Assignment A1>

Analysis and Design Document

Student:Baidoc Vlad

**Group:30641**

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 Swing/C# API to design and implement an application for the clients and employees of a

bank. The application should have two types of users: a regular user which would be a client of

the bank and an administrator user represented by an employee of the bank. Both users have to

provide a username and a password in order to use the application.

# Functional Requirements

The regular user can perform the following operations:

- View all accounts (account information: identification number, type, amount of money,

date of creation).

- Transfer money between his/her own accounts.

- Pay utilities bills from his/her own accounts to a predefined list of recipients.

- Generate reports containing account transactions in a particular period.

The administrator user can perform the following operations:

- CRUD on clients information (name, identity card number, personal numerical code,

address, etc.).

- Create/update/delete/view clients' accounts.

- Generate reports for a particular period containing the activities performed by a client.

# Non-functional Requirements

# Availability

# The application is available 24h/day, 365 days/yeah.

# Performance

# In order to have maximum performance, the system must provide a fully functional and friendly user graphic interface. The system needs a local server to store the database and to host the application. This local server, in order to have maximum performance, must provide the medium specification.

**Platform compatibility**

In order to run the application you need a platform that have Java installed.

**Security**

Security of the system will be insure by: each employee and administrator will have a unique username and password.

In order to not damage the system, employees will have different privileges from administrator.

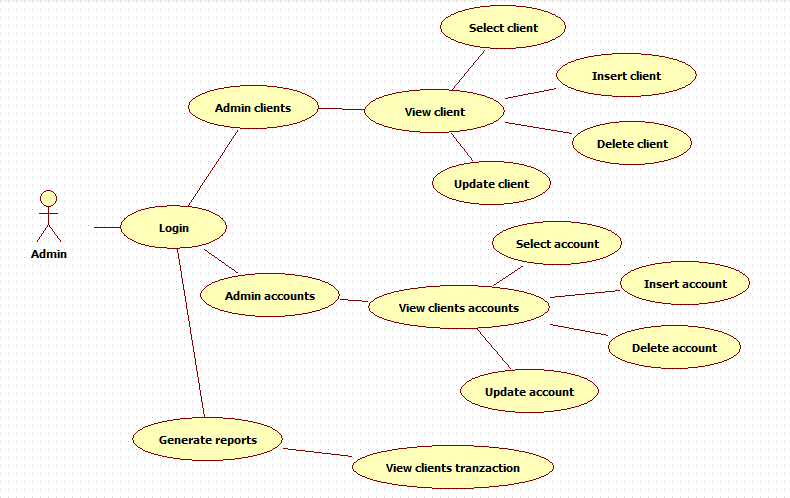
2. Use-Case Model

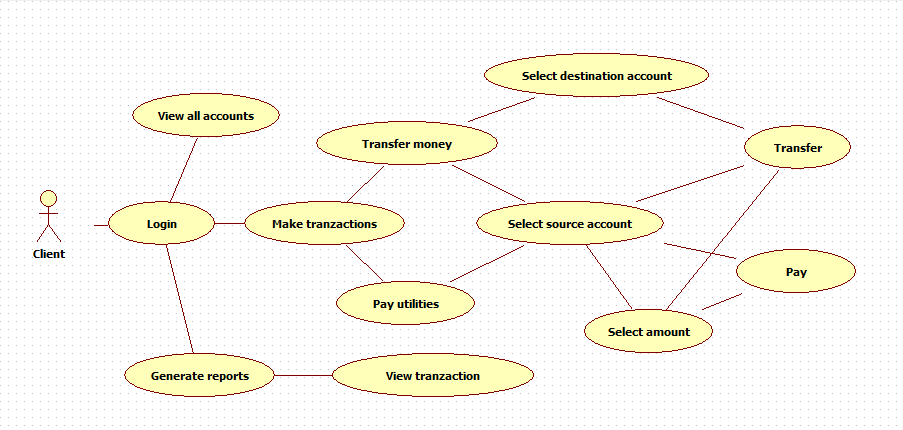
Use case: Transfer money

Level: sub-function

Primary actor: client

Main success scenario: login, view accounts, select account source, select account destination, select amount of money, click on “transfer” button





3. System Architectural Design

**3.1 Architectural Pattern Description**

Three-tier architecture is a client–server architecture in which the user interface (presentation), functional process logic ("domain logic"), computer data storage and data access are developed and maintained as independent modules, most often on separate platforms.

THREE PRINCIPAL LAYERS

Presentation logic

o Interaction with user

o Command-line or rich client or Web interface

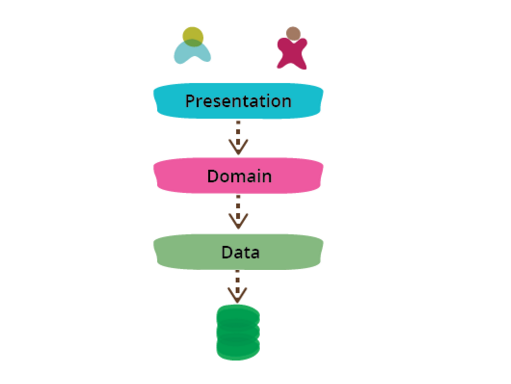
Domain logic

o Validation of input and calculation of results

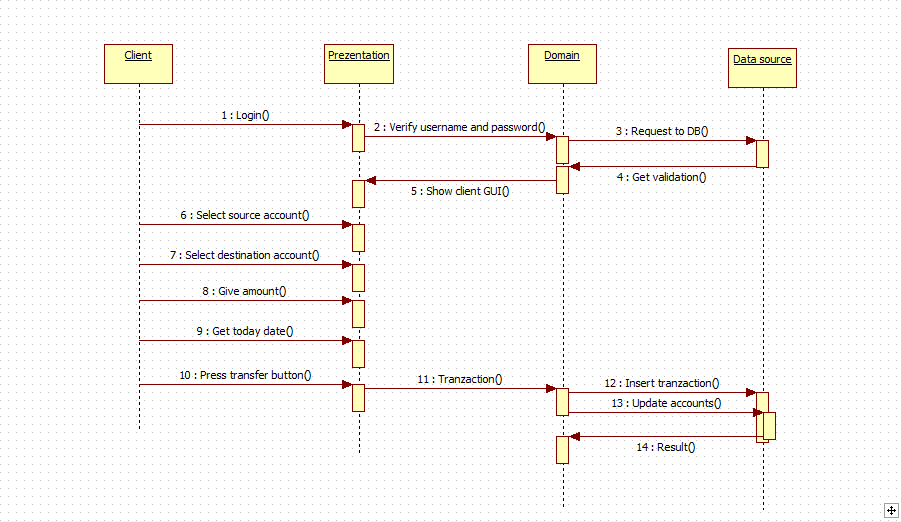
Data source logic

o Communication with database and other applications

**3.2 Diagrams**



4. UML Sequence Diagrams



5. Class Design

**5.1 Design Patterns Description**

Table Module

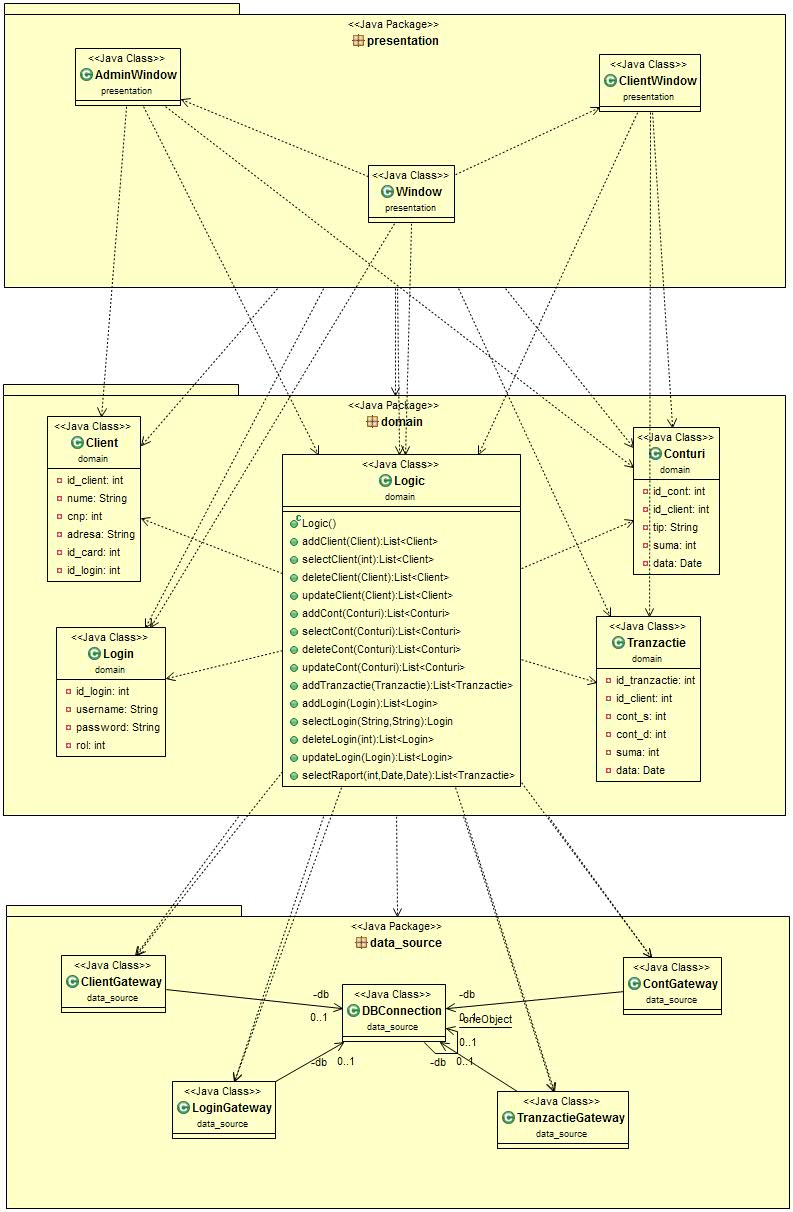
* One class for a table in the database
* A single instance of a class contains the various procedures that will act on data
* Unlike domain model, it does not have inheritance
* Problems: handling complex logic
* Works well with Active Record = Row Data Gateway(RDG) + Domain Logic.

A Table Module is a Domain Logic Pattern in the sense it can contain the BUSINESS LOGIC related to a particular table. A Table Data Gateway is supposed handle Database interface only and is not supposed to contain any Business Logic. Read "database interface" here as "insert/update/delete/read rows from a table".

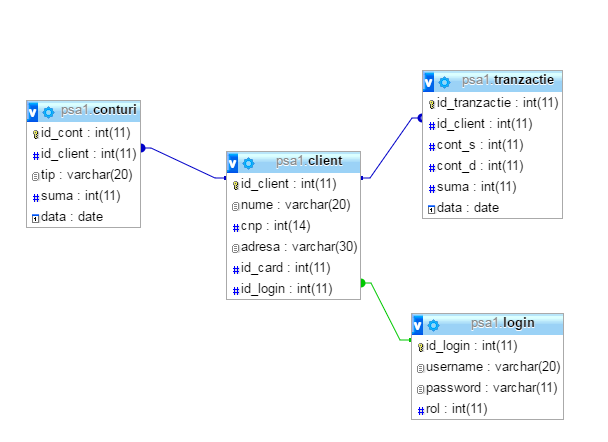
Please note Martin Fowler's definitions which clearly specify this distinction.

A Table Module organizes domain logic with one class per table in the database, and a single instance of a class contains the various procedures that will act on the data. The primary distinction with Domain Model is that, if you have many orders, a Domain Model will have one order object per order while a Table Module will have one object to handle all orders.

**5.2 UML Class Diagram**



6. Data Model

7. System Testing

The application has been tested on various data and is working well and fast. If you give the correct data in the interface the program will work exactly as planed and if you don’t give the correct data an error will be show in the console.

8. Bibliography

<http://docs.oracle.com/javase/tutorial/uiswing/>

<https://martinfowler.com/eaaCatalog/tableModule.htm>

<http://stackoverflow.com/questions/433819/table-module-vs-domain-model>

<http://stackoverflow.com/questions/6072087/table-module-and-table-data-gateway-patterns>