bank accounts.

2.2 Product Functions

Use Case Diagram



The main functionalities of the applications are:

- Log in;
- Create a new account (with administrator validation);
- Password recovery;
- View bank clients (only for administrator);
- View application users (only for administrator);
- Search for a specific user (only for administrator);
- Create new user account (only for administrator);
- Validate user requests (only for administrator);
- Manage account details (both for administrator and regular user);
- Create new monetary account ;
- Manage monetary account;
- View the Transactions Report (for application users);
- Make domestic payments (for application users);
- Make online transactions (for application users).

2.3 User Classes and Characteristics

The application users will be bank employees (as administrators) and private individuals (as regular users), the targeted social group being wage-earners.

2.4 Operating Environment

The application supports the following web browsers: Google Chrome v57.0.2987 and Mozilla Firefox v52.0.

IDE: Microsoft Visual Studio.

2.5 Design and Implementation Constraints

The application will be displayed only in English.

Backend language: C#

Frontend: HTML, CSS, JavaScript

Framework: ASP MVC

Technology: .NET

The application will be a freeware.

2.6 User Documentation

System Requirements Specifications document, System Design Specifications document and testing documents.

2.7 Assumptions and Dependencies

The users will need an active Internet connection in order to use the application.

Introduced data must respect the formatting options of the field in case.

3. External Interface Requirements

3.1 User Interfaces

The application should be accessible via web.

The layout should be as follows:

- The header which displays
 - the logo
 - the currently logged user
 - the Log Out and Manage Account menu
- The body, which displays
 - the menu, on the left side (links to Users, Clients and Requests for administrators and links to Transactions Report, Payment and Transfer for regular users)
 - the content, on the right side
- The footer
 - Contact
 - It is a link to a page which contains:
 - the address of the bank's main office
 - the telephone number of the bank
 - timetable
 - email
 - registration details: CUI, capital, SWIFT, BIC
 - Terms and Conditions (it is a link to a page with the terms and conditions)
 - Copyright

The user interface will contain:

- Alphanumeric fields with the personal details of the client
- A table with the client's accounts (displaying the account code, currency and amount)
- A list of options for each account (block, delete, view)
- A field for adding a new monetary account
- A menu (on the left)

The administrator interface will contain:

- A welcoming message
- A list with requests from users

3.2 Hardware Interfaces

Not applicable

3.3 Software Interfaces

The application is connected to the database (MySQL) through a connection string located in the web. config file from the application project.

There will be a communication interface between the server and the database (functions to read from and write into the database) and another one between the server and the client application (get input text and display data).

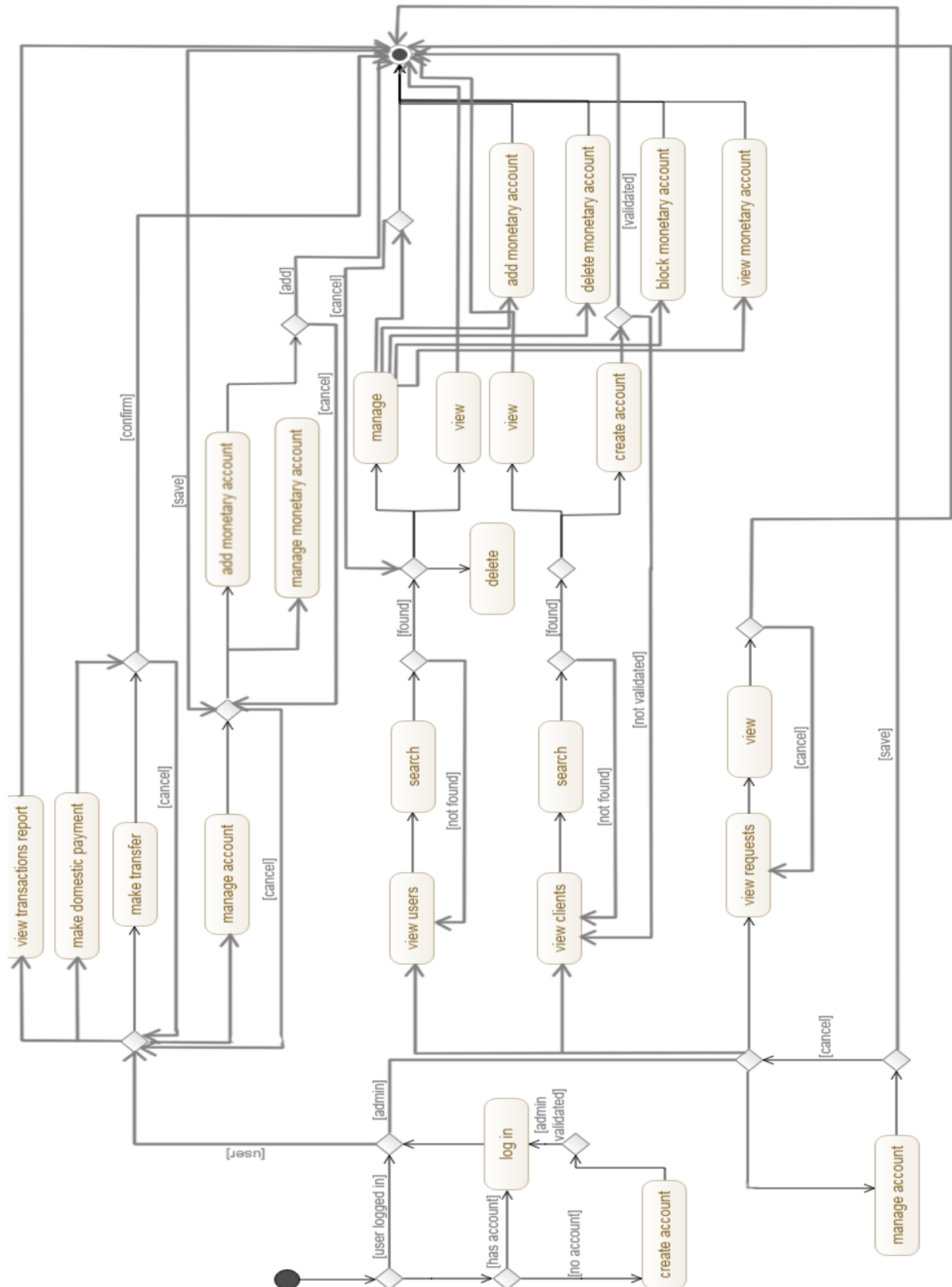
Function prototypes:

- `public Data GetData(int index, dbtable table);`
- `public void AddData(Data data, dbtable table);`

3.4 Communications Interfaces

Each administrator will use an email address to communicate with the users.
Transfer protocol used: HTTP.

4. System Features



4.1.1 Description and Priority

4.1.2 Stimulus/Response Sequences

4.1.3 Functional Requirements

- **Log in**

- The user can log into his account by typing in a user ID (email by default) and password
- If either the user ID or password introduced by the user is incorrect, an error message will be displayed and the log in operation will not succeed.

This interface will consist of:

- ✓ A text box for user code (7 alphanumeric symbols – it is auto-generated)
- ✓ A text box for password (min 8 alphanumeric symbols, containing both numbers and chars and at least one uppercase letter)
- ✓ A link for recovering the password (with the tag: 'Forgot password?')
- ✓ A "Log in" button
- ✓ A "Create account" button

- **Create a new account**(with administrator validation)

- Users can make a request for new account
- Users will fill in a form with personal details
- All fields are mandatory. If a field is left empty, or does not meet the size and formatting constraints, a warning message is displayed.
- The user must agree to the terms and conditions.
- Only bank clients will receive validation (from an administrator)
- The account will be created and the user is assigned an ID (his email) and password
- There will be a "Cancel" button, in case the user does not want to proceed creating the account.

This interface will consist of:

- ✓ A text box for Name (max 20 chars)
- ✓ A text box for Surname (max 30 chars)
- ✓ A text box for CNP (13 numeric symbols)
- ✓ A text box for ID card series (2 chars)
- ✓ A text box for ID card number (6 numeric symbols)
- ✓ A textbox for telephone number (10 numeric symbols)
- ✓ A text box for the email address (<user>@<domain>)
- ✓ A text box for address (max 50 chars)
- ✓ A checkbox for agreeing with the terms and conditions
- ✓ A "Submit" button
- ✓ A "Cancel" button

- **Password recovery**

- There will be an option for users to request a new password (in case they forgot their password)
- If the user introduces an invalid email, a warning message will be displayed

This interface will consist of:

- ✓ a textbox for the email
- ✓ a “Submit” button
- ✓ a “Cancel” button

- **View bank clients (only for administrator)**

- Bank clients will be listed in a table
- There will be some “View” functionality, to display more details about a specific client

For this functionality there will be an interface which will contain:

- ✓ a “search” button
- ✓ a table with all of the bank`s clients who use the application. The table will have the following columns:
 - Id
 - Name
 - Surname
 - CNP
 - Email
 - Actions

- **View application users (only for administrator)**

- Application users will be listed in a table
- There will be a list of options for each client (operations with users accounts)
- Administrators will be able to view details about users
- Administrators will be able to manage user accounts (change details)
- Administrators will be able to delete user accounts

For this functionality there will be an interface which will contain:

- ✓ a “search” button (which will redirect the user to the “search” page)
- ✓ a table with all of the bank`s clients who use the application. The table will have the following columns:
 - Id
 - Name
 - Surname
 - CNP
 - Email
 - User Code
 - Actions (containing buttons for “delete”, “view” and “manage”).

