Software Requirements Specification

for

Bank Account Management

Prepared by: Diaconu Alexandru
Florescu Emilia
Hinoveanu Elena
Ica Alexandra

`

Table of Contents

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 revised	1.1
Diaconu Alexandru	05.04.2017	Class Diagram Added	1.2

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

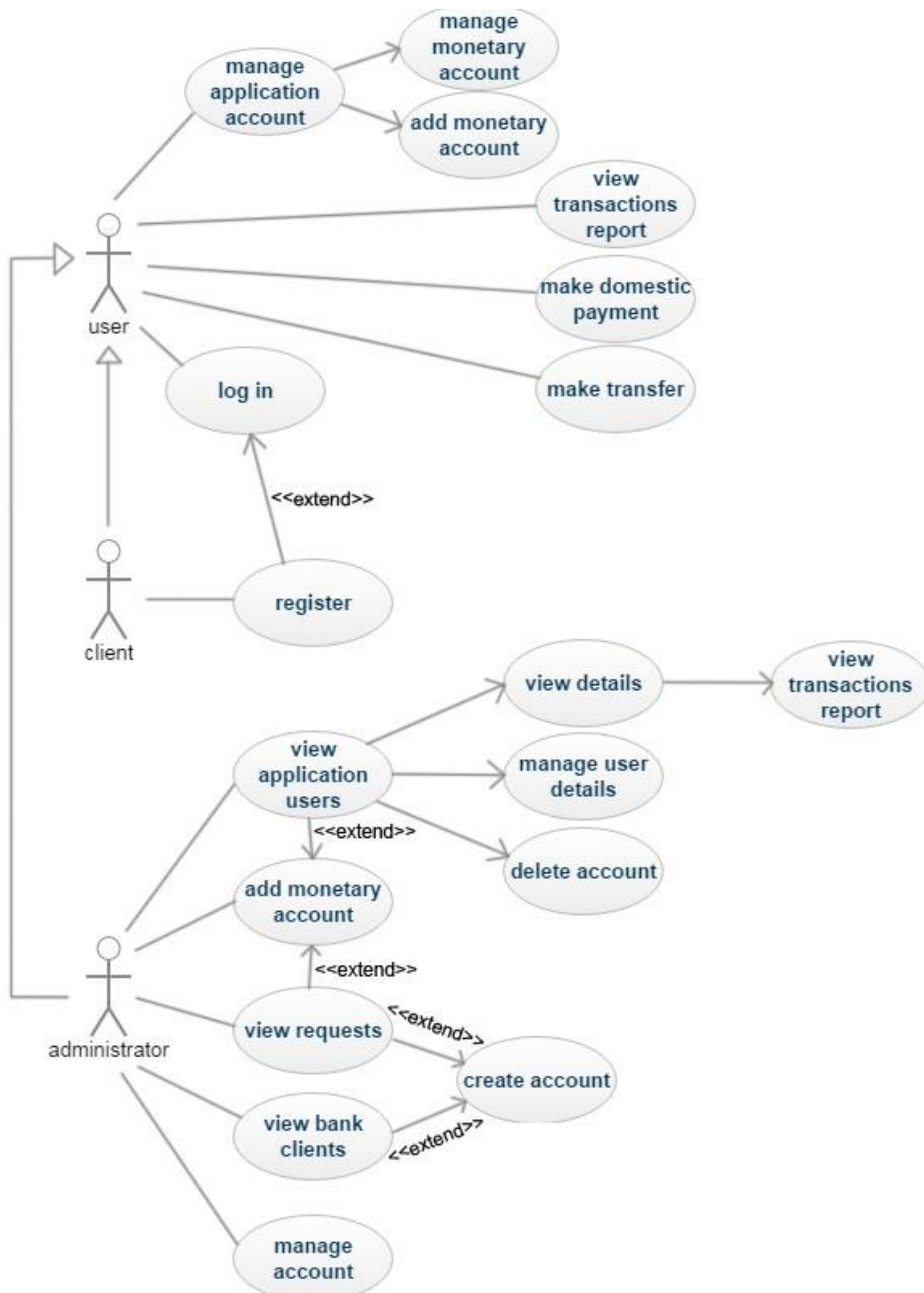
2. Overall Description

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:

1. Log in
2. Create a new account (with administrator validation)
3. Password recovery
4. View bank clients (only for administrator)
5. View application users (only for administrator)
6. Search for a specific user (only for administrator)
7. Create new user account (only for administrator)
8. Validate user requests (only for administrator)
9. Manage account details (both for administrator and regular user)
10. Create new monetary account
11. Manage monetary account
12. View the Transactions Report (for application users)
13. Make domestic payments (for application users)
14. 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.
15. The **layout** should be as follows:

