PROIECT
LANȚ DE POLICLINICI

Disciplina “Introducere în baze de date”

Realizat de: Birle Silviu Adrian
Catargiu Adriana
Vilcelean Dalia

Facultatea de Automatica și Calculatoare
Anul II
Grupa 30222

Profesor coordonator: As. Ing. Cosmina Ivan

Cuprins

1. Introducere

O **bază de date**, uneori numită și **bancă de date** (abreviat **BD**), reprezintă o modalitate de stocare a unor informații și date pe un suport extern (un dispozitiv de stocare), cu posibilitatea extinderii ușoare și a regăsirii rapide a acestora.

De obicei o bază de date este memorată într-unul sau mai multe fișiere. Bazele de date sunt manipulate cu ajutorul sistemelor de gestiune a bazelor de date

Cel mai răspândit tip de baze de date este cel relațional, în care datele sunt memorate în tabele. Pe lânga tabele, o bază de date relațională mai poate conține: indecși, proceduri stocate, declanșatori, utilizatori și grupuri de utilizatori, tipuri de date, mecanisme de securitate și de gestiune a tranzacțiilor etc.

In proiectul nostru am incercat sa implementam o baza de date care ajuta la usurarea munca angajatilor si utilizatorilor unui lant de policlinici, pentru a reduce timpul de cautare a anumitor date personale prin multiple bibliorafturi sau dosare. Acum, aceste date fiind stocate intr-o baza de date condusa de un Administrator care le poate controla oricand.

Cu alte cuvinte, printr-o simpla cautare a numelui sau a CNP-ului, programul va afisa tot ce este nevoie.

Acest proiect dezvolta o aplicatie cu o interfata in limbajul JAVA, iar in MySQL se vor retine si prelucra datele.

Se vor prezenta conceptele folosite, modelul de date, procedure din limbajul MySQL, interfata din JAVA si posibilele implementari ulterioare ale aplicatiei.

2. Specificatie Proiect

Se doreşte implementarea unui sistem informatic destinat gestiunii activităţilor dintr-un lanț de policlinici.

Lanțul de policlinici este format din mai multe unități medicale, fiecare fiind caracterizată prin denumire, adresă, descrierea serviciilor oferite și programul de funcționare, pentru fiecare zi a săptămânii.

Aplicaţia va trebui să utilizeze un sistem de gestiune pentru baze de date MySQL, iar interacţiunea cu acesta va fi realizată doar prin interfaţa grafică . Funcţionalităţile pe care le va oferi programul vizează operaţii ce ţin de gestiunea angajaţilor, serviciul financiar-contabil şi administrarea operațiilor curente din cadrul policilinicii (gestiunea pacienților programați, completarea unui raport medical, emiterea unui bon fiscal).

Aplicaţia va putea fi accesată, pe baza unui proces de autentificare, de către mai multe tipuri de utilizatori, operând în departamentele resurse umane, financiar-contabil sau medical. Pentru fiecare tip de utilizator se vor reţine informaţii precum CNP, nume, prenume, adresa, număr de telefon, email, cont IBAN, numărul de contract, data angajării, funcţia deţinută în cadrul lanțului de policlinici. Fiecare utilizator își va putea vizualiza datele personale imediat după ce va accesa sistemul informatic, fără a avea însă posibilitatea de a le modifica. Totodată, programul trebuie să ofere și o funcționalitate pentru deautentificare, prin care se revine la fereastra care solicită datele de acces, astfel încât

și un alt utilizator să îl poată folosi ulterior, fără a fi necesară repornirea sa.

Utilizatorul de tip **administrator** poate adăuga, modifica şi şterge informaţii în baza de date legate de utilizatori. De asemenea, va exista şi un rol de **super-administrator** care poate opera inclusiv asupra utilizatorilor de tip administrator.

Pentru un utilizator de tip **angajat** se va reține salariul negociat și numărul de ore care trebuie realizat în fiecare lună. Funcțiile ce pot fi deținute în cadrul lanțului de policlinici sunt inspector resurse umane, expert financiar-contabil, recepționer, asistent medical și medic, corespunzătoare departamentelor de resurse umane, economic, respectiv medical.

Pentru un **asistent medical** se va reține suplimentar tipul2 și gradul3.

Pentru un **medic** se va reține suplimentar specialitatea sau specialitățile în care își desfășoară activitatea, gradul4, codul de parafă, competențele pe care le deține pentru realizarea unor proceduri ce necesită acreditări speciale4, titlul științific5, postul didactic6Totodată, fiecare medic are negociat un procent din serviciile medicale realizate care îi revine, adiţional față de salariul negociat.

Lanțul de policlinici oferă pacienților un set de servicii medicale. O parte dintre acestea sunt disponibile pentru toate specialitățile (consultație, distinctă în funcție de gradul medicului care o realizează: specialist, primar, profesor / conferențiar), altele sunt specifice pentru fiecare specialitate în parte. Pentru un serviciu medical disponibil se va specifica specialitatea din care face parte,

necesitatea existenței unei competențe a medicului care o efectuează, prețul asociat și durata (exprimată în minute).

Sistemul va fi format din mai multe module care vor putea fi accesate de angajaţi în funcţie de drepturile pe care le deţin. Astfel, vor fi implementate un **modul pentru gestiunea resurselor umane** ce vizează gestiunea programului de lucru şi al concediilor angajaţilor, un **modul pentru operaţii financiar-contabile** care determină profitul operaţional ca diferenţă între venituri (sume încasate pentru serviciile medicale) şi cheltuieli (plăţi efectuate către angajaţi aferente salariilor) și un **modul pentru gestiunea activităților operaționale** (programarea pacienților pentru servicii medicale și înregistrarea acestora în momentul în care se prezintă în clinica medicală, emiterea bonului fiscal de către recepționeri, completarea rapoartelor medicale de către asistenții medicali și medici).

Toate aceste module vor fi integrate în cadrul aceluiași sistem informatic, sub forma unor meniuri care vor conține funcționalitățile pe care acestea le oferă, disponibilitatea lor fiind însă limitată și de permisiunile pe care le posedă utilizatorul autentificat la momentul respectiv de timp. Drepturile de acces ale angajaţilordin diferite departamente la modulele sistemului sunt descrise în tabelul de mai jos:

departament / tip angajat / modul	resurse umane	economic	medical		
			recepționer	asistent medical	medic
gestiunea resurselor umane	✓	⚠ doar date referitoare la propria persoană	⚠ doar date referitoare la propria persoană	⚠ doar date referitoare la propria persoană	⚠ doar date referitoare la propria persoană
operații financiar contabile	⚠ doar date referitoare la propria persoană	✓	⚠ doar date referitoare la propria persoană	⚠ doar date referitoare la propria persoană	⚠ doar date referitoare la propria persoană + profitul propriu
gestiunea activităților operaționale	✗	✗	✓ doar submodulele programare, înregistrare pacient, emitere bon fiscal	✓ doar submodulul raport medical analize	✓ doar submodulele istoric și raport medical

Drepturile de acces ale angajaţilor aparţinând unui departament

Legendă	
✓	utilizatorul are drepturi de citire și de scriere
⚠	utilizatorul are doar drepturi de citire / limitate la anumite funcționalități
✗	utilizatorul nu are nici un fel de drepturi

Modulul I.

Prin intermediul **modulului pentru gestiunea resurselor umane**, un inspector poate căuta
un angajat (de orice tip), în funcție de parametrii pe care îi indică: nume, prenume, funcție. Pentru fiecare angajat se va specifica un orar de lucru, acesta putând fi generic (același pentru o anumită zi a săptămânii) sau specific (pentru o anumită dată calendaristică). Orarul este caracterizat prin ziua la care se referă (zi a săptămânii sau dată calendaristică), intervalul orar (momentul de început și momentul de sfârșit), locația (unitatea medicală) în care se desfășoară. De asemenea, pentru fiecare angajat se poate specifica o perioadă de concediu, răstimp în care nu poate furniza servicii medicale.

