Software Requirements Specification

for

Distributed computer system for banking transactions

Version 2.0

Bușe-Dragomir Alexandru

FACULTY OF AUTOMATICS, COMPUTER SCIENCE AND ELECTRONICS

13/03/2019

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
BAT	08.03.2019	Initial format	1.0
BAT	13.03.2019	Added requirements	2.0

1. Introduction

1.1 Purpose

The purpose of the application is to simulate an informatic system which allows clients to verify their accounts and transactions.

1.2 Document Conventions

This Document was created based on the IEEE template for System Requirement Specification Documents.
For more document conventions see Appendix A: Glossary.

1.3 Intended Audience and Reading Suggestions

- *Typical Users, such as students of other users, who want to use BAT (Bank-account Administration and Transaction) to manage their bank account and make transactions.*
- *Programmers who are interested in working on the project for further development.*

1.4 Product Scope

BAT allows clients to verify their accounts and online transactions. The application contains a Graphical Interface for the user, a module for security measures, a Database Server (that will store information about the clients, their accounts and their transactions) and a WEB Server. The user has access to the service only from the Graphical Interface. The system will offer the banking features only after the user has logged in. The features will be:

- *Info about the account*
- *Change address*
- *Change passwords*
- *Make transactions*
- *Make payments to certain services*
- *Visualize transaction history*

The clients can have multiple accounts with different currencies like EUR, USD, RON.

1.5 References

- *BAT's Bit-Bucket Repository*
  https://bitbucket.org/NyKto/banking-transactions-system

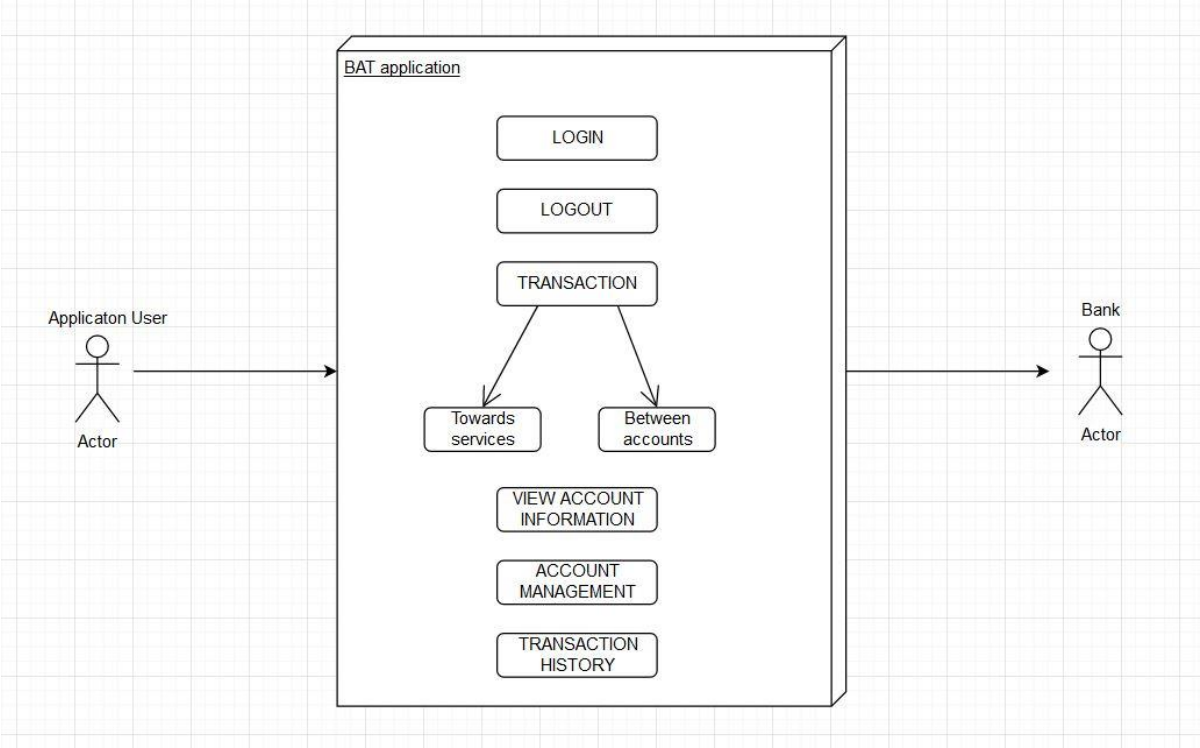
- *IEEE Template for System Requirement Specification Documents*
  https://goo.gl/nsUFwy

- *Java SE Development Kit 8*
  https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

2. Overall Description

2.1 Product Perspective

BAT is a new, self-contained desktop application that allows its users to administrate their bank accounts, to make transactions between either different accounts or services.



Simple functionality of the application

2.2 Product Functions

- ✓ *Login/Logout*
- ✓ *Register*
- ✓ *Change password*
- ✓ *Edit profile*
- ✓ *Create bank account with currency*
- ✓ *Transfer money between accounts (including currency conversion if needed)*
- ✓ *Transfer money to other services*
- ✓ *View transaction history*
- ✓ *Deposit money*
- ✓ *View account information*
- ✓ *Balance check*

2.3 User Classes and Characteristics

- *Typical Users, such as students of other users, who want to use BAT (Bank-account Administration and Transaction) to manage their bank account and make transactions.*

2.4 Operating Environment

The minimum hardware requirements of BAT are:
Windows 10 (8u51 and above)
Windows 8.x (Desktop)
Windows 7 SP1
Windows Vista SP2
Windows Server 2008 R2 SP1 (64-bit)
Windows Server 2012 and 2012 R2 (64-bit)

RAM: 128 MB Disk space: 124 MB for JRE; 2 MB for Java Update
Processor: Minimum Pentium 2 266 MHz processor
Browsers: Internet Explorer 9 and above, Firefox, Chrome, Opera

2.5 Design and Implementation Constraints

Programming language: Java 8 (JDK 8u201)
Graphical User Interface: Java FX 8.0.1
Integrated Development Environment: Intellij IDEA 2018.3.4
Project Management Tool: Apache Maven 3.6.0
Software Version Control: Git 2.21.0
Git Client: Source Tree 3.0.17
Version Control Repository Hosting Service: BitBucket
Relational Database Management System: MySQL 8.0.15

2.6 User Documentation

The user documentation will be provided after the completion of the project, under the form of a simple to understand Java Doc.
In the meantime, this document is a good starting point in understanding the basic structure of the project and of its interface and functionality.
We recommend, for both the casual application user and for the new developer, the reading of this document at first; this will ensure a quick and efficient understanding of the documentation from the first read.

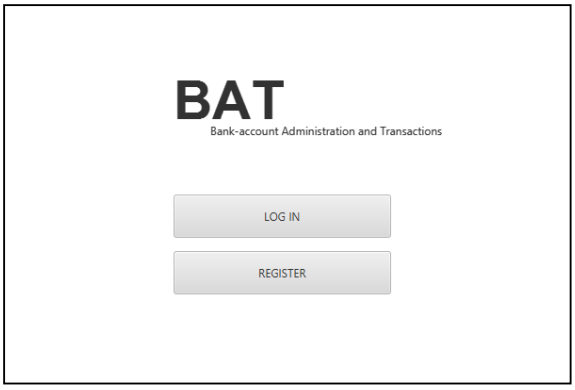
2.7 Assumptions and Dependencies

N.A.

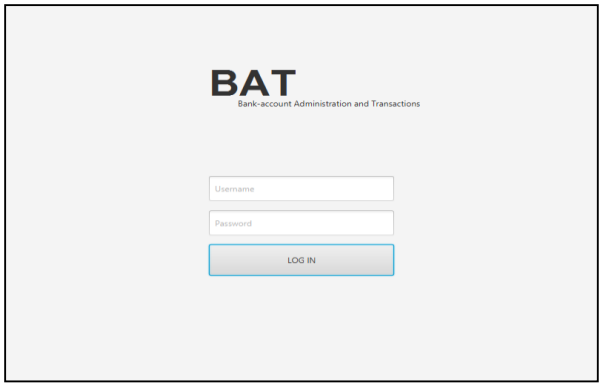
3. External Interface Requirements

3.1 User Interfaces

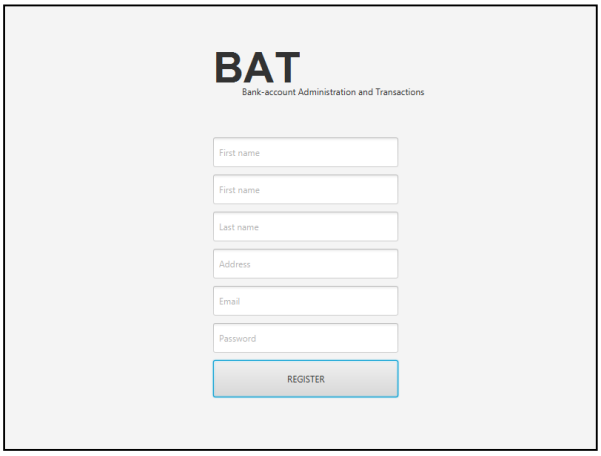
1. Startup screen



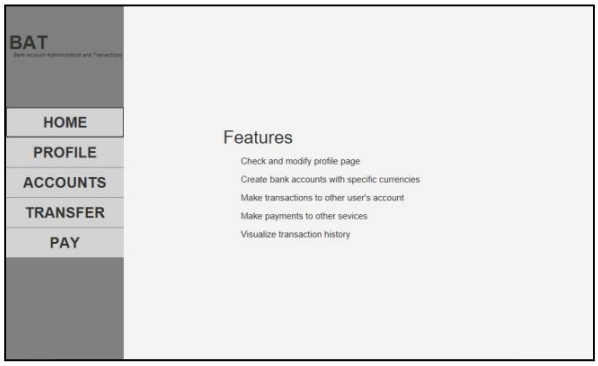
2. Login screen



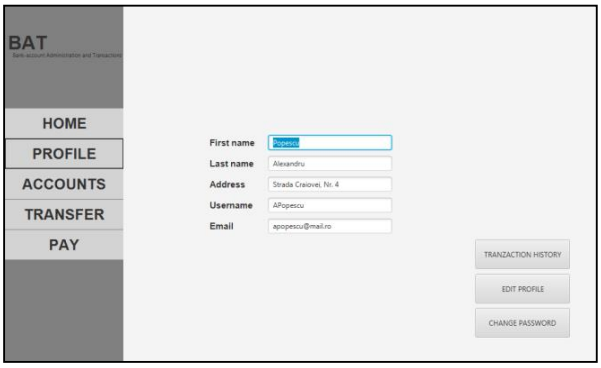
3. Register screen



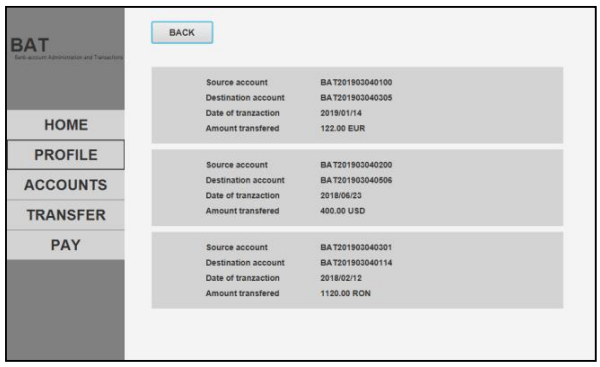
4. Home screen



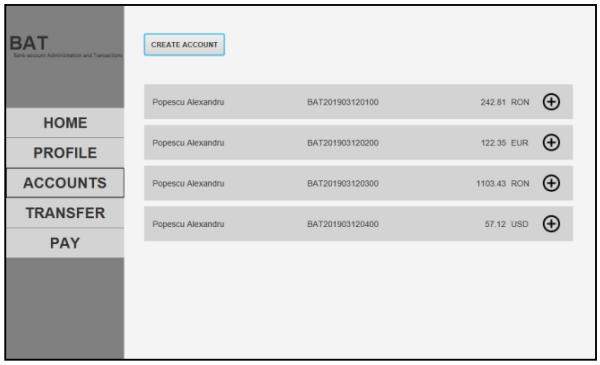
5. Profile screen



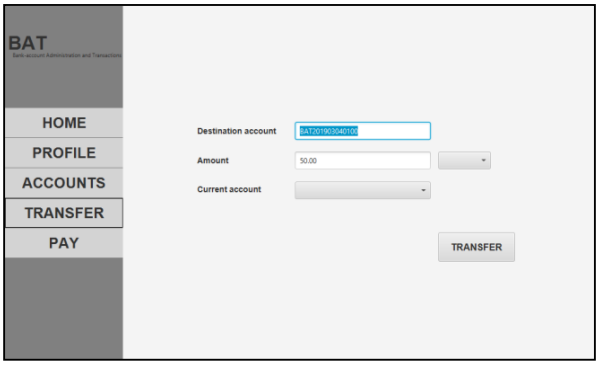
6. Profile transaction screen



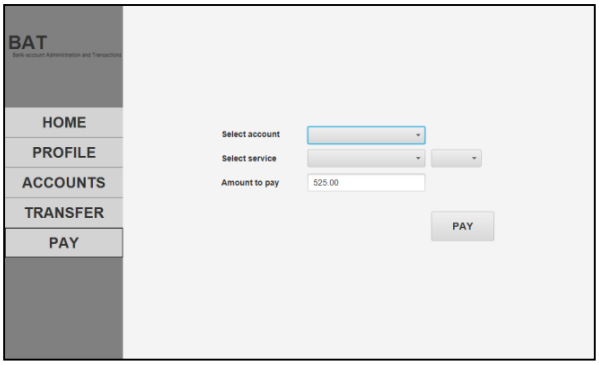
7. Accounts screen



8. Transfer screen



9. Pay page



3.2 Hardware Interfaces

N.A.

3.3 Software Interfaces

public Connection getDbConnection() throws ClassNotFoundException, SQLException

3.4 Communications Interfaces

BAT requires a stable internet connection in order to access the bank accounts and to make transactions. The connection is obtained through a Secure Hyper Text Transfer Protocol (HTTPS). A connection trough a MODEM should be avoided as it might be unstable, so we recommend:

- *a DSL connection with an average speed of at least 5 Mbps that won't go below 1 Mbps*
- *a cable connection with speeds in the range 1Mbps – 20 Mbps*
- *a Wireless connection with a good signal*
- *currently, although a 4G service ensures a speed of up to 100 Mbps, our application is not*
- *available for mobile devices, so a cellular connection will not be considered*

4. System Features

4.1 User registration

4.1.1 Description and Priority

In order to use the application, each new user must go through the registration process.
High priority.

4.1.2 Stimulus/Response Sequences

In order to create a new account, the user must provide his full name, a username and a password, as well as his email and address.

4.1.3 Functional Requirements

REQ-1: In order to create a new account, the user must press the Register button from the starting page

REQ-2: Field for first name; Checks to see if all characters are letters from the English alphabet and text has less than 20 characters; This field cannot be left empty

REQ-3: Field for last name; Checks to see if all characters are letters from the English alphabet and text has less than 20 characters; This field cannot be left empty

REQ-4: Field for username; Checks to see if all characters are letters from the English alphabet, text has less than 15 characters and the username is unique; This field cannot be left empty

REQ-5: Field for address; Checks to see if the text has less than 50 characters and it can only contain a few special symbols; This field cannot be left empty

REQ-6: Field for email; Checks to see if the text has less than 40 characters and it has a valid format; This field cannot be left empty

REQ-7: Field for password; Checks to see if the text has between 8 - 20 characters, letters from the English alphabet (both upper case and lower case), at least one special character and at least one number; This field cannot be left empty

REQ-8: The password will be saved under the form of a hash inside the database in order to ensure an extra layer of security in case of a malicious attempt

REQ-9: Button for completing the registration; If all inserted data is valid the account will be created and stored in the database

4.2 User authentication (login/logout)

4.2.1 Description and Priority

In order to use the application, each new user must be logged in.
High priority.

4.2.2 Stimulus/Response Sequences

In order to log in, the user must provide the username and the password that have been used for the account creation.

4.2.3 Functional Requirements

REQ-10: In order to log in, the user must press the Login button from the starting page

REQ-11: Field for username; The user must provide the username of his/her account

REQ-12: Field for password; The user must provide the password linked to his/her specific username; This field is case-sensitive

REQ-13: Button for logging in; Once pressed, the button will trigger the process of username and password validation

REQ-14: After a successful log in, the user will be redirected to the Home Page, where details about the application are displayed

REQ-15: Once logged in, a welcome message will be displayed in top left corner, above the log out button

REQ-16: Once logged in, the user will have access to a logout button that can be found in the upper left corner. This button will redirect the user back to the login screen

4.3 User profile

4.3.1 Description and Priority

Each user will have the possibility to check and edit his/her profile information, view his/her transaction history and change his/her password.
Low priority.

4.3.2 Stimulus/Response Sequences

The user will be able to edit the first name, last name, username, email, address and password. He will also be able to check his transaction history.

4.3.3 Functional Requirements

REQ-17: Field for first name; The user is able to change his first name but will still need to follow the same validation rules as in the register form; Available after the Edit Profile button

REQ-18: Field for last name; The user is able to change his last name but will still need to follow the same validation rules as in the register form; Available after the Edit Profile button

REQ-19: Field for address; The user is able to change his address but will still need to follow the same validation rules as in the register form; Available after the Edit Profile button

REQ-20: Field for username; The user is able to change his username but will still need to follow the same validation rules as in the register form; Available after the Edit Profile button

REQ-21: Field for email; The user is able to change his email but will still need to follow the same validation rules as in the register form; Available after the Edit Profile button

REQ-22: Button for checking the transaction history of the user. This button will open a new panel containing all the transactions made by the user

REQ-23: The transaction history contains a label for each performed transaction. The label is composed of a source account, a destination account, transaction date and the amount transferred

REQ-24: The source account will be the account where the money is transferred from

REQ-25: The destination account will be the account the money is transferred to

REQ-26: The transaction date will be the date when the transaction is made

REQ-27: The transferred amount is the sum of money that has been transferred

REQ-28: Button for editing the profile. When pressed, all the fields in the page will become editable and the Edit Profile button will be replaced by a Confirm Button

REQ-29: Button for confirming the profile editing. When pressed, the information in the fields will be saved in the database and the user profile will be changed according to the new information provided

REQ-30: Button for changing the password. When pressed, a new page will be shown to the user and he will be asked for his old password, followed by 2 prompts to enter the new password

REQ-31: Button for going back to the profile; When pressed, the user will be taken back to the main profile page

4.4 User bank accounts

4.4.1 Description and Priority

Each user will have the possibility to create multiple bank accounts with different currencies.
Medium priority.

4.4.2 Stimulus/Response Sequences

For each account the user will have to choose the currency type.

4.4.3 Functional Requirements

REQ-32: The button for creating a new bank account; When pressed, the user will be presented with a new panel

REQ-33: In the new panel he will have to choose the currency type for the new account using a choice box

REQ-34: A label for each account created that will contain the full name of the user, the account name, the currency type and a button for adding funds to that specific account

REQ-35: The account name is generated following the next rule: BAT (the name of the application), the year, month and day of the account creation, the user id and the account id

REQ-36: The add funds button will show a pop-up where the user will be asked for the sum of money he wants to add to that specific account

4.5 Transaction page

4.5.1 Description and Priority

The user will be able to transfer a sum of money to another user's account
Medium priority.

4.5.2 Stimulus/Response Sequences

The user must input: destination account, sum of money to be transferred, currency and the account he wants to transfer from

4.5.3 Functional Requirements

REQ-37: Field for destination account; The user must enter the account he wants to transfer money to; The account must be valid, otherwise the transaction will fail

REQ-38: Field for amount to be transferred; The user must enter the amount of money he wants to transfer; The value must be greater than 0 and smaller than $10.000 or the equivalent in other currencies

REQ-39: Choice box for the account to be transferred from; The user must choose from his bank accounts; The funds in the chosen account must be greater or equal with the sum to be transferred

REQ-40: Choice box for the currency; The user must choose a currency for the value entered in the transfer amount field; The value entered in the transfer amount will be converted according to the chosen currency

REQ-41: The amount of funds will be converted according to the money flow at the transaction time

REQ-42: The button for completing the transaction; In order for the transaction to take place, the above requirements must be met; When pressed, a pop-up message will appear

REQ-43: The new pop-up message will ask the user to confirm his transaction

REQ-44: If the transaction is successful, it will be saved in the Transaction History section in the profile page

4.6 Payment page

4.6.1 Description and Priority

The user will be able to pay a sum of money to a specific service.
Low - Medium priority.

4.6.2 Stimulus/Response Sequences

The user must input: source account, amount to be paid, currency and the service he wants to send money to

4.6.3 Functional Requirements

REQ-45: Choice box for selecting the account; The user must choose a bank account from which he wants to send money

REQ-46: Choice box for selecting the service; The user must choose a service he wants to send money to

REQ-47: Choice box for selecting the currency; The user must select the currency for the amount he wants to pay to the service

REQ-48: Choice box for selecting the currency; The user must select the currency for the amount he wants to pay to the service

REQ-49: Field for entering the amount of money; The user must enter the amount of money he wants to pay

REQ-50: The button for confirming the payment; This button will show a pop-up on the screen

REQ-51: The pop-up shown from the pressing of the Pay Button will ask the user for confirmation regarding the payment he is about to make

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Performance depends on the quality of the internet connection or the speed of the computer the application is running on. For more details see 2.4 and 3.4.
The average speed for the tasks that can be executed is 500ms – 1.000ms. The upper limit can reach 5.000ms.

5.2 Safety Requirements

N.A.

5.3 Security Requirements

- *in order to use the services offered by BAT, one must firstly create an account and, for further accesses log in using a username and a password*
- *the user is logged out after exiting the application or after pressing the log out button present on the graphical interface*
- *the username should contain only letters from the English alphabet (from a to z), both lower case and upper case*
- *the password should contain at least 8 characters:*
    a) *letters from the English alphabet (from a to z), both lower case and upper case*
    b) *numbers from 0 to 9*
    c) *characters belonging to the following group !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~*
    d) *the password should contain at least one number, one special character and one uppercase letter*
- *the password will be saved under the form of a hash inside the database in order to ensure an extra layer of security in case of a malicious attempt*
- *once logged in, the user can edit its profile data, make a transaction and use all other facilities offered by the current version of the BAT application*

5.4 Software Quality Attributes

During the development phase of the project, certain coding conventions have been applied in order to confer a consistent set of rules that had to be followed by all developers working on the project.
The following rules have been used and are part of the google java code style:

1. *Source files are encoded in UTF-8*
2. *Wildcard imports, static or otherwise, are not used*
3. *Each top-level class resides in a source file of its own*
4. *Braces are used with if, else, for, do and while statements, even when the body is empty or contains only a single statement*
5. *Each statement is followed by a line break.*
6. *Java code has a column limit of 100 characters. A "character" means any Unicode code point*
7. *Every variable declaration (field or local) declares only one variable: declarations such as int a, b; are not used*

8. *Local variables are not habitually declared at the start of their containing block or block-like construct. Instead, local variables are declared close to the point they are first used (within reason), to minimize their scope. Local variable declarations typically have initializers, or are initialized immediately after declaration*
9. *The square brackets form a part of the type, not the variable: String[] args, not String args[]*
10. *Class and member modifiers, when present, appear in the order recommended by the Java Language Specification: public protected private abstract default static final transient volatile synchronized native strictfp*
11. *Package names are all lowercase, with consecutive words simply concatenated together (no underscores)*
12. *Class names are written in UpperCamelCase*
13. *Test classes are named starting with the name of the class they are testing, and ending with Test*
14. *Method names are written in lowerCamelCase*
15. *Constant names use CONSTANT_CASE: all uppercase letters, with each word separated from the next by a single underscore*
16. *@Override: always used*
17. *It is extremely rare to override Object.finalize*

5.5 Business Rules

N.A.

6. Other Requirements

T.B.A.

Appendix A: Glossary

BAT – Bank-account Administration and Transaction
JDK – Java Development Kit
JRE – Java Runtime Environment
JVM – Java Virtual Machine
IDE – Integrated Development Environment
DSL – Digital Subscriber Line
SVN – Apache Subversion
HTTPS - Secure Hyper Text Transfer Protocol
MODEM – Modulator Demodulator
UTF – Unicode Transformation Format

Appendix B: Analysis Models

T.B.A.

Appendix C: To Be Determined List

N.A.