- **15.1** The **header** which displays
  - 15.1.1 the logo
  - 15.1.2 the currently logged user
  - 15.1.3 the Log Out and Manage Account menu
- **15.2** The **body**, which displays
  - 15.2.1 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)
  - 15.2.2 the content, on the right side
- **15.3** The **footer**
  - 15.3.1 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
  - 15.3.2 Terms and Conditions (it is a link to a page with the terms and conditions)
  - 15.3.3 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

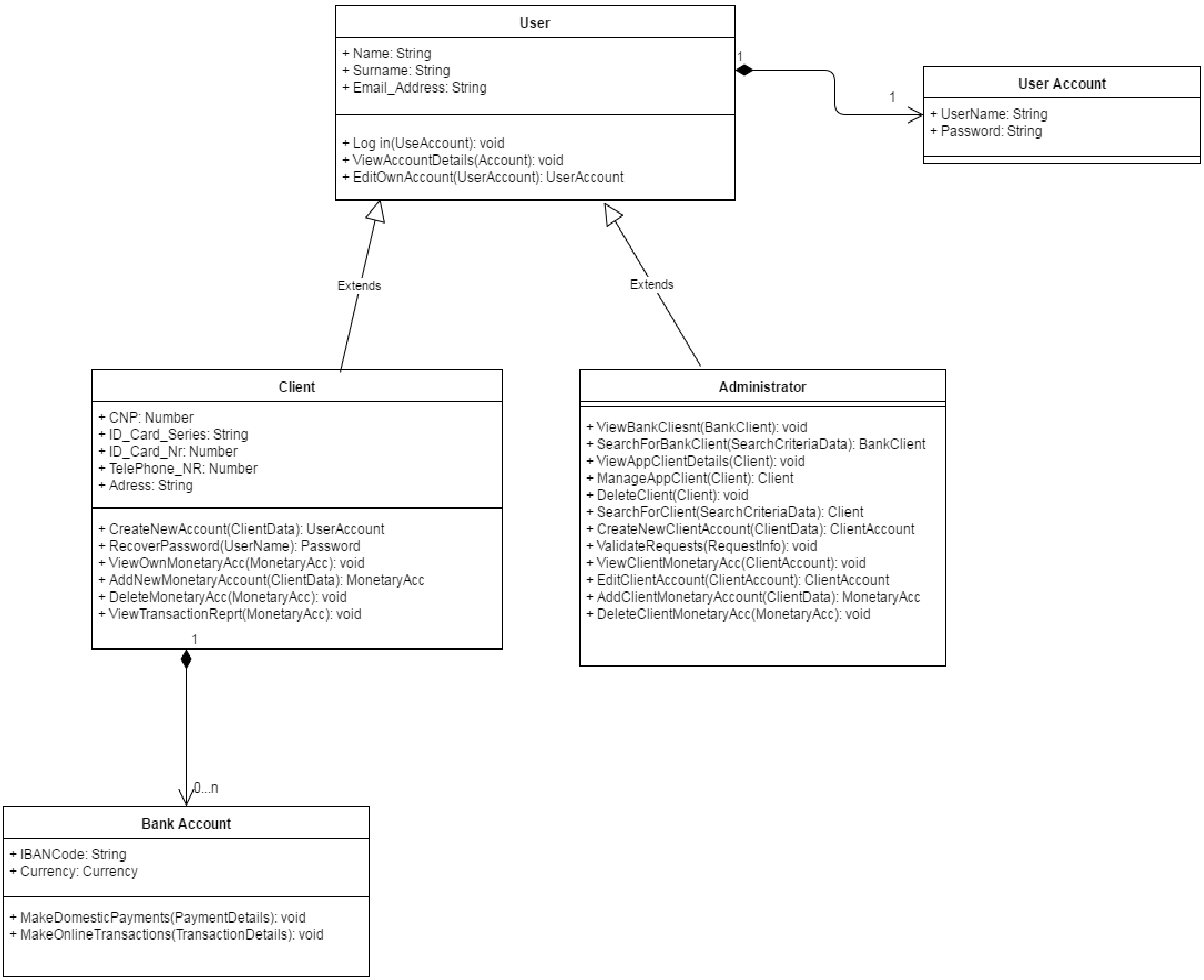
3.3 Software Interfaces

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:

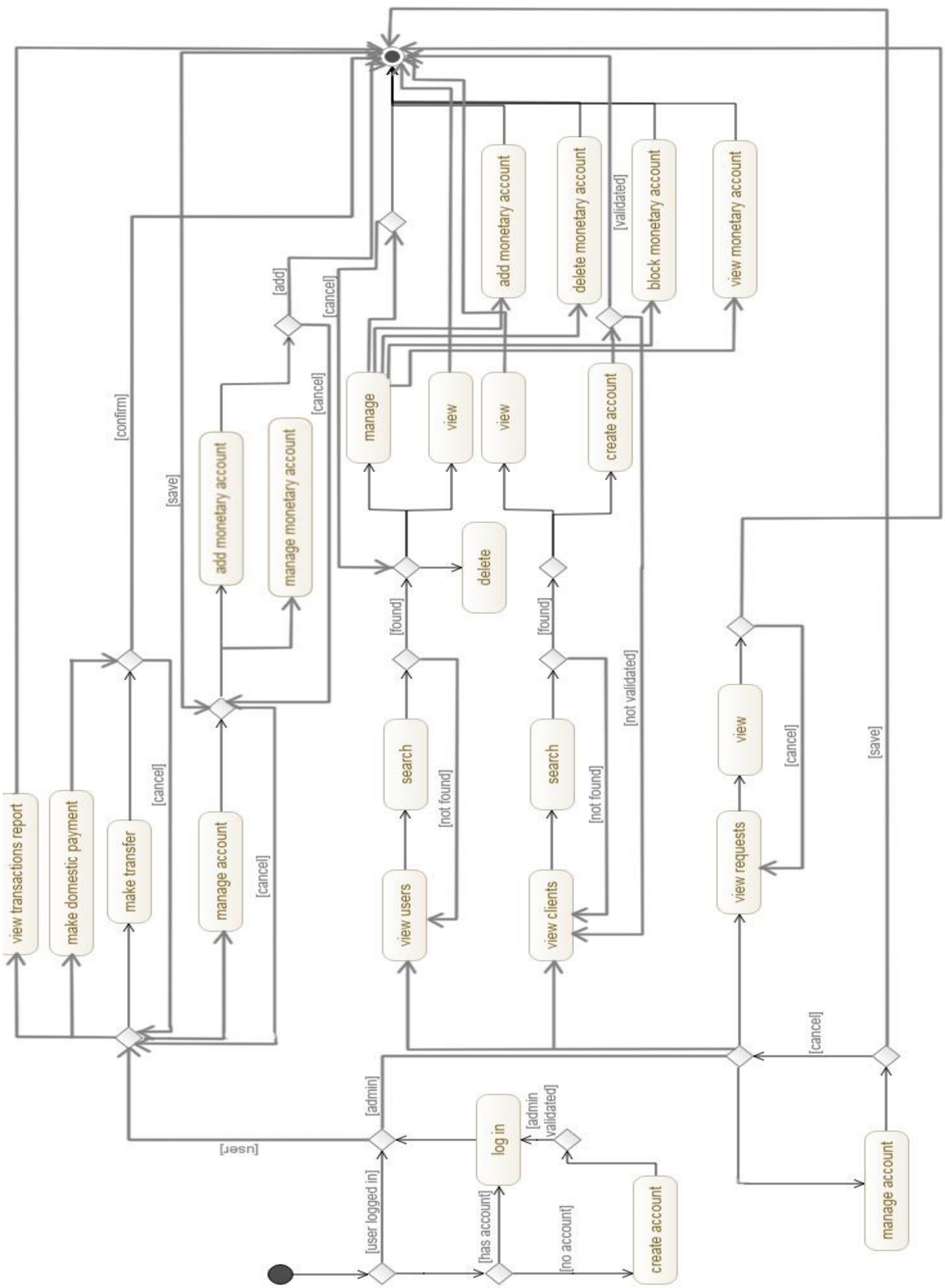
- public Data GetDatafromDb(int index, dbtable table);
- public void AddDatatoDb(Data data, dbtable table);
- public Data GetDatafromClient(int index, clienttable table);
- public void DisplayDataforClient(Data data, clienttable table);



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

1. **Log in**

1.1. The user can **log in**to his account by typing in a user ID (email by default) and password

1.2. 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:

1.1.1 A text box for user code (7 alphanumeric symbols; auto-generated)

1.1.2 A text box for password (min 8 alphanumeric symbols, containing both numbers and chars and at least one uppercase letter)

3.1 A link for recovering the password (with the tag: ‘Forgot password?’)

1.1.3 A “Log in” button

1.3 A “Create account” button (link to registering form)

2. **Create a new account** (with administrator validation)

2.1 Users will fill in a **form** with personal details

2.2 Users can make a request for new account (by filling in and **sending the form**)

2.3 All fields are mandatory. If a field is left empty, or does not meet the size and formatting constraints, a **warning message** is displayed.

2.4 The user must **agree to the terms and conditions**.

2.5 Only bank clients will receive **validation** (from an administrator)

2.6 The **account will be created** and the user is assigned an ID (his email) and password

2.7 There will be a “**Cance**l” button, in case the user does not want to proceed creating the account.

This interface will consist of:

2.1.1. A text box for Name (max 20 chars)

2.1.2. A text box for Surname (max 30 chars)

2.1.3. A text box for CNP (13 numeric symbols)

2.1.4. A text box for ID card series (2 chars)

2.1.5. A text box for ID card number (6 numeric symbols)

2.1.6. A textbox for telephone number (10 numeric symbols)

2.1.7. A text box for the email address (<user>@<domain>)

2.1.8. A text box for address (max 50 chars)

2.4 A checkbox for agreeing with the terms and conditions

2.2 A “Submit” button

2.7.1 A “Cancel” button

3. **Password recovery**
   3.1. There will be an option for users to **request a new password** (in case they forgot their password)
   3.2. Form to recover password
   This interface will consist of:
   3.2.1 a textbox for the email
   3.2.2 a "Submit" button
   3.2.3 a "Cancel" button
   3.3. If the user introduces an invalid email, an **error message** will be displayed

4. **View bank clients (only for administrator)**
   4.1. Bank clients will be listed in a **table**
   4.2. 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:
   4.1.1 a "search" button
   4.1.2 a table with all of the bank`s clients who use the application. The table will have the following columns:
   4.1.2.1 Id
   4.1.2.2 Name
   4.1.2.3 Surname
   4.1.2.4 CNP
   4.1.2.5 Email
   4.2.2.6. Actions

5. **View application users (only for administrator)**
   5.1. Administrators will be able to **view details** about users
   5.2. Application users will be listed in a **table**
   5.3. There will be a list of **options** for each client (operations with users` accounts)
   5.4. Administrators will be able to **manage** user accounts (change details)
   5.5. Administrators will be able to **delete** user accounts

   For this functionality there will be an interface which will contain:
   5.2.1 a "search" button (which will redirect the user to the "search" page)
   5.2.2 a table with all of the bank`s clients who use the application. The table will have the following columns:
   5.2.2.1 Id
   5.2.2.2 Name
   5.2.2.3 Surname
   5.2.2.4 CNP
   5.2.2.5 Email
   5.2.2.6 User Code
   5.2.2.7 Actions (containing buttons for "delete", "view" and "manage").

6. **Search for a specific client (only for administrator)**

6.1. 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:

6.1.1. a textbox for introducing the user`s CNP,
6.1.2. a textbox for email,
6.1.3. a “search” button,
6.1.4. a “cancel” button.

7. **Search for a specific user (only for administrator)**

7.1. 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:

7.1.1. a textbox for introducing the user`s CNP
7.1.2. a textbox for email
7.1.3. a textbox for the user code
7.1.4. a “search” button
7.1.5. a “cancel” button

8. **Create new user account (only for administrator)**

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

9. **Validate user requests (only for administrator)**

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

9.2. Administrators can **validate** users requests for creating **new monetary** accounts

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

10.1. Both types of users will be able to **access the personal details** from their account

10.2. Administrators can **view any user**`s account details

10.1.1 The interface for **viewing own details** will contain fields for:

10.1.1.1 Name
10.1.1.2 Surname
10.1.1.3 CNP
10.1.1.4 ID Card Series
10.1.1.5 ID Card number
10.1.1.6 Telephone number
10.1.1.7 Email Address

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

10.2.1 fields for:

10.2.1.1 Name
10.2.1.2 Surname
10.2.1.3 CNP
10.2.1.4 ID Card Series
10.2.1.5 ID Card number
10.2.1.6 Telephone number
10.2.1.7 Email Address

10.2.2 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)

10.2.3 a “Cancel” button.

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

10.3.1 Alphanumeric fields for: Account number, type of currency, Total Amount, IBAN

10.3.2 A table for the transactions report with columns for: Date of transaction, Details (IBAN of the transferring account), Amount.

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

10.4.1 alphanumeric fields for Name, Surname, CNP, ID Card Series, ID Card number, Telephone number, Email Address

10.4.2 a “Create Account” button

10.4.3 a “Cancel” button.

