

# **Bank Application Analysis and Design Document**

**Student:Erdei Alexandru Vasile  
Group:30235**

# 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	5
5. Class Design	5
6. Data Model	6
7. System Testing	7
8. Bibliography	7

# 1. Requirements Analysis

## 1.1 Assignment Specification

Proiectarea și implementarea unei aplicații desktop pentru angajații unei bănci. Aplicația trebuie să aibă două tipuri de useri (unul reprezentat de angajații băncii, iar altul administratorul) care se vor identifica printr-un user și o parolă pentru a utiliza aplicația. Aplicația oferă și o securitate pentru utilizatori deoarece sunt nevoiți să introducă și CNP-ul, acesta putând oferi o cale mult mai sigură pentru a afla utilizatorul.

## 1.2 Functional Requirements

### Regular user:

- Adăugare/Actualizare/Vizualizare/Ștergere informații client
- Create/Actualizare/Ștergere/Vizualizare conturi client
- Transfer bani între conturi
- Plata facturi utilități

### Administrator :

- CRUD asupra informațiilor angajaților
- Generare raport în funcție de o perioadă care să conțină informații despre activitatea angajaților

## 1.3 Non-functional Requirements

Securitatea este acoperită prin implementarea log-in cu ajutorul unui nume și o parolă, acestea sunt unice la nivel de utilizator.

Propun ca aplicația să fie performantă, la nivel de utilizare să nu necesite mai mult de 4 secunde tranziția de la o cerință la alta.

Dacă aplicația nu este utilizată în mod corect, acesta își va relua operațiile anterioare, adică nivelul de disponibilitatea va fi ridicat.

# 2. Use-Case Model

### Use-Case description format:

Use case: adăugare cont tip debit

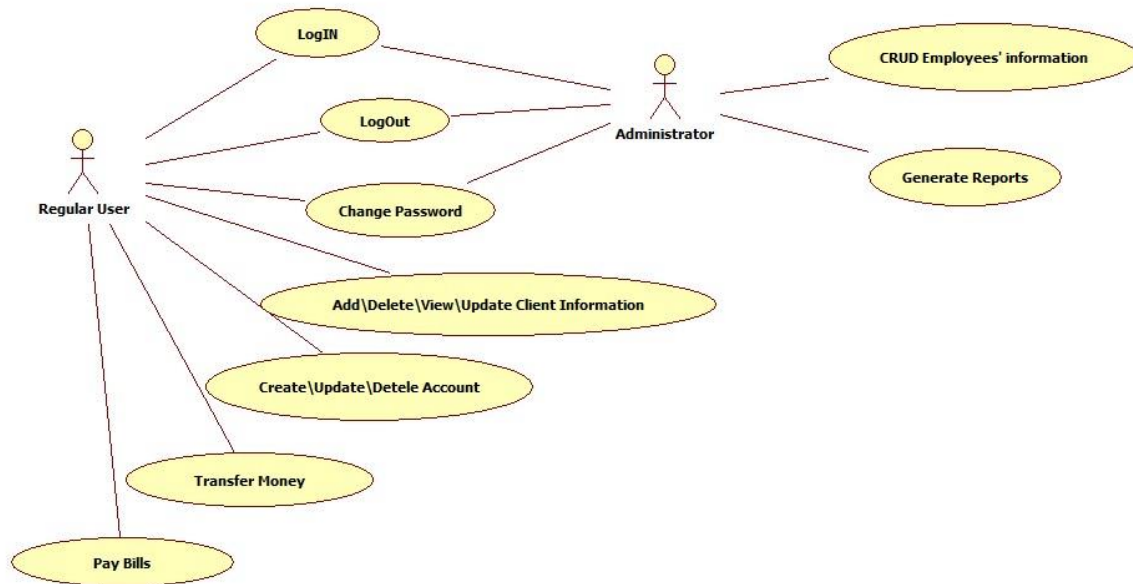
Level: user-goal level

Primary actor: Simple user (Employee bank) Utilizatorul normal

Main success scenario: clientul trimite un e-mail la banca, deoarece dorește să își deschidă un cont, acesta primește un cont după ce primește anunțul de succes prin SMS, asta dacă nu era deja în sistem. Astfel având cont (cu datele personale), se vor insera datele automat de aplicație în baza de date a sistemului, contul de ID se referă la CNP și este unic la fiecare persoană. Astfel clientul va putea crea contul de debit pe care si-l dorește (având cont de client activ).

Extensions: Datele de autentificare ale clientului nu sunt valide => Eroare astfel nu se creează cont

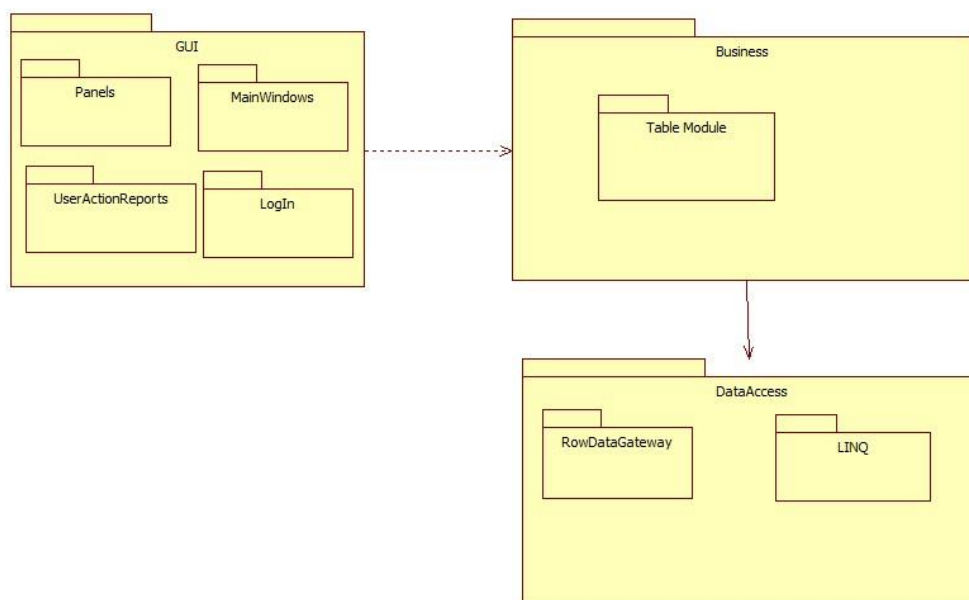
USE – CASE:



### 3. System Architectural Design

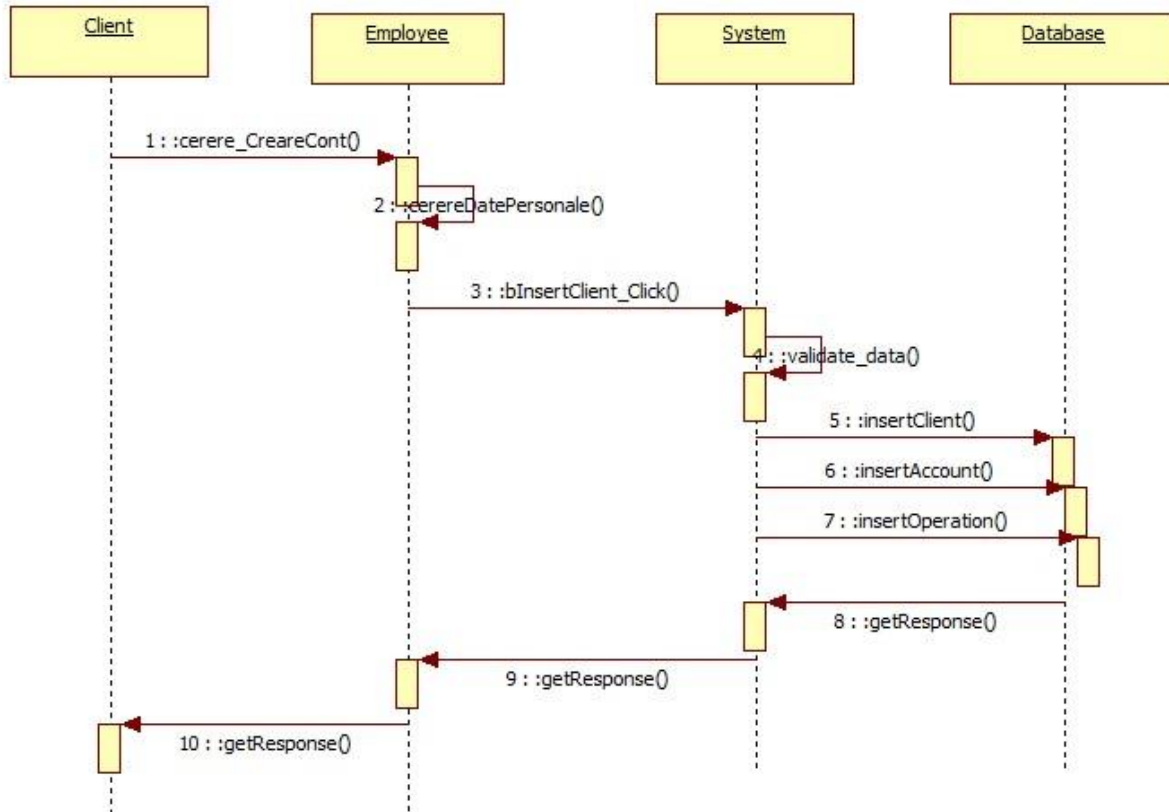
#### 3.1 Architectural Pattern Description

#### 3.2 Diagrams



## 4. UML Sequence Diagrams

Diagrama de secvență pentru adăugare client:



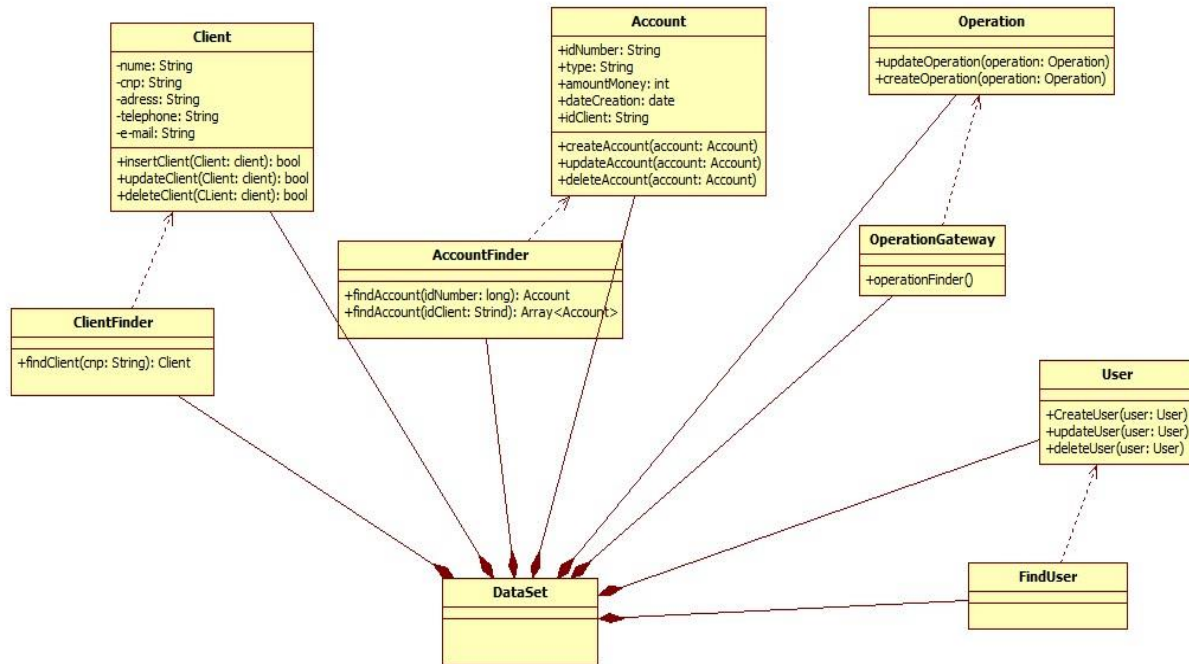
## 5. Class Design

### 5.1 Design Patterns Description

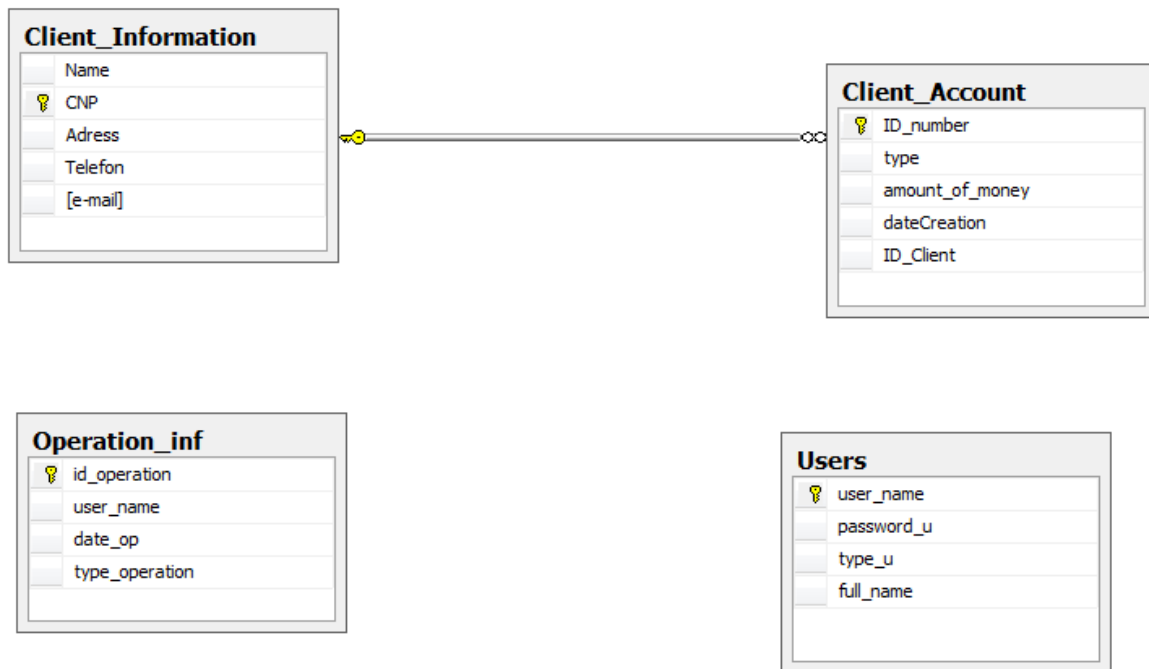
Pattern-ul Data Mapper se utilizează pentru realizarea legăturii între obiecte și baza de date menținându-le totuși independente. Astfel obiectele nu vor ști nici măcar că există o bază de date, nu conțin cod SQL, toate acestea fiind realizate prin intermediul Mapper-ului, odată ce serverul a fost creat și adăugat soluției (conexiunea realizată) Design dbI va crea automat modelul din baza de date.

Pattern-ul Layers este un model structural care ajută la organizarea aplicației care poate fi scoasă în grupuri de cerințe mici, fiecare din ele aflându-se la un anumit nivel de abstracție. Principalele sale probleme sunt legate de structura sa verticală, adică operațiile de nivel înalt se bazează pe cele de nivel mai scăzut. Operațiunile de pe aceleași nivele de abstractizare sunt independente, astfel putem observa structura orizontală. Dacă se aplică principii care să ofere securitate, descompunere pentru componente complexe, performanța va crește.

## 5.2 UML Class Diagram



## 6. Data Model



## 7. System Testing

Au fost testate toate cazurile de utilizare a aplicației, atât în cazuri favorabile, cât și nefavorabile. Operațiile CRUD pentru a testa aplicația rulează fără ca sistemul să dea eșec, adică să realizeze funcții fără ca utilizatorul să fie anunțat.

## 8. Bibliography

- [1] Cursul de Software Design, Mihaela Dinșoreanu
- [2] <https://msdn.microsoft.com/en-us/library/ff647343.aspx>
- [3] <https://code.msdn.microsoft.com/windowsdesktop/Data-Mapper-Design-Pattern-50ad1c32>
- [4] <http://www.telerik.com/>