

**Software Requirements**

**Specification**

**for**

**Informatic System for Banking Transactions**

**Version 1.0**

**Prepared by Biraianu Alex-Valentin**

**Ungureanu Adrian**

**Gingoveanu Stefan**

**Banica Mihai-Vlad**

**Becherescu Corina**

**Prepared for university project**

**12.03.2017**

# 

# Table of Contents

[Table of Contents 1](#_1fob9te)

[Revision History 1](#_3znysh7)

[1. Introduction 2](#_tyjcwt)

[1.1 Purpose 2](#_3dy6vkm)

[1.2 Product Scope 2](#_1t3h5sf)

[1.3 Intended Audience and Reading Suggestions 2](#_4d34og8)

[1.4 Definitions, Acronyms, and Abbreviations 2](#_2s8eyo1)

[1.5 References 2](#_17dp8vu)

[2. Overall Description 3](#_26in1rg)

[2.1 Product Perspective 3](#_lnxbz9)

[2.2 Product Functions 3](#_35nkun2)

[2.3 User Classes and Characteristics 3](#_1ksv4uv)

[2.4 Operating Environment 3](#_44sinio)

[2.5 Design and Implementation Constraints 3](#_2jxsxqh)

[2.6 User Documentation 3](#_z337ya)

[2.7 Assumptions and Dependencies 4](#_1y810tw)

[3. External Interface Requirements 4](#_4i7ojhp)

[3.1 User Interfaces 4](#_2xcytpi)

[3.2 Hardware Interfaces 4](#_1ci93xb)

[3.3 Software Interfaces 4](#_3whwml4)

[3.4 Communications Interfaces 4](#_2bn6wsx)

[4. System Features 5](#_3as4poj)

[4.1 System Feature 1 5](#_1pxezwc)

[4.2 System Feature 2 (and so on) 5](#_49x2ik5)

[5. Other Nonfunctional Requirements 5](#_2p2csry)

[5.1 Performance Requirements 5](#_147n2zr)

[5.2 Safety Requirements 6](#_23ckvvd)

[5.3 Security Requirements 6](#_ihv636)

[5.4 Software Quality Attributes 6](#_32hioqz)

[5.5 Business Rules 6](#_1hmsyys)

[6. Other Requirements 6](#_41mghml)

[Appendix A: Glossary 6](#_2grqrue)

[Appendix B: Analysis Models 6](#_vx1227)

[Appendix C: To Be Determined List 7](#_1v1yuxt)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Comments** | **Version** |
|  |  |  |  |
|  |  |  |  |
| Biraianu Alex-Valentin  Ungureanu Adrian  Gingoveanu Stefan  Banica Mihai-Vlad  Becherescu Corina | 12.03.2017 | First Revision | 1 |
|  |  |  |  |

# 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 Definitions, Acronyms, and Abbreviations

## The online audience is a online meeting between an user and a municipality employee.

The **document flow management** is an informatics system that allows the distribution, storage and finding the documents in electronic format.

## 1.5 References

*https://github.com/UAdrian/banking-app*

# 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 the following OS Windows.

## 2.5 Design and Implementation Constraints

*<Describe any items or issues that will limit the options available to the developers. These might include: corporate or regulatory policies; hardware limitations (timing requirements, memory requirements); interfaces to other applications; specific technologies, tools, and databases to be used; parallel operations; language requirements; communications protocols; security considerations; design conventions or programming standards (for example, if the customer’s organization will be responsible for maintaining the delivered software).>*

## 2.6 User Documentation

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

## 2.7 Assumptions and Dependencies

*<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>*

# 3. External Interface Requirements

## 3.1 User Interfaces

User:

* A Login interface that contains the box fields

*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## 3.2 Hardware Interfaces

Not applicable for this type of software.

## 3.3 Software Interfaces

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## 3.4 Communications Interfaces

The application will have a server that will communicate between the app and it’s database and it will encrypt and decrypt the data from the database.

# 4. System Features

*<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>*

## 4.1 System Feature 1

*<Don’t really say “System Feature 1.” State the feature name in just a few words.>*

1. Description and Priority

*<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>*

1. Stimulus/Response Sequences

*<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>*

1. Functional Requirements

*<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary.*

*Use “TBD” as a placeholder to indicate when necessary information is not yet available.>*

*<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>*

REQ-1:

REQ-2:

## 4.2 System Feature 2 (and so on)

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

It requires a pc with at least a microprocessor.

## 5.2 Safety Requirements

Not applicable for this software.

## 5.3 Security Requirements

## 5.4 Software Quality Attributes

The software respects the rules of usability,testability and maintainability.

## 5.5 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

# Appendix B: Analysis Models

The Analysis model shall be waterfall model.

# Appendix C: To Be Determined List

*<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>*