

**Software Requirements**

**Specification**

**for**

**Bank Account Management**

**Prepared by: Diaconu Alexandru,**

**Florescu Emilia,**

**Hinoveanu Elena,**

**Ica Alexandra**

`

Table of Contents

[1. Introduction 2](#_Toc477291279)

[1.1 Purpose 2](#_Toc477291280)

[1.2 Product Scope 2](#_Toc477291281)

[1.2 Intended Audience and Reading Suggestions 2](#_Toc477291282)

[1.4 Definitions, Acronyms, and Abbreviations 2](#_Toc477291283)

[1.5 References 2](#_Toc477291284)

[2. Overall Description 3](#_Toc477291285)

[2.1 Product Perspective 3](#_Toc477291286)

[2.2 Product Functions 4](#_Toc477291287)

[2.3 User Classes and Characteristics 5](#_Toc477291288)

[2.4 Operating Environment 5](#_Toc477291289)

[2.5 Design and Implementation Constraints 5](#_Toc477291290)

[2.6 User Documentation 5](#_Toc477291291)

[2.7 Assumptions and Dependencies 5](#_Toc477291292)

[3. External Interface Requirements 6](#_Toc477291293)

[3.1 User Interfaces 6](#_Toc477291294)

[3.2 Hardware Interfaces 6](#_Toc477291295)

[3.3 Software Interfaces 7](#_Toc477291296)

[3.4 Communications Interfaces 7](#_Toc477291297)

[4. System Features 8](#_Toc477291298)

[5. Other Nonfunctional Requirements 14](#_Toc477291299)

[5.1 Performance Requirements 14](#_Toc477291300)

[*5*.2 Safety Requirements 14](#_Toc477291301)

[5.3 Security Requirements 14](#_Toc477291302)

[5.4 Software Quality Attributes 14](#_Toc477291303)

[6. Other Requirements 15](#_Toc477291304)

[Appendix A: Glossary 15](#_Toc477291305)

[Appendix B: Analysis Models 15](#_Toc477291306)

[Appendix C: To Be Determined List 15](#_Toc477291307)

# 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

Transactions report

CNP

CUI

SWIFT code

BIC code

## 1.5 References

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

# 2. Overall Description

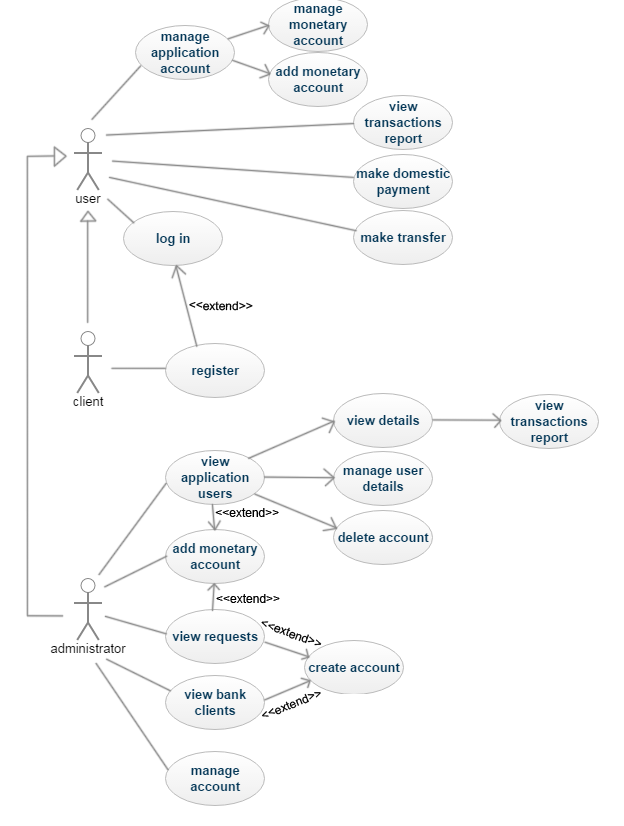
## 2.1 Product Perspective

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;
* Access to the database;
* Managing account details (both for administrator and regular user);
* The administrator validates user requests (creating accounts for a type of currency);
* The administrator can search for a specific account in the database;
* The user can add a new bank account using specific currency (with administrator validation);
* The user can view the Transactions Report;
* The user can make domestic payments;
* The user can make online transactions.

## 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 LogOut 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) with the following options: Transactions Report, Payment, Transfer

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).

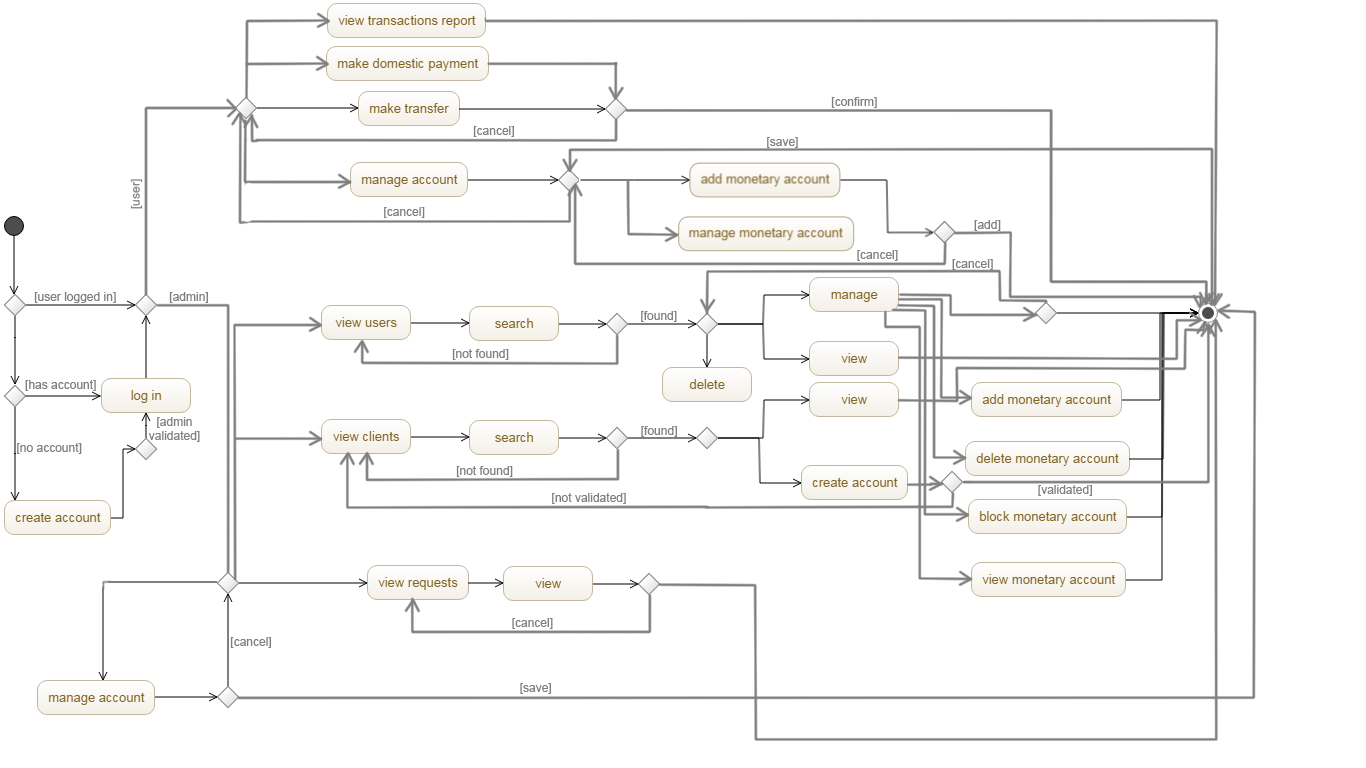
## 3.4 Communications Interfaces

Each administrator will use an email address to communicate with the users.

Transfer protocol used: HTTP.

# 4. System Features

1. Description and Priority
2. Stimulus/Response Sequences

**

1. Functional Requirements

* **Log in**

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

If either the user or password is incorrect a message will be displayed (“User/password is incorrect”).

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

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

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 request cannot be submitted without agreeing to the terms and conditions.

If the form is correctly filled in, the request is sent to an administrator. If the user is a client of the bank, the administrator validates the request and generates the user code and password (the account is created and added to the database). A validation email is sent to the client.

* **Password recovery**

The “Forgot password?” link redirects the user to a page with a textbox for the email, a “Submit” button and a “Cancel” button.

If the user introduces an invalid email, a warning message is displayed, otherwise a new password is generated and the client receives an informative email.

* **Access to the database**

- Administrators have **access to the application users**` database. 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, User Code and Actions (containing buttons for “delete”, “view” and “manage”).

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

* alphanumeric 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.

- Administrators have **access to the bank`s clients** database. 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 and Actions (containing a button for “view”).

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

1. **Editing account details as admin** (the administrator can only edit the details of a client who has an account in the application – any client registered)

**-** 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 **adding a monetary account** for a user will consist of:

* An alphanumeric field for the IBAN code
* A dropdown list for choosing currency
* A “Cancel” button
* An “Add” 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

1. **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
* **The administrator validates user requests (creating accounts)**

An administrator can create accounts for bank`s clients (look up the CNP in the database). If the person is in the database, the account can be created (by pressing the “create account” button). The interface consists of:

* A field for password
* A “Generate Credentials” button
* A “Cancel” button

If an incorrect password is introduced, credentials are not generated and an error message is displayed.

When a user wants to add a monetary account, the administrator will receive a notification. He has to add that account if he validates the action.

* **The administrator can search for a specific account in the database**

- 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.

- 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.
* **The user can add a new bank account using specific currency**

- The interface for **adding a monetary account** will consist of:

* An alphanumeric field for the IBAN code
* A dropdown list for choosing currency
* A “Cancel” button
* An “Add” button

The account will not be added without administrator permission.

* **The user can view theTransactions Report** (both admins and regular users)

This interface will contain:

* A dropdown list for choosing the account for which the user wants to see the transactions report
* A table with the following columns: Date, Details (IBAN of the transferring account) and Amount
* **The user can make domestic payments**

The interface will contain:

* A dropdown list to select the company to which the payment will be transferred
* 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 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
* **The user can make online transactions**

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

**Non-functional requirements:**

* The application will run on Windows 7/8/10 OS
* System requirements:

- processor : 3.5 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. Other Nonfunctional Requirements

## 5.1 Performance Requirements

## *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

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.

**5.5 Busin**e**ss 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 andis 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: To Be Determined List