**Bank Application**

**Student:Florea Laurentiu-Vlad**

**Group:30441**

**Table of Contents**

1. Requirements Analysis 3

1.1 Assignment Specification 3

1.2 Functional Requirements 3

1.3 Non-functional Requirements 3

2. Use-Case Model 3

3. System Architectural Design 3

4. UML Sequence Diagrams 3

5. Class Design 3

6. Data Model 3

7. System Testing 3

8. Bibliography 3

**1. Requirements Analysis**

* **Assignment Specification**

The application is designed both for clients and for bank employees using Swing API. Those are the 2 actors for which the application is designed. Both users have to provide an username and a password in order to use the application.

* **Functional Requirements**

The regular user can perform the following operations:

* Add/update/view client information (name, identity card number, personal numerical code, address, etc.).
* Create/update/delete/view client account (account information: identification number, type, amount of money, date of creation).
* Transfer money between accounts.
* Process utilities bills.

The administrator user can perform the following operations:

* CRUD on employees’ information.
* Generate reports for a particular period containing the activities performed by an employee.

* **Non-functional Requirements**

The data will be stored in a database. Use the Layers architectural pattern to organize your application. Use a domain logic pattern (transaction script or domain model) / a data source hybrid pattern (table module, active record) and a data source pure pattern (table data gateway, row data gateway, data mapper) most suitable for the application

All the inputs of the application will be validated against invalid data before submitting the data and saving it in the database.

**2. Use-Case Model**

Use case: CRUD on clients

Level: user-goal level

Primary actor: administrator

Main success scenario: The client information is modified.

Use case: Generate raport

Level: user-goal level

Primary actor: administrator

Main success scenario: A report is generated for a certain client between some certain dates.

Use case: Generate raport

Level: user-goal level

Primary actor: client

Main success scenario: A report is generated for the client using the application between some certain dates.

Use case: Create/update/view client

Level: user-goal level

Primary actor: administrator

Main success scenario: The information concerning clients is modified.

Use case: Create/update/delete/view client account

Level: user-goal level

Primary actor: administrator

Main success scenario: The information concerning accounts is modified.

Use case: Transfer money between accounts

Level: user-goal level

Primary actor: client

Main success scenario: The information of the two participating accounts are modified.

Use case: Process utilities bills

Level: user-goal level

Primary actor: client

Main success scenario: Bills are processed for a certain client.

**3. System Architectural Design**

**3.1 Architectural Pattern Description**

The required modules for the application:



**3.2 Diagrams**

Package design:



**4. UML Sequence Diagrams**



**5. Class Design**

**5.1 Design Patterns Description**

As design patterns I have used SINGLETON design pattern which is a design pattern that restricts the instantiation of a class to a single, static instance and provides a static point of access to it.

**5.2 UML Class Diagram**



**6. Data Model**



**7. System Testing**

Application texting resumes to data validation, which take place when performing different operations. Among the validations I mention:

* Checking for empty fields when creating a client or an account.
* Checking if user has required amounts in order to perfom the required operation8. Bibliography