Aplicatie Bancara

Document de analiza si design

Student:Suciu Vlad

**Groupa :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 4

4. UML Sequence Diagrams 5

5. Class Design 6

6. Data Model 7

7. System Testing 7

8. Bibliography 8

1. Requirements Analysis

# Assignment Specification

Acest assigment a dorit sa ne aduca in prim plan o aplicatie uzitata la o banca , care trebuie implementata in orice limbaj vrem noi.

# Functional Requirements

*Adiminstratorul poate:*

*-CRUD pe angajatii bancii.*

*-sa genereze rapoarte cu privire la o perioada de acitivitate au unui angajat bancar.*

*Angajatul banicii poate:*

*-CRUD pe Clientii bancii.*

*-CRUD pe conutrile clientilor la banca X.*

*-sa faca operatiuni de transfer*

*- sa efectueze plata utilitatilor.*

# Non-functional Requirements

*Cerinte nonfunctionale exista ca peste tot , mai ales in sectorul bancar , aici aduc vorba de securitate in primul rand unde am folosit un sistem de logare//*

*Apoi incapsularea datelor , era absolut necesara , unele date precum cnp ul raman private angajatului bancii etc*

2. Use-Case Model

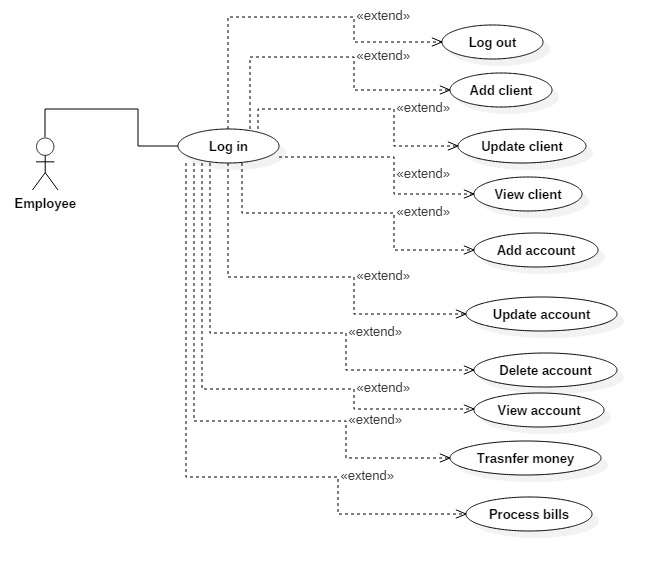
*Use case: plata facturilor*

*Level: user goal level*

*Primary actor: angajatul bancii*

*Main success scenario: succes in plata, fonduri suficiente*

*Extensions: sold insuficient.*



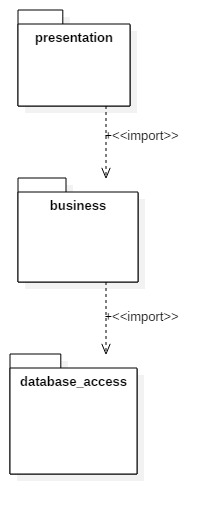
3. System Architectural Design

**3.1 Architectural Pattern Description**

*In prima instanta am folosit 3-tier architecture, deoarece am vrut sa existe o distinctie clara intre operatiile ce au loc, logica aplicatiei si interfata. Cele trei nivele sunt :nivelu de presentare (GUI) , nivelul de lofica si cel de acces la baza de date.*

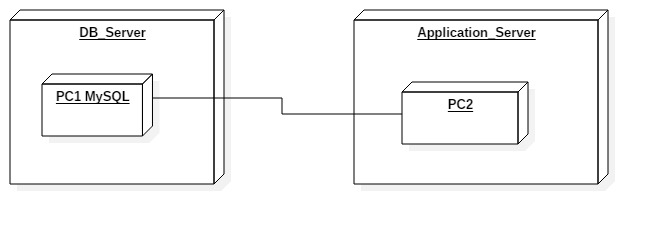
**3.2 Diagrams**

*Cele 3 layere fiecare fiind reprezentat de pachetul lui.*



**Desfasurarea** e basic si s-a realizat cu XAMPP pentru a tine serveru de DB.

//mai tb sa adaug artifactele.



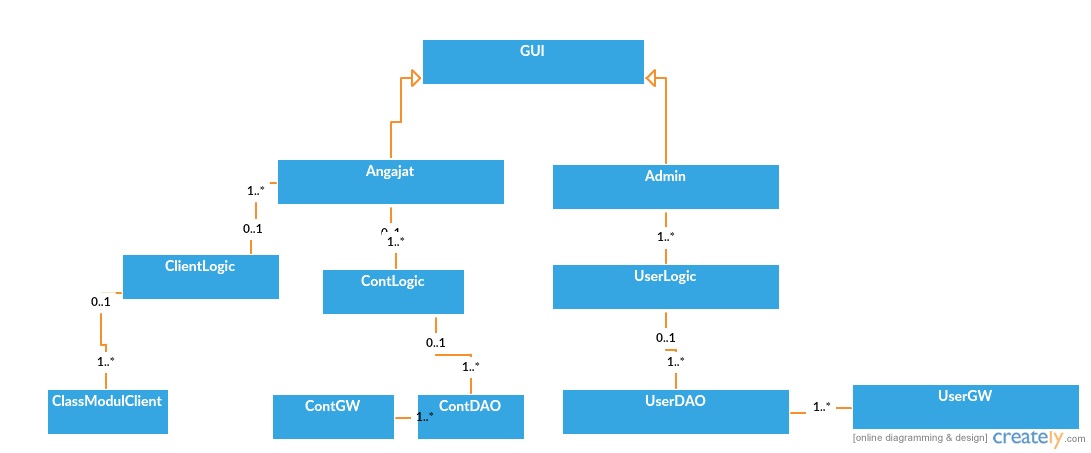
5. Class Design

**5.1 Design Patterns Description**

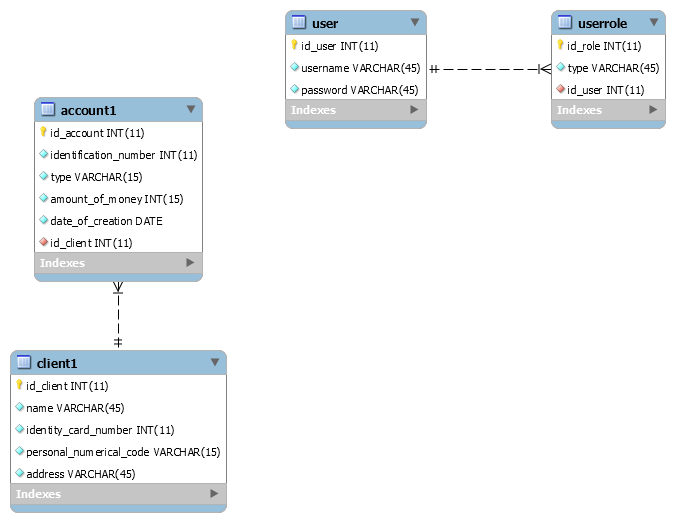
*In aplicatia creata am folosit un sablon de tipuul Data Source hibrid si anume Table Module pentru a-mi facilita accesul la fiecare tabela a bazei de date creand o corespondenta bijectiva.*

*Al doilea sablon folosit e tot din gama Data Source dar e de tip pure si se numeste* Table Data Gateway .

**5.2 UML Class Diagram**

**

6. Data Model .



*In baza mea de date am 4 tabele (pt. clienti , conturi , tipuri si utilizatori).Tabela de conturi are ca si coloane id-ul contului , numarul cardului , tipul contului care e un string , data crearii si suma de banii detinuta. Etc.*

7. System Testing.

Testarea am facut-o atat din cod la implementare cat si din interfata si dupa, nerealizand un sistem de testare cu Junit test. In interfata am afisat mesaje de validare a datelor introduse precum si ascunderea parolei la introducere pentru a nu fi vizibila celor din jur.

8. Bibliography

1.<https://stackoverflow.com/>

2.Martin Fowler et. al, Patterns of Enterprise Application Architecture, Addison Wesley, 2003