11. Edit account details (both for administrator and regular user)

11.1 Both types of users will be able **change the personal details** from their account

11.2 There will be some “**Cancel**” functionality, in case the user changes his mind and does not want to change anything.

11.3 Administrators will, also, be able to **change application users account details**

Editing account details as admin

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

11.3.1 A textbox for Name
11.3.2 A textbox for Surname
11.3.3 A textbox for CNP
11.3.4 A textbox for ID series
11.3.5 A textbox for ID Number
11.3.6 A textbox for Telephone Number
11.3.7 A textbox for Email Address
11.3.8 A textbox for Address
11.3.9 A table with user`s monetary accounts (with columns for Account Number, Currency, Amount and State) and a list of options: Block, Delete
11.3.10 A “Generate new password” button
11.3.11 An “Add monetary account” button
11.3.12 A “Save” button
11.3.13 A “Cancel” button

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

11.1.1.1 A textbox for Name
11.1.1.2 A textbox for Surname

11.1.1.3 A textbox for Email Address
11.1.1.4 A “Save” button
11.1.1.5 A “Cancel” button

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

11.1.2.1 The interface for **editing details** will consist of:
11.1.2.2 A textbox for Name
11.1.2.3 A textbox for Surname
11.1.2.4 A textbox for CNP
11.1.2.5 A textbox for ID series
11.1.2.6 A textbox for ID Number
11.1.2.7 A textbox for Telephone Number
11.1.2.8 A textbox for Email Address
11.1.2.9 A textbox for Address
11.1.2.10 A **table** with user`s monetary accounts (with columns for Account Number, Currency, Amount and State) and a list of options: View, Block, Delete
11.1.2.11 An “**Add account**” button
11.1.2.12 A “**Save**” button
11.1.2.13 A “**Cancel**” button

12. Create new monetary account

12.1. Users shall make a **request** when they want to add a new monetary account (by choosing the type of currency for their new account)

12.2. 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:

12.2.1. An alphanumeric field for the IBAN code
12.2.2. A **dropdown** list for choosing **currency**
12.2.3. A “**Cancel**” button
12.2.4. An “**Add**” button

13. Manage monetary account

13.1. **Users** will have the possibility to **delete** any of their existing monetary accounts

13.2. **Administrators** will be able to **delete** any user`s monetary accounts

13.3. **Users** will have the possibility to **(un)block** any monetary account of theirs

13.4. **Administrators** can **(un)block** any monetary account, belonging to any user

13.5. **Management of monetary** accounts will be performed from the “Edit account details” interface

14. View the Transactions Report (for application users)

14.1. 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

14.2. 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:

14.2.1. A **dropdown list with user`s accounts**

14.1.1 A **table** with the following columns:

14.1.1.1 Date

14.1.1.2 Details (IBAN of the transferring account)

14.1.1.3 Amount

15. Make domestic payments (for application users)

15.1 Users will have the possibility to **choose the company** to which they want to pay from a list

15.2 Users shall fill in a **form**

15.3 The users will have to **choose** from a list with their **accounts** the one from which the sum will be deducted

15.4 There will be an option to introduce the **bill`s unique code**

15.5 To finalize action, the user will have to **confirm payment**

15.6 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:

15.1.1 A dropdown list with companies

15.3.1 A dropdown list user`s accounts

15.2.1 An alphanumeric field displaying the current balance of the selected account

15.2.2 A textbox for introducing the amount of money

15.2.3 Alphanumeric fields for the identification details of the beneficiary (IBAN, Name)

15.2.4 A text field displaying the name of the bank of the beneficiary

15.4.1 A textbox for introducing the bill`s code

15.5.1 A "Confirm payment" button

15.6.1 A "Cancel" button

16. Make online transactions (for application users)

16.1 Users shall fill in a **form**

16.2 The users will have to **choose** from a list with their **account**s the one from which the sum will be deducted

16.3 They will have to **introduce** the IBAN of the **beneficiary** and the sum they want to transfer

16.4 To finalize action, the user will have to **confirm payment**

16.5 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:

16.2.1 A dropdown list to select the (user`s) account from which the sum will be paid

16.1.1 An alphanumeric field displaying the current balance of the selected account

16.3.1 A textbox for introducing the IBAN of the beneficiary

16.3.2 A textbox for beneficiary`s name
16.1.2 A textbox for introducing the amount of money
16.3.3 A textbox for the name of the beneficiary`s bank
16.4.1 A “Confirm transaction” button
16.5.1 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: