Tema 1

Document de analiza si design

Student: Meciu Mihai

Grupa:30238

Cuprins

1.Analiza Cerintei

2.Use-Case Diagram

3. Design architectural al sistemului

4.Diagrame de secventa (UML)

5.Design-ul claselor

6.Model de date

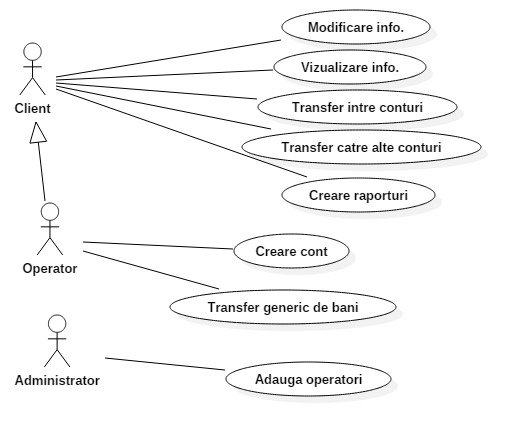
7.Bibliografie

1. Analiza Cerintei
   1. Specificatii

Creati o aplicatie de internet banking. Aplicatia trebuie sa permita logarea a trei tipur diferite de utilizatori: client, operator bancar si administrator. Logarea este necesara pentru a putea folosi aplicatia. Conturile vor fi de trei tipuri: active, deposit, credit.

* 1. Cerinte functionale
     1. Crearea de conturi noi (doar operatori pot executa aceasta actiune);
     2. Modificarea informatiei unui client;
     3. Vizualizarea informatiilor despre cont;
     4. Transfer de bani intre conturi;
     5. Transfer generic de bani intre 2 clienti (doar operatori pot executa aceasta actiune);
     6. Adaugare de operatori(administrator);
     7. Crearea de rapoarte, clientii doar pe anumite perioade de timp, iar operatorii pe unul sau mai multi clienti si intr-o anumita perioada de timp.
  2. Cerinte non-funtionale
     1. Folosirea unei baze de date relationala pentru stocara datelor;
     2. Folosirea pattern-ului Layer pentru organizarea aplicatiei;
     3. Folosirea Domain Logic Pattern
     4. Folosirea pattrn-ului Factory pentru a crea rapoartele atat in format XML cat si JSON.

1. Use-Case Model



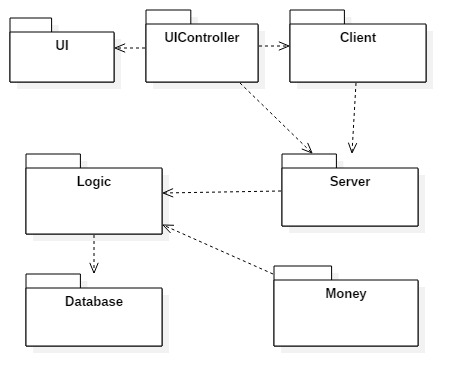
1. Design architectural al sistemului
   1. Descrierea pattern-urilor arhitecturale

In acest proiect am folosit arhitectura Layers, aplicatia este implartita in 3 nivele: nivelul de prezentare ( interfata grafica ), nivelul de logica si cel de operatii asupra bazei de date. Pentru a prelucra datele introduce de utilizator am folosit arhitectura MVC astfel: componentele View si Controller se afla in primul nivel, componentele Controller si Model se afla in al doilea nivel.

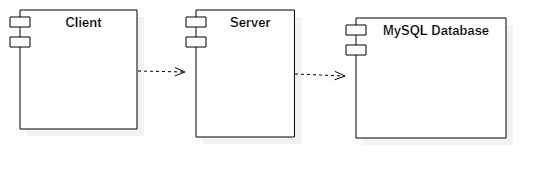
Proiectul a fost modificate pentru a respecta modelul Client-Server.

* 1. Diagrame

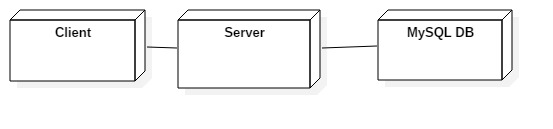
Package diagram:



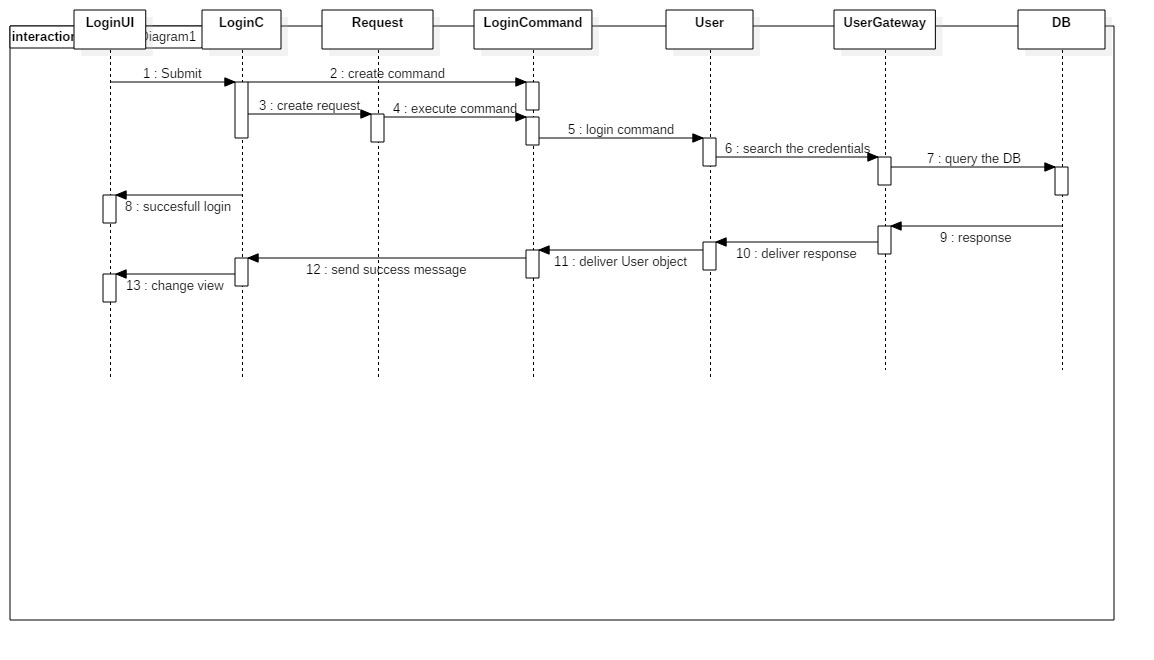
Component diagram:



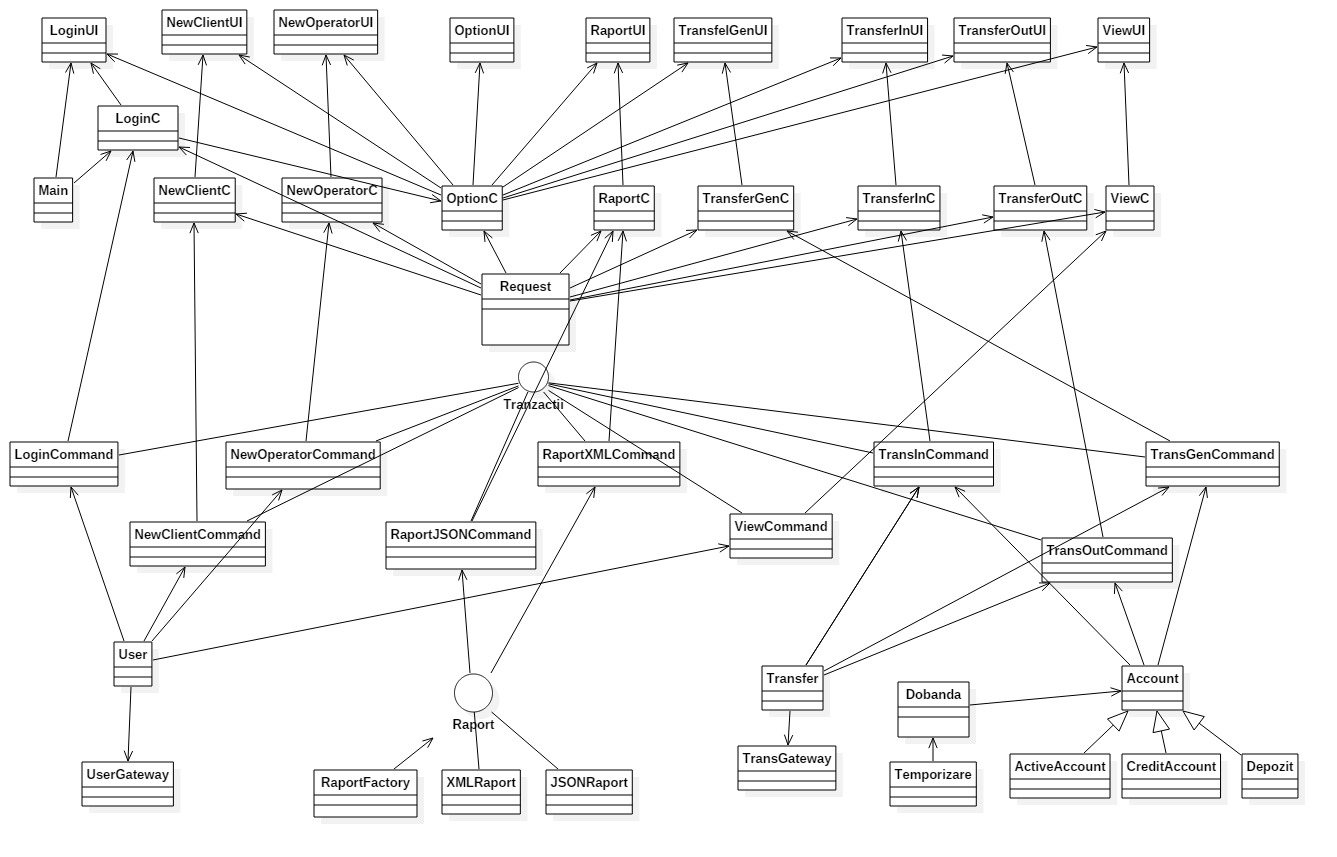
Deployment diagram:



1. Diagrame de secventa (UML)



1. Design-ul claselor
   1. Pentru comunicarea cu baza de date am folosit atat pattern-ul Active Record in clasa Account, cat si Row data gateway pentru a introduce client si operatori noi cat si pentru a tine evidenta transferurilor;
   2. Pentu crearea rapoartelor atat in format XML cat si JSON s-a folosit pattern-ul Factory;
   3. Pentru a respecta modelul Client-Server s-a folosit patter-ul Command pentru a executa toate use case-urile.
   4. UML Class Diagram



1. Model de date

Pentru a comunica cu DB-ul am folosit structurile Statement si ResultSet.

1. Bibliografie

<http://www.w3schools.com/sql/> - sintaxa si comenzi SQL;

<http://www.vogella.com/tutorials/MySQLJava/article.html> - conexiune Java -> SQL;

<https://docs.oracle.com/javase/7/docs/api/overview-summary.html> - clase si metode ale anumitor clase.