- **Search for a specific client (only for administrator)**

- Administrators will be able to search for a specific client of the bank by CNP or email

The interface for **searching for clients** in the database will contain:

- ✓ a textbox for introducing the user`s CNP,
- ✓ a textbox for email,
- ✓ a “search” button,
- ✓ a “cancel” button.

- **Search for a specific user (only for administrator)**

- Administrators will be able to search for a specific user by CNP, user code or email

The interface for **searching for users of the application** in the database will contain:

- ✓ a textbox for introducing the user`s CNP
- ✓ a textbox for email
- ✓ a textbox for the user code
- ✓ a “search” button
- ✓ a “cancel” button

- **Create new user account (only for administrator)**

- If the solicitor is a client of the bank, the administrator can approve his request for new account

- **Validate user requests (only for administrator)**

- If the solicitor is a client of the bank, the administrator can approve his request for new account
- Administrators can validate users requests for creating new monetary accounts

- **View account details** (both for administrator and regular user)

- Both types of users will be able to access the personal details from their account
- Administrators can view any user`s account details

The interface for **viewing a user`s details** will contain:

- ✓ fields for:
 - Name
 - Surname
 - CNP
 - ID Card Series
 - ID Card number
 - Telephone number
 - Email Address

- ✓ a table with the user`s accounts (with columns for account number, currency, amount , state – meaning that the account is active or blocked - and actions – a “View” button)
- ✓ a “Cancel” button.

The interface for **viewing a monetary account** of a user contains:

- ✓ Alphanumeric fields for: Account number, type of currency, Total Amount, IBAN
- ✓ A table for the transactions report with columns for: Date of transaction, Details (IBAN of the transferring account), Amount.

The interface for **viewing a client`s details** will contain:

- ✓ alphanumeric fields for Name, Surname, CNP, ID Card Series, ID Card number, Telephone number, Email Address
- ✓ a “Create Account” button
- ✓ a “Cancel” button.

- **Edit account details** (both for administrator and regular user)

- Both types of users will be able change the personal details from their account
- There will be some “Cancel” functionality, in case the user changes his mind and does not want to change anything.
- Administrators will, also, be able to change application users account details

- 1. **Editing account details as admin**

- The interface for **managing a user`s details** will consist of:

- ✓ A textbox for Name
 - ✓ A textbox for Surname
 - ✓ A textbox for CNP
 - ✓ A textbox for ID series
 - ✓ A textbox for ID Number
 - ✓ A textbox for Telephone Number
 - ✓ A textbox for Email Address
 - ✓ A textbox for Address
 - ✓ A table with user`s monetary accounts (with columns for Account Number, Currency, Amount and State) and a list of options: Block, Delete
 - ✓ A “Generate new password” button
 - ✓ An “Add monetary account” button
 - ✓ A “Save” button
 - ✓ A “Cancel” button

- The interface for **editing admin details** will consist of:

- ✓ A textbox for Name
 - ✓ A textbox for Surname
 - ✓ A textbox for Email Address
 - ✓ A “Save” button
 - ✓ A “Cancel” button

2. Editing account details as regular user (a regular user can only edit his personal details)

- The interface for **editing details** will consist of:
 - ✓ A textbox for Name
 - ✓ A textbox for Surname
 - ✓ A textbox for CNP
 - ✓ A textbox for ID series
 - ✓ A textbox for ID Number
 - ✓ A textbox for Telephone Number
 - ✓ A textbox for Email Address
 - ✓ A textbox for Address
 - ✓ A table with user's monetary accounts (with columns for Account Number, Currency, Amount and State) and a list of options: View, Block, Delete
 - ✓ An "Add account" button
 - ✓ A "Save" button
 - ✓ A "Cancel" button

- **Create new monetary account**

- Users shall make a request when they want to add a new monetary account (by choosing the type of currency for their new account)
- The administrator will have an "Add monetary" account option for each application user

The interface for **adding a monetary account** (both for regular user and admin) will consist of:

- ✓ An alphanumeric field for the IBAN code
- ✓ A dropdown list for choosing currency
- ✓ A "Cancel" button
- ✓ An "Add" button

- **Manage monetary account**

- Users will have the possibility to delete any of their existing monetary accounts
- Administrators will be able to delete any user's monetary accounts
- Users will have the possibility to (un)block any monetary account of theirs
- Administrators can (un)block any monetary account, belonging to any user
- Management of monetary accounts will be performed from the "Edit account details" interface

- **View the Transactions Report** (for application users)

