

**Software Requirements**

**Specification**

**for**

**Informatic System for Banking Transactions**

**Version 1.3**

**Prepared by Biraianu Alex-Valentin**

**Ungureanu Adrian-Ion**

**Gingeoveanu Stefan**

**Banica Mihai-Vlad**

**Becherescu Corina**

**Prepared for university project**

**12.03.2017**

# Table of Contents

[Table of Contents 2](#_Toc482177705)

[Revision History 3](#_Toc482177706)

[1. Introduction 4](#_Toc482177707)

[1.1 Purpose 4](#_Toc482177708)

[1.2 Product Scope 4](#_Toc482177709)

[1.3 Intended Audience and Reading Suggestions 4](#_Toc482177710)

[1.4 References 4](#_Toc482177711)

[2. Overall Description 4](#_Toc482177712)

[2.1 Product Perspective 4](#_Toc482177713)

[2.2 Product Functions 4](#_Toc482177714)

[2.3 User Classes and Characteristics 5](#_Toc482177715)

[2.4 Operating Environment 5](#_Toc482177716)

[2.5 Design and Implementation Constraints 5](#_Toc482177717)

[2.6 User Documentation 5](#_Toc482177718)

[3. External Interface Requirements 5](#_Toc482177719)

[3.1 User Interfaces 5](#_Toc482177720)

[3.2 Hardware Interfaces 7](#_Toc482177721)

[3.3 Software Interfaces 7](#_Toc482177722)

[4. System Features 7](#_Toc482177723)

[4.2 Database: 8](#_Toc482177724)

[5. Other Nonfunctional Requirements 9](#_Toc482177725)

[5.1 Performance Requirements 9](#_Toc482177726)

[5.2 Software Quality Attributes 9](#_Toc482177727)

[5.3 Business Rules 9](#_Toc482177728)

[6. Other Requirements 10](#_Toc482177729)

[Appendix A: Glossary 10](#_Toc482177730)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Comments** | **Version** |
|  |  |  |  |
|  |  |  |  |
| Biraianu Alex-Valentin  Ungureanu Adrian | 12.03.2017 | First Revision | 1 |
| Biraianu Alex-Valentin  Ungureanu Adrian | 03.04.2017 | Added the remaining software features.  Added Activity and use Case Diagrams.Added 10 Misra Rules | 1.1 |
| Biraianu Alex-Valentin | 09.04.2017 | Added Class Diagrams and software interfaces.Added task list. | 1.2 |
| Biraianu Alex-Valentin  Ungureanu Adrian | 01.05.2017 | Modified revision history, modified table of content and changed some of the requirements of 4.3 and 4.4 | 1.3 |

# 

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to present a detailed description of the Informatic System for a Banking application. It will explain the features of the system, the interfaces of the system and what the system will do.

## 1.2 Product Scope

The system will be built on a client/server architecture for assisting the bank on achieving the following tasks:

* Allowing users to see information about their own accounts
* Changing their personal information and password
* Making transactions and pay for different services
* Viewing their transaction history

## 1.3 Intended Audience and Reading Suggestions

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

## 1.4 References

Link to the app*:* [*https://github.com/UAdrian/banking-app*](https://github.com/UAdrian/banking-app)

Link to the license: https://github.com/UAdrian/banking-app/blob/master/information/LICENSE

# 2. Overall Description

## 2.1 Product Perspective

This product is a self-contained system intended for managing a bank and the accounts of its clients.

## 2.2 Product Functions

The product will have the following main functions:

* Viewing and modifying Accounts of users
* Transactions between different accounts and services
* A transcribed log containing all performed actions of the user

## 2.3 User Classes and Characteristics

The application users will be bank employees which will have administrator privileges and clients which will have regular accounts.

## 2.4 Operating Environment

The application supports Windows platform.

## 2.5 Design and Implementation Constraints

The software will have a GNU General Public License.

## 2.6 User Documentation

The user will be given a user manual containing all the applications functions.

