S.A Bank

Analysis and Design Document

Student: Andrei Tiberiu

**Group: 30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 17.3.2020 | 1.0 | First version | Andrei Tiberiu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 5

III. Elaboration – Iteration 1.2 6

1. Design Model 6

1.1 Dynamic Behavior 6

1.2 Class Design 7

2. Data Model 9

IV. Construction and Transition 9

1. System Testing 9

2. Future improvements 10

VI. Bibliography 10

# Project Specification

S.A. Bank

S.A. Bank is an online banking system, which uses web development for a satisfying modern customer experience. The web app is going to follow the MVC pattern. ASP .NET Core MVC is going to be used. The customer user will be able to do the following operations for managing his banking account: log in, transfer money (in RON or different currency), currency exchange between his banking accounts, request a personalized credit offer and view his transactions.

# Elaboration – Iteration 1.1

# Domain Model

*O imagine care conține captură de ecran

Descriere generată automat*

# Architectural Design

## Conceptual Architecture

The architecute used will respect the MVC (Model View Controller) pattern, which is a widely use architecture for this type of application thanks to being easy to modify and maintain.

## Package Design

O imagine care conține captură de ecran, pasăre

Descriere generată automat

## Component and Deployment Diagrams

# 

.

O imagine care conține captură de ecran

Descriere generată automat

O imagine care conține captură de ecran

Descriere generată automat

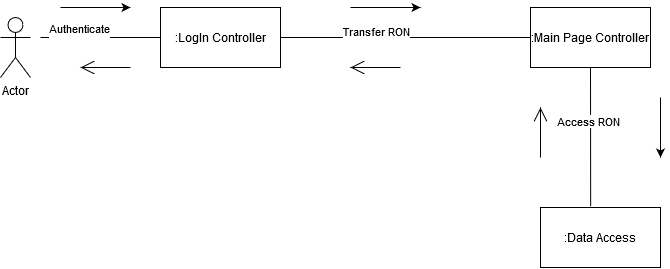
# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

O imagine care conține captură de ecran

Descriere generată automat

**

## Class Design

Factory Method: used to create transactions

O imagine care conține captură de ecran, pasăre

Descriere generată automat

Observer: used to notify the user when transactions take place

O imagine care conține captură de ecran

Descriere generată automat

Mediator: used for communication between AccountRon and AccountEur

O imagine care conține captură de ecran

Descriere generată automat

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

O imagine care conține text, hartă, captură de ecran

Descriere generată automat

# Construction and Transition

# System Testing

Testing scenarios:

1. Log In:

The user is entering his username and password. If the combination is correct the redirection should take place. If not an error message should apeear, describing the reason of failure (user does not exist or wrong data entered)

1. Transfer:

The user selects one of his bank accounts, an amount to be transferred and a target bank account. If the operation is successful, the total amount of both accounts is modified. The actual values will be checked in the database.

1. Loan requests:

The customer user will confirm by pressing a button that he wants a custom loan. After that, we will check if the request has been registered in the database. If so, an employee user will be able to view the request. The data of the user will be inserted into a trained ml model, which will give the employee user a suggestion. After the employee send s a response, the customer should be able to see the received offer.

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography