Bank

Student: Pop Laura-Maria

**Group: 30233**

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

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:

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

The administrator user can perform the following operations:

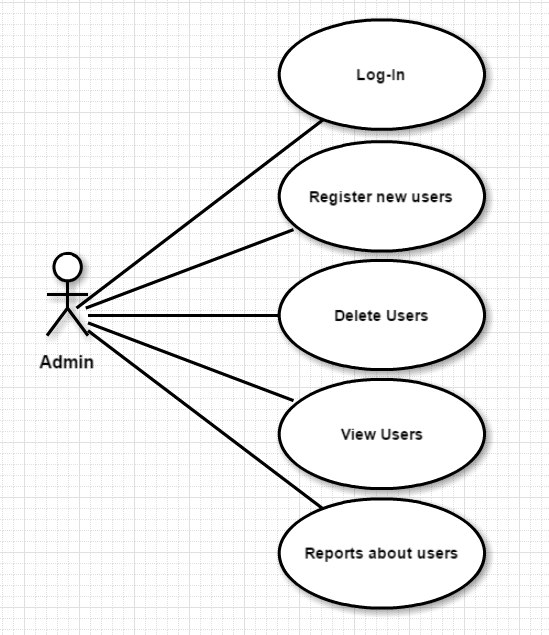
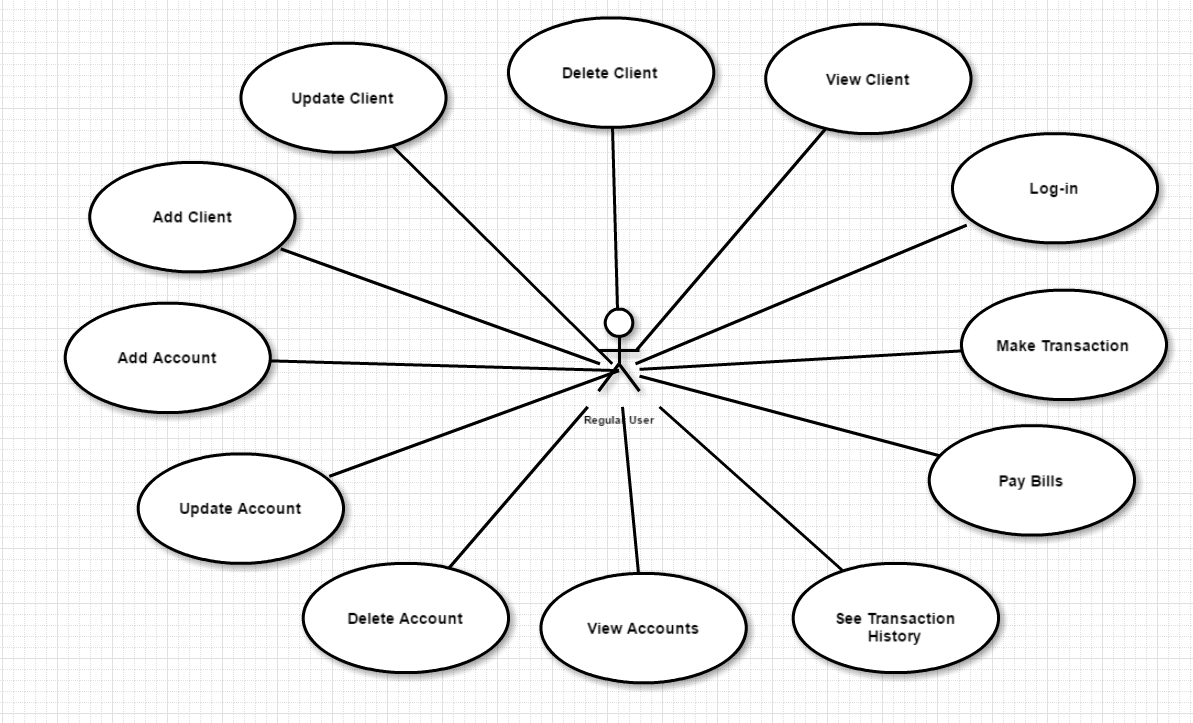
1. CRUD on employees’ information.
2. Generate reports for a particular period containing the activities performed by an employee.

# 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: LOG-IN***

***Level: user-goal***

***Primary actor: both Regular User and Admin***

***Main success scenario: the actor fills in the email and the password fields and if these are correct then the actor will see a menu of actions he/she can perform***

***Extensions: if the actor does not have an account, he will receive one from the Admin (only for Regular User)***

3. System Architectural Design

**3.1 Architectural Pattern Description**

*[Describe briefly the used architectural patterns.]*

**3.2 Diagrams**

*[Create the system’s conceptual architecture; use architectural patterns and describe how they are applied. Create package, component and deployment diagrams]*

4. UML Sequence Diagrams

*[Create a sequence diagram for a relevant scenario.]*

5. Class Design

**5.1 Design Patterns Description**

*[Describe briefly the used design patterns.]*

**5.2 UML Class Diagram**

*[Create the UML Class Diagram and highlight and motivate how the design patterns are used.]*

6. Data Model

*[Present the data models used in the system’s implementation.]*

7. System Testing

*[Present the used testing strategies (unit testing, integration testing, validation testing) and testing methods (data-flow, partitioning, boundary analysis, etc.).]*

8. Bibliography