A logo with black and red text

Description automatically generated

**FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE**

**DEPARTAMENTUL CALCULATOARE**

**PROIECT**

la disciplina

BAZE DE DATE

**Titlu**

**GESTIUNEA UNUI LANȚ DE POLICLINICI**

**Mesaroș Radu-Andrei**

**Samoilă Denis-Valentin**

**Silaghi Andrei**

**An academic: 2023-2024**

A logo with black and red text

Description automatically generated

**Cuprins**

1. Introducere
2. Functionalitati principale
3. Modelul de date
   1. Schema UML restransa
   2. Schema UML desfasurata
4. Tabele distincte pentru elementele programabile
   1. Tabel proceduri
   2. Tabel functii
   3. Tabel proceduri si functii
5. Scheletul aplicatiei
6. Interfata
   1. Autentificare
   2. Adauga Angajat
   3. Adauga Medic
   4. Adauga Asistent
   5. Pagina Administrator
   6. Analize
   7. Asistent
   8. Contabil
   9. Detalii Angajat
   10. Inspector
   11. Modifica Angajat
   12. Orar Lunar
   13. Orar Saptamanal
   14. Form Proceduri
   15. Rapoarte Medici
   16. Receptioner

A logo with black and red text

Description automatically generated

**1. Introducere**

Aplicatia care urmeaza a fi prezentata isi propune gestionarea activitatilor dintr-un lant de policlinici, fiind format din mai multe unitati medicale. Dezvoltarea si implementarea acestei aplicatii aduce o serie de beneficii semnificative, cum ar fi: eficientizarea gestionarii policlinicilor, integrarea si centralizarea datelor intr-o singura baza de date, controlul accesului pe baza de roluri, imbunatatirea experientei utilizatorilor prin interfata grafica, adaptibilitate si scalabilitate, permitand adaugarea de noi functionalitati si module pe masura ce nevoile policlinicilor se schimba.

Cunostintele necesare sunt:

* programarea in limbajul Java;
* manipularea bazelor de date folosind MySQL;
* folosirea API-ului Java DataBase Connectivity;

A logo with black and red text

Description automatically generated

**2. Functionalitati principale**

Gestiunea angajatilor

-un utilizator de tip administrator poate adauga, modifica si sterge informatii in baza de date legate de utilizatori, operand in departamentele resurse umane, financiar-contabil, si administrarea operatiilor curente din cadrul policlinicii

-aplicatia poate fi accesata, pe baza unui proces de autentificare, de catre mai multe tipuri de utilizatori

Servicii medicale

-lantul de policlinici ofera pacientilor un set de servicii medicale, creand programari pentru pacienti

Rapoarte medicale

-medicii ofera rapoarte medicale pacientilor dupa ce au fost consultati

Administrare contabilitate

-se ofera raportul de venituri/cheltuieli in anumite perioade de timp (lunare)

Pentru intreaga lista de functionalitati se recomanda parcurgerea integrala a proiectului.

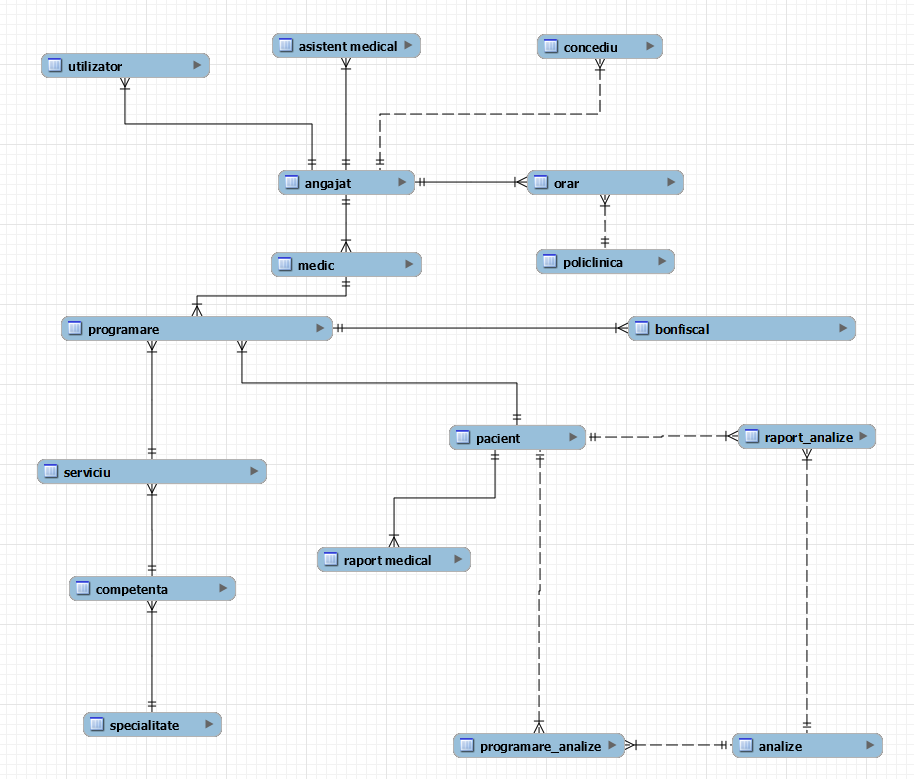
A logo with black and red text

Description automatically generated

**3. Modelul de date – UML**

**3.1 Schema UML restransa**

Modelul de date, proiectat in MySQL, are urmatoarea schema restransa:



A logo with black and red text

Description automatically generated

**3.2 Schema UML desfasurata**

Modelul de date, proiectat in MySQL, are urmatoarea schema desfasurata:

A computer screen shot of a computer

Description automatically generated

A logo with black and red text

Description automatically generated

**4. Tabele distincte pentru elemente programabile**

**4.1 Tabel proceduri**

**A computer screen shot of a code

Description automatically generated**

**4.2 Tabel functiiA computer screen shot of a computer code

Description automatically generated**

A logo with black and red text

Description automatically generated

**4.3 Tabel procedure si functii**

**A computer screen shot of a code

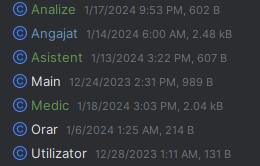
Description automatically generated**

A logo with black and red text

Description automatically generated

**5. Scheletul aplicatiei**

Clasele principale ale aplicatiei sunt:



Scheletul codului pentru aceste clase:

import java.sql.Date;  
public class Angajat {  
 public Integer idangajat;  
 public String cnp;  
 public String nume;  
 public String prenume;  
 public String adresa;  
 public String email;  
 public String telefon;  
 public String iban;  
 public Date dataAngajarii;  
 public String functia;  
 public Integer salariu;  
 public Integer nrOre;  
 private Integer procentAditional;  
 public void setProcentAditional(Integer procentAditional) {  
 this.procentAditional = procentAditional; }  
 public String getNume(){  
 return nume; }  
 public Integer getIdangajat() {  
 return idangajat; }  
 public String getCnp() {  
 return cnp; }  
 public String getPrenume() {  
 return prenume;  
 }  
 public String getAdresa() {  
 return adresa;  
 }  
 public String getEmail() {  
 return email;  
 }  
  
 public String getTelefon() {  
 return telefon;  
 }  
A logo with black and red text

Description automatically generated public String getIban() {  
 return iban;  
 }  
 public Date getDataAngajarii() {  
 return dataAngajarii;  
 }  
 public String getFunctia() {  
 return functia;  
 }  
 public Integer getSalariu() {  
 return salariu;  
 }  
 public void setIdangajat(Integer idangajat) {  
 this.idangajat = idangajat;  
 }  
 public void setCnp(String cnp) {  
 this.cnp = cnp;  
 }  
 public void setNume(String nume) {  
 this.nume = nume;  
 }  
 public void setPrenume(String prenume) {  
 this.prenume = prenume;  
 }  
 public void setAdresa(String adresa) {  
 this.adresa = adresa;  
 }  
 public void setEmail(String email) {  
 this.email = email;  
 }  
 public void setTelefon(String telefon) {  
 this.telefon = telefon;  
 }  
 public void setIban(String iban) {  
 this.iban = iban;  
 }  
 public void setDataAngajarii(Date dataAngajarii) {  
 this.dataAngajarii = dataAngajarii;  
 }  
 public void setFunctia(String functia) {  
 this.functia = functia;  
 }  
 public void setSalariu(Integer salariu) {  
 this.salariu = salariu;  
 }  
 public Integer getNrOre() {  
 return nrOre;  
 }  
 public void setNrOre(Integer nrOre) {

this.nrOre = nrOre;  
 }  
 @Override  
 public String toString() {  
 return nume + " " + prenume; }

}

A logo with black and red text

Description automatically generated

Scheletul pentru interfete este urmatorul:

import javax.swing.\*;  
import java.awt.\*;  
import java.awt.event.ActionEvent;  
import java.awt.event.ActionListener;  
  
public class DetaliiAngajat extends JFrame {  
  
 private JPanel topPanel;  
 private JButton logOutButton;  
 private JButton homeButton;  
 private JLabel numeLabel;  
 private JLabel numeLabel1;  
 private JLabel prenumeLabel;  
 private JLabel cnpLabel;  
 private JLabel prenumeLabel1;  
 private JLabel adresaLabel;  
 private JLabel cnpLabel1;  
 private JLabel adresaLabel1;  
 private JLabel emailLabel;  
 private JLabel emailLabel1;  
 private JLabel telefonLabel1;  
 private JLabel telefonLabel;  
 private JLabel ibanLable;  
 private JLabel ibanLabel1;  
 private JLabel dataLabe;  
 private JLabel dataLabel1;  
 private JLabel functiaLabel;  
 private JLabel functiaLabel1;  
 private JLabel salariuLabel;  
 private JLabel salariuLabel1;  
 private JLabel nrOreLabel;  
 private JLabel nrOreLabel1;  
 private JPanel detailsPanel;  
  
 public DetaliiAngajat(Angajat angajat){  
 setTitle("AdaugaAngajatForm");  
 setContentPane(detailsPanel);  
 setMinimumSize(new Dimension(500, 450));  
 setSize(500, 450);  
 setDefaultCloseOperation(WindowConstants.*EXIT\_ON\_CLOSE*);  
  
 numeLabel1.setText(angajat.getNume());  
 prenumeLabel1.setText(angajat.getPrenume());  
 cnpLabel1.setText(angajat.getCnp());  
 adresaLabel1.setText(angajat.getAdresa());  
 emailLabel1.setText(angajat.getEmail());  
 telefonLabel1.setText(angajat.getTelefon());  
 ibanLabel1.setText(angajat.getIban());  
 dataLabel1.setText(String.*valueOf*(angajat.getDataAngajarii()));  
 functiaLabel1.setText(angajat.getFunctia());  
 salariuLabel1.setText(String.*valueOf*(angajat.getSalariu()));  
 nrOreLabel1.setText(String.*valueOf*(angajat.getNrOre()));

A logo with black and red text

Description automatically generated

homeButton.setFocusPainted(false);  
 homeButton.setBorderPainted(false);  
  
 logOutButton.setFocusPainted(false);  
 logOutButton.setBorderPainted(false);  
  
 homeButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 dispose();  
 if(angajat.getFunctia().equals("administrator")){  
 AdministratorForm newAdmin = new AdministratorForm(angajat);  
 }  
 if(angajat.getFunctia().equals("inspector resurse umane")){  
 InspectorForm2 newInspector = new InspectorForm2(angajat);  
 }  
 if(angajat.getFunctia().equals("expert financiar contabil")){  
 ContabilForm newContabil = new ContabilForm(angajat);  
 }  
 }  
 });  
  
 logOutButton.addActionListener(new ActionListener() {  
 @Override  
 public void actionPerformed(ActionEvent e) {  
 dispose();  
 Autentificare autentificare = new Autentificare(null);  
 }  
 });  
  
 setVisible(true);  
 }  
 public static void main(String[] args) {  
 SwingUtilities.*invokeLater*(new Runnable() {  
 @Override  
 public void run() {  
 try {  
 // Opțional: setează look and feel-ul sistemului  
 UIManager.*setLookAndFeel*(UIManager.*getSystemLookAndFeelClassName*());  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 Angajat angajat = new Angajat();  
 new DetaliiAngajat(angajat).setVisible(true);  
 }  
 });  
 }  
}

A logo with black and red text

Description automatically generated**6. Interfata**

**6.1 Autentificare**

**A screenshot of a computer

Description automatically generated**

**6.2 Adauga angajat**

**A screenshot of a computer

Description automatically generated**

A logo with black and red text

Description automatically generated

**6.3 Adauga Medic**

**A screenshot of a computer

Description automatically generated**

**6.4 Adauga Asistent**

**A screenshot of a computer

Description automatically generated**

A logo with black and red text

Description automatically generated

**6.5 Pagina Administrator**

**A screenshot of a computer

Description automatically generated**

**6.6 Analize**

**A screenshot of a computer

Description automatically generated**

A logo with black and red text

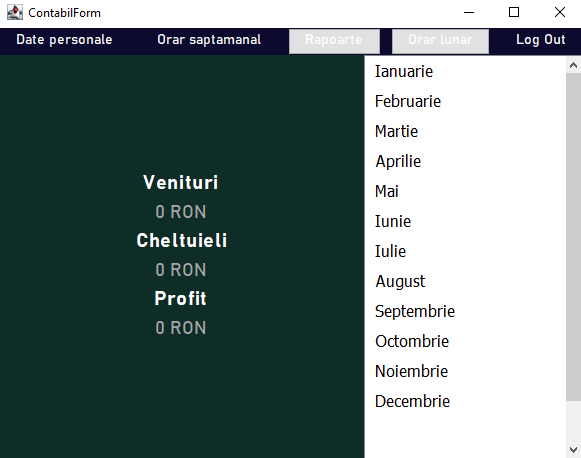
Description automatically generated

**6.7 Asistent**

**A screenshot of a computer

Description automatically generated**

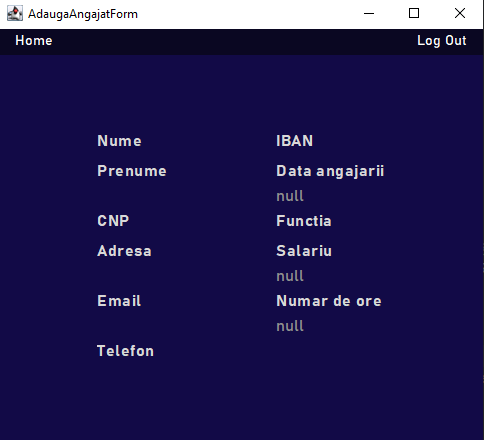
**6.8 Contabil**

****

A logo with black and red text

Description automatically generated

**6.9 Detalii Angajat**

****

**6.10 Inspector**

**A screenshot of a computer

Description automatically generated**

A logo with black and red text

Description automatically generated

**6.11 Modifica Angajat**

**A screenshot of a computer

Description automatically generated**

**6.12 Orar Lunar**

**A screenshot of a computer

Description automatically generated**

A logo with black and red text

Description automatically generated

**6.13 Orar Saptamanal**

**A screenshot of a computer

Description automatically generated**

**6.14 Form Proceduri**

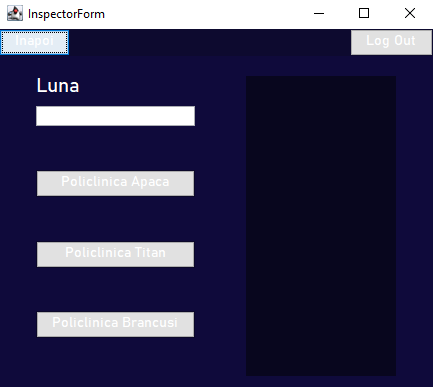
**A screenshot of a computer

Description automatically generated**

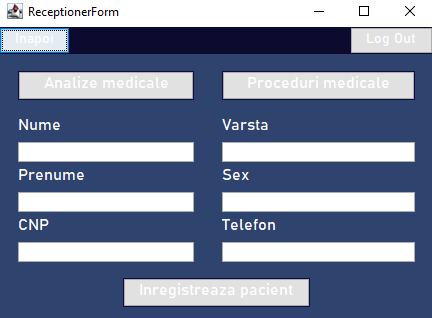
A logo with black and red text

Description automatically generated

**6.15 Rapoarte Medici**

****

**6.15 Receptioner**

****