- Application users will have the possibility to view all the transactions that have been made in the specified period of time (current month/year): this means transfers to their account, as well as withdrawals
- There will be an option for the users to select the monetary account for which they want to see the report

This interface will contain:

- ✓ A dropdown list with user`s accounts
- ✓ A table with the following columns:
 - Date
 - Details (IBAN of the transferring account)
 - Amount

- **Make domestic payments** (for application users)

- Users will have the possibility to choose the company to which they want to pay from a list
- Users shall fill in a form
- The users will have to choose from a list with their accounts the one from which the sum will be deducted
- There will be an option to introduce the bill`s unique code
- To finalize action, the user will have to confirm payment
- There will also be a “Cancel” button, in case the user does not want to proceed paying the bill in case

The interface will contain:

- ✓ A dropdown list with companies
- ✓ A dropdown list user`s accounts
- ✓ An alphanumeric field displaying the current balance of the selected account
- ✓ A textbox for introducing the amount of money
- ✓ Alphanumeric fields for the identification details of the beneficiary (IBAN, Name)
- ✓ A text field displaying the name of the bank of the beneficiary
- ✓ A textbox for introducing the bill`s code
- ✓ A “Confirm payment” button
- ✓ A “Cancel” button

- **Make online transactions** (for application users)

- Users shall fill in a form
- The users will have to choose from a list with their accounts the one from which the sum will be deducted
- They will have to introduce the IBAN of the beneficiary and the sum they want to transfer
- To finalize action, the user will have to confirm payment
- There will also be a “Cancel” button, in case the user does not want to proceed paying the bill in case

The interface will contain:

- ✓ A dropdown list to select the (user`s) account from which the sum will be paid
- ✓ An alphanumeric field displaying the current balance of the selected account
- ✓ A textbox for introducing the IBAN of the beneficiary

- ✓ A textbox for beneficiary's name
- ✓ A textbox for introducing the amount of money
- ✓ A textbox for the name of the beneficiary's bank
- ✓ A "Confirm transaction" button
- ✓ A "Cancel" button

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The application will run on Windows 7/8/10 OS
- System requirements:
 - processor: 1 GHz;
 - memory: at least 128 Mb;
 - free hard disk space: 10 Mb minimum
- The application requires Internet connection
- The application will run on Google Chrome and Mozilla Firefox

5.2 Safety Requirements

Not applicable

5.3 Security Requirements

Personal data is stored in the database and only administrators have access to the specified details.

5.4 Software Quality Attributes

Conditions for writing the code:

Rule 1 (required): No reliance shall be placed on undefined or unspecified behavior.

Rule 2 (required): Insert one space between the comment delimiter (//) and the comment text, as shown in the following example.

Rule 3 (advisory): No identifier in one name space should have the same spelling as an identifier in another name space, with the exception of structure member and union member names.

Rule 4(advisory): Write only one declaration per line.

Rule 5 (advisory): Add at least one blank line between method definitions and property definitions.

Rule 6 (advisory): Place the comment on a separate line, not at the end of a line of code.

Rule 7 (advisory): Use the + operator to concatenate short strings, as shown in the following code.

Rule 8 (advisory): Use meaningful names for query variables.

Rule 9 (advisory): Use camelCasing for method arguments and local variables.

Rule 10 (advisory): Prefix interfaces with the letter I. Interface names are noun (phrases) or adjectives.

Product qualities:

1. Reliability: personal information and sensitive data will be kept safe. The transactions, which may have delayed results, will be safe procedures.

2. Usability: The application will have a broad customer base, therefore using it shall be as simple as making a payment in person. It will meet the customers' need to be able to use the web site without training. The software will flow according to the users' expectations and offer only needed information (when needed).

3. Security: Customer data (such as passwords, IBAN, id card details etc) and other electronic information will be handled as securely as possible.

4. Availability: The application will have "24/7" availability, and it will be operational every day of the year. Availability means that the website will be and running 24/7/365 and it will be available when accessed by diverse compatible browsers.

5.5 Business Rules

Disclaimer

Use of this Bank website is subject to the following conditions and restrictions:

The accuracy and completeness of the information cannot be guaranteed. The website provides only financial products and services. Changes can be made without prior notice. The information provided is not intended as a substitute for professional advice. Without verification or further advice, the information is used on their own account and risk by the users.

6. Other Requirements

Appendix A: Glossary

The **transactions report** is the history of the payments to and from the user`s selected account.

CNP represents the personal numerical code (unique identifier for Romanian citizens).

CUI represents the unique registration code of the company.

The **SWIFT code** stands for Society for Worldwide Interbank Financial Telecommunication and is a unique identification code of each bank, used for international transactions.

The **BIC code** stands for Bank Identifier Code.

Appendix B: Analysis Models

The strategy adopted for this project is the V-model approach

Appendix C: