Front Desk Bank Application

Analysis and Design Document

Student: Coman Vasile

**Group: 30231**

Table of Content

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 Chira2018

# Assignment Specification

Use JAVA/C# API to design and implement an application for the front desk employees of a bank. The application should have two types of users (a regular user represented by the front desk employee and an administrator user) which have to provide a 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

2. Use-Case Model

*Use case: Create user*

*Level: user-goal level*

*Primary actor: admin*

*Main success scenario: - after the authentication is successfully the admin can add/view all/update/delete employees*

*- after he completing the fields he can add a new employees as admin or regular user*

*- if there are no error messages the new user will be added to database*

*Extensions: then he can choose other operations like see all employees or generate some reports*

*Use case: Process bill*

*Level: user-goal level*

*Primary actor: regular user*

*Main success scenario: - after the authentication is successfully the user can add/view all/update/delete clients and account or process bills or process a transfer*

*- the user needs to select the Transfer button*

*- another form will be opened*

*- he will need to select the refresh button and the data from tables will pe up to date*

*- he will choose from first table and account*

*- he will select the check box for bills*

*- he will insert the amount of money*

*- then will process the pay by pushing Transfer button*

*- in case of bad using a command will appear message dialog*

*Extensions: then he will process others transfers or bills*

*Use case: Generate reports*

*Level: user-goal level*

*Primary actor: admin*

*Main success scenario: - after the authentication is successfully the admin can add/view all/update/delete employees or generate reports*

*- after the fields are completed the admin can choose the date between the report will be generated and he will choose the employee which made the transfers or bill pay*

*- if there are no error messages the admin can push Generate button and see the result of report*

*Extensions: then he can choose other operations like see all employees or generate another report*

3. System Architectural Design

**3.1 Architectural Pattern Description**

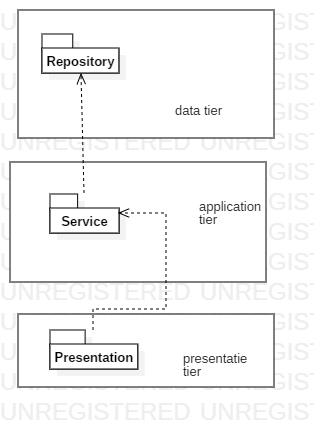
*A 3-tier architecture is a type of software architecture which is composed of three “tiers” or “layers” of logical computing.*

*Presentation Tier- The presentation tier is the frontend layer in the 3-tier system and consists of the user interface.*

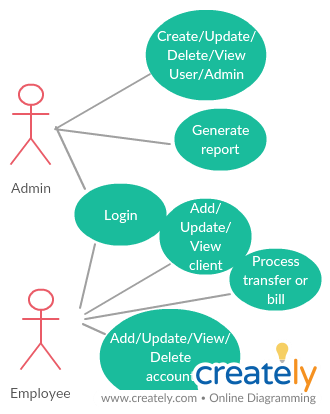
*Application Tier- The application tier contains the functional business logic which drives an application’s core capabilities.*

*Data Tier- The data tier comprises of the database/data storage system and data access layer.*

**3.2 Diagrams**



4. UML Sequence Diagrams



5. Class Design

**5.1 Design Patterns Description**

**5.2 UML Class Diagram**



6. Data Model

*Class User contains information about the employees of the bank. They can be admin or regular user. They have a username and password so they can log in to application.*

*Class Client contains information about the clients added by the bank employees.*

*Class Account contains information about accounts associated to a client.*

*Class Utility contains information about companies which issue bills.*

*Class Bill contains information about a payed bill, source account and destination utility, amount of money, employee which made the transaction.*

*Class Transfer contains information about transactions with source account, destination account, amount of money, employee which made the transaction.*

7. System Testing

*System was tested using the graphic interface by entry different sets of data.*

8. Bibliography

*Labs*

[*https://stackoverflow.com*](https://stackoverflow.com)

[*https://www.w3schools.com*](https://www.w3schools.com)

[*https://docs.oracle.com*](https://docs.oracle.com)