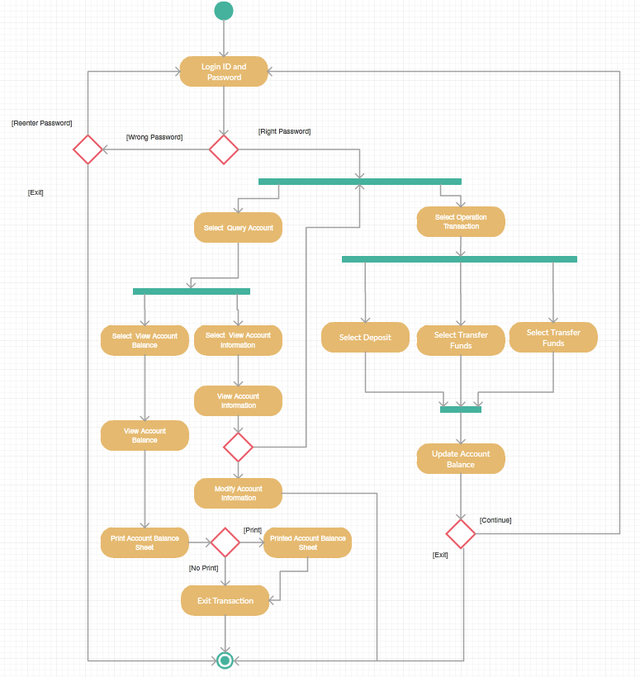
# 3. External Interface Requirements

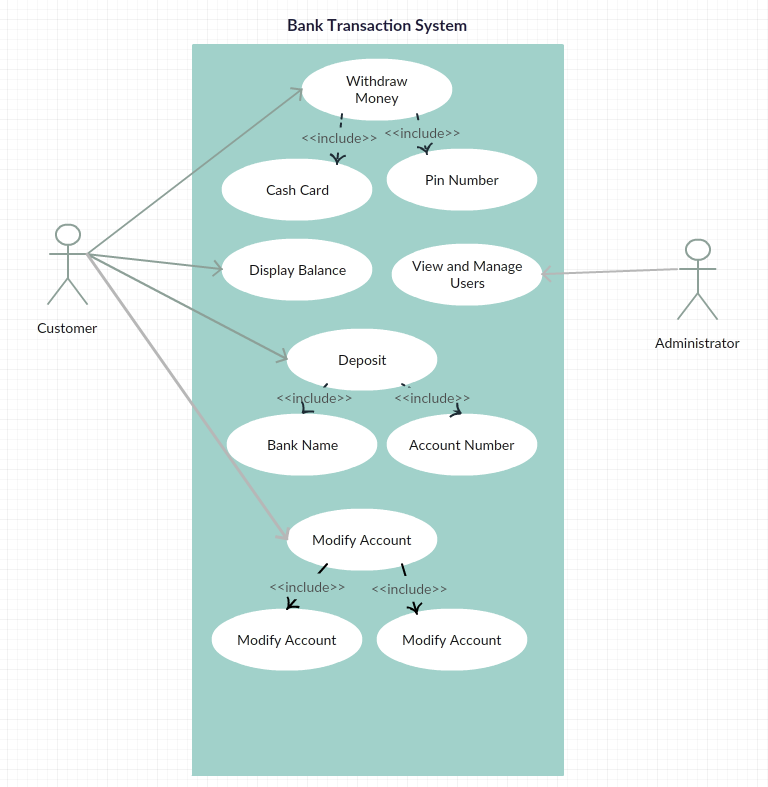
## User Interfaces

The interface will have the standard help, minimize and close buttons.

The log in interface will open in the middle of the screen.

After the user logs in this interface will close and the main app will open, this facilitates the operations between the user and the app.***For more information see 4.3***





## 3.2 Hardware Interfaces

Not applicable for this type of software.

## 3.3 Software Interfaces

Function Account GetPersonalData(String UserID,String Password) sends it's parameters as keys for the function and returns an Account object containing the personal data of the user.

# 4. System Features

**4.1 General:**

1. User Interface should be secure and accessed with user and password.
2. The user and password of the client (pre-generated) are kept in database and are accessed by the app.
3. The data presented in the interface should be easy to understand and accessible to the user.
4. The user will be able allowed to see:
   1. Account details
   2. Change personal information
   3. Change password
   4. Perform transaction to other accounts
   5. Perform transaction to special services
   6. View his Transaction history
5. All of the above functionalities should be accessible fromt he main body of the aplication.
6. The accounts should be of different types depending on the currency.
7. The user can download a copy of the transaction history as PDF.

## 4.2 Database:

1. Management of data for clients, accounts and the bank.
2. The database should allow:
   1. Data recovery at a specified time.
   2. Support of offline backup.
   3. Administration of data.
   4. Users to access information concurrently.
   5. Definitions of diferite types of users with different access rights.
   6. Monitoring all types of queries and implementing complex data structures.
3. Should have a ‘log’ for the activities performed.
4. The database will contain 3 tables:
   1. Account for Usernames and passwords
   2. Clients for normal level Accounts
   3. Admins for Bank level Accounts
5. The Clients database and admin contain the key field which will be used to give them acces to different pages.
6. 1 for Admin and 0 for Clients.
   1. **Stand-alone application:**
7. It should facilitate the login of the users, access.
8. The Login screen is a graphical interface containing two box fields for the user and password, ’login’ button and a ’forgot password’ button.
9. The password box field is case sensitive and conceals the password.
10. Pressing the login button sends the date to the web api to be processed.
11. If they are correct it leads to the account page else an error message appears declaring either one or both of the fields maybe incorrect .
12. The Login screen is in the middle of the screen.
13. The Account Page is the main body of the application:
    1. It greats the user with a welcome message and notifications about the account.
    2. It contains a menu on the left alowing access to the different sections.
14. It sends the inputed data to the web api.
15. In the middle of the window there are some basic informations about the bank accounts the user has access to.
16. The aplication will ask the user after each transaction if he wishes to download the info of the transaction he just did.

**4.4 Web server:**

1. Facilitates the contact between the stand-alone application and the database.
2. The transformed data shall be encode and only decode once it has arrived at it’s destination.
3. The system should contain at least the following modules :
   1. Module that extracts the data from the database
   2. Module that writes the data to the database
   3. Module that facilitates the transactions.
4. The system will manage information about the account and transactions.

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

It requires a pc with at least a microprocessor.

## 5.2 Software Quality Attributes

1.No reliance shall be placed on undefined or unspecified behaviour.

2.Assembly language shall be encapsulated and isolated.

3.Sections of code should not be “commented out”.

4.A tag name shall be a unique identifier.

5.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.

6.If objects or functions are declared more than once their types shall be compatible.

7.There shall be no definitions of objects or functions in a header file.

8. All automatic variables shall have been assigned a value before being used.

9. For each function parameter the type given in the declaration and definition shall be identical, and the return types shall also be identical.

10.Octal constants (other than zero) and octal escape sequences shall not be used.

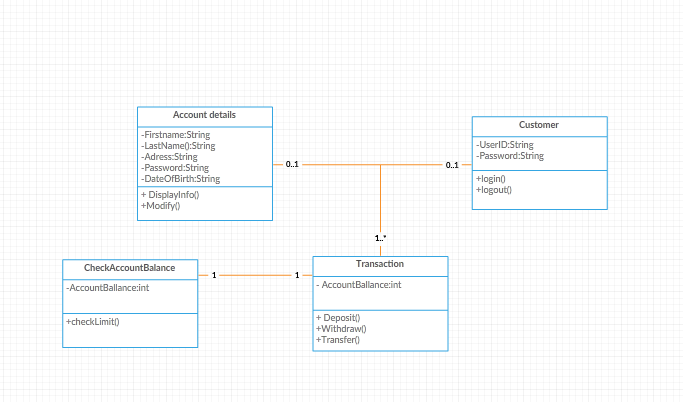
## 5.3 Business Rules

Disclaimer this product is made for the intent use as a banking application for administrators of a bank and clients of said bank not for any other purpose or intent. Using this application for any other purpose invalidates the agreement between the software makers and users .

# 6. Other Requirements

The Applications language will be English.

# Appendix A: Glossary



|  |  |  |
| --- | --- | --- |
| Task | Member | Deadline(dd-mm-yyyy) |
| Database Management and Test Creation | Becherescu Corina | 08.05.2017 |
| Server Management ,Encoding Data and Transfer | Gingeoveanu Stefan | 08.05.2017 |
| Aplication GUI Elements implementaition, Login Screen | Ungureanu Adrian | 08.05.2017 |
| Account Balance functions | Biraianu Alex | 08.05.2017 |
| GUI Elements creation, Account personal information screen and functions | Banica Vlad | 08.05.2017 |

# Appendix B: Analysis Models

The Analysis model shall be waterfall model.