Student: Milas Bogdan-Adrian

**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

Aceasta aplicatie este creata si implementata pentru angajatii de la receptia unei banci.Aplicatia are 2 tipuri de utilizatori :un user obisnuit reprezentat de angajatii bancii si un administrator.Ambii vor trebui sa furnizeze un nume si o parola pentru a putea folosi aplicatia.

# Functional Requirements

User-ul obisnuit va trebui sa efectueze urmatoarele operatii: adaugare ,updatare si vizualizare a informatiilor despre un client .Clientul se identifica prin :nume ,card de identitate ,con numeric personal si adresa.

Va putea efectua operatiile CRUD(create-read-update-delete) asupra conturilor clientilor.Un con teste identificat printr-un id , tip ,suma de bani si data creatiei.

Alte functii care vor mai putea fi efectuate de un user obisnuit ar fi transferal de bani dintre conturi si proces de gestionare al facturilor la utilitati.

Administratorul va putea efectua operatiile CRUD asupra angajatilor si va putea genera un raport de activitate pentru o anumita perioada pentru fiecare angajat.

# Non-functional Requirements

Inca nu am reusit sa fac legatura cu interfata grafica .

2. Use-Case Model

**

*Use-Case description format:*

Use case: Log in

Level: user-goal level

Primary actor: Regular user

Main success scenario: primeste un Username si o parola si apasa butonul de Connect

Extensions: actorii nu au conturi.

*]*

3. System Architectural Design

**3.1 Architectural Pattern Description**

Pentru realizarea acestui proiect am folosit Layered Architectural Pattern cu 3layers-uri principale:

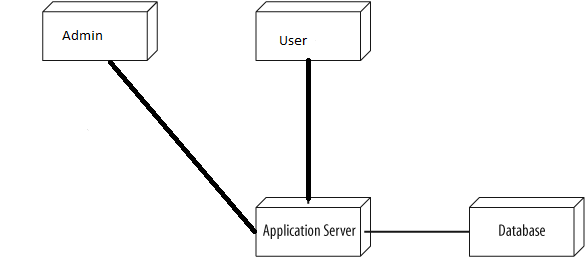
* Bussines : folosit pentru logica aplicatiei
* DataAccess : folosit pentru realizarea conexiunii cu baza de date
* Presentation : folosit pentru realizarea interfetei grafice

**3.2 Diagrams**

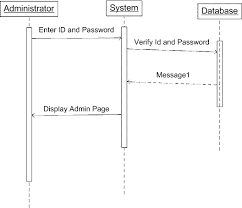
**PACKAGE Diagram**

****

**DEPLOYMENT Diagram**

****

4. UML Sequence Diagrams



5. Class Design

**5.1 Design Patterns Description**

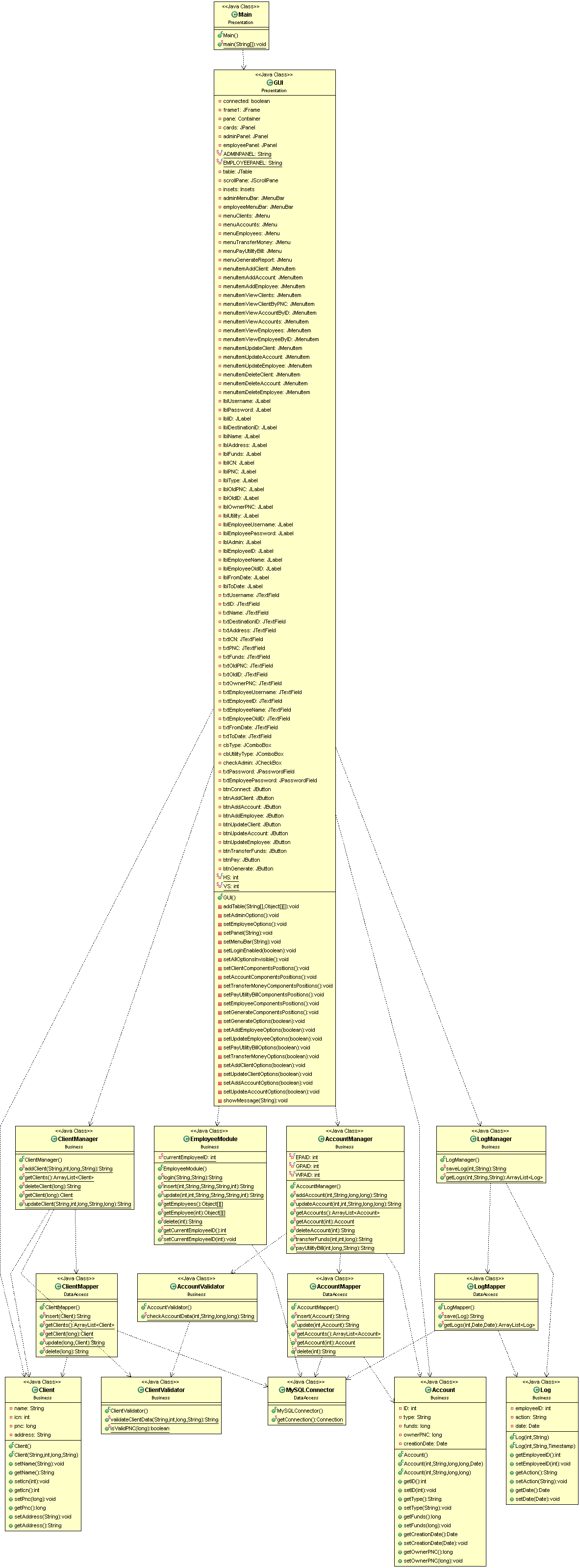
-data mapper

A Data Mapper is a Data Access Layer that performs bidirectional transfer of data between a persistent data store (often a relational database) and an in-memory data representation (the domain layer). The goal of the pattern is to keep the in-memory representation and the persistent data store independent of each other and the data mapper itself. The layer is composed of one or more mappers (or Data Access Objects), performing the data transfer.

-table module

A Table Module organizes domain logic with one class per table in the data-base, and a single instance of a class contains the various procedures that will act on the data

**5.2 UML Class Diagram**



6. Data Model

Baza mea de date are urmatoarele tabele: clients , accounts ,employees ,logs.**Accounts** are campurile ID,type,funds,ownerPNC si creationDate. **Clients** are campurile name ,ICN, PNC, si address .**Employees**:ID,Name,username,password si admin iar tabelul **logs** are campurile employeeID,action si date.

7. System Testing

Pentru testare am implementat metode care verifica ca suma detinuta de un client intr-un cont sa nu fie mai mica de 0 si ca CNP-ul sa fie format din 13 cifre.

8. Bibliography

<http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html>

<http://www.codejava.net/java-se/jdbc/jdbc-tutorial-sql-insert-select-update-and-delete-examples>

<http://www.codejava.net/java-se/jdbc/jdbc-tutorial-sql-insert-select-update-and-delete-examples>

<https://www.tutorialspoint.com/jdbc/jdbc-db-connections.htm>

<https://www.youtube.com/watch?v=BCqW5XwtJxY>

<https://www.w3schools.com/sql/sql_select.asp>

<http://stackoverflow.com/questions/19415170/what-is-setbounds-and-how-do-i-use-it>

<http://stackoverflow.com/questions/1081486/setting-background-color-for-the-jframe>

<https://www.youtube.com/watch?v=v_bYm091w3g>

<https://en.wikipedia.org/wiki/Table_(database)>