Modulul II.În cadrul **modulului de operaţii financiar contabile**, un expert poate vizualiza informații cu privire la profitul realizat de lanțul de policlinici, pentru lunile precedente în care s-au înregistrat activități. Profitul este definit ca diferență între venituri și cheltuieli. Veniturile sunt obținute pentru plățile realizate de pacienți în urma furnizării de servicii medicale, iar cheltuielile sunt determinate în funcție de salarii (inclusiv comisioanele medicilor). În cazul salariilor, acestea se consideră ponderate cu numărul de ore realizat în luna respectivă, raportat la numărul de ore specificat în contractul de muncă pentru fiecare angajat în parte. De asemenea, vor fi disponibile rapoarte cu privire la profitul realizat de fiecare medic în parte, pe fiecare locație (unitate medicală) sau pe fiecare specialitate.

Orice angajat poate vizualiza salariile obținute în lunile precedente. În plus, un medic poate consulta profitul pe care l-a generat, calculat ca diferență dintre sumele încasate de către lanțul de policlinici de la pacienți pentru serviciile furnizate și sumele reprezentând salariul și comisioanele sale.

Modulul III.În cadrul **modulului pentru gestiunea activităților operaționale** un recepționer poate realiza o programare pentru un pacient. O programare se face doar pentru o dată calendaristică ulterioară, la un anumit moment de timp și pentru un medic, fiind specificate unul sau mai multe servicii medicale care urmează a fi furnizate (pentru fiecare medic în parte se vor putea selecta doar acele servicii medicale corespunzătoare specialității sau specialităților sale, pentru care deține competențele necesare), durata consultației fiind calculată ca sumă a timpului alocat pentru fiecare procedură în parte. Pentru ziua curentă, un recepționer

poate înregistra un pacient în momentul în care acesta se prezintă în clinică. Totodată, recepționerul emite un bon fiscal ulterior consultației, cuprinzând fiecare serviciu medical care a fost efectuat.

Un **asistent medical** poate completa informații în rapoartele pentru analizele medicale corespunzătoare pacienților care au fost înregistrați pentru acestea. Rezultatul furnizat poate fi o valoare numerică (raportată la un interval de referință), respectiv o valoare binară de tipul pozitiv / negativ.

Un raport pentru analize medicale va fi validat, ulterior nemaiputând fi modificat, fiind însă disponibil pentru consultare în cadrul istoricului pacientului.

Un **medic** poate vizualiza pacienții programați la el pentru ziua calendaristică în curs, listă în care sunt evidențiați cei care au fost înregistrați. De asemenea, pentru un astfel de pacient poate fi consultat întregul istoric, compus din rapoarte medicale anterioare. Pentru fiecare pacient consultat, medicul va completa un raport medical, care va conține, în mod obligatoriu, informații administrative: numele și prenumele pacientului, numele și prenumele medicului care a realizat consultația, numele și prenumele medicului care a recomandat consultația (opțional), numele și prenumele asistentului medical (opțional), data consultației, precum și următoarele secțiuni medicale: istoric, simptome, investigații, diagnostic, recomandări. Secțiunea de investigații va cuprinde subsecțiuni pentru fiecare serviciu medical furnizat, medicul având posibilitatea de a completa rezultatul obținut. Medicul va putea gestiona și lista serviciilor medicale (adăugare, ștergere), în funcție de necesitatea / inoportunitatea

realizării anumitor proceduri. În momentul în care un raport medical este complet, acesta este parafat, astfel că ulterior nu mai este posibilă modificarea sa, acesta putând fi vizualizat în cadrul istoricului pacientului.

3. Model de date

Modelul de date folosit pentru această aplicație este rațional și folosește un server MySQL.

CODUL MySQL pentru crearea tabelelor

```
CREATE SCHEMA IF NOT EXISTS policlinica

USE policlinica ;
CREATE TABLE IF NOT EXISTS utilizator (
  `idutilizator` INT NOT NULL AUTO_INCREMENT,
  `CNP` INT NULL DEFAULT NULL,
  `Nume` VARCHAR(45) NULL DEFAULT NULL,
  `Prenume` VARCHAR(45) NULL DEFAULT NULL,
  `Numar_telefon` INT NULL DEFAULT NULL,
  `Email` VARCHAR(45) NULL DEFAULT NULL,
  `IBAN` VARCHAR(45) NULL DEFAULT NULL,
  `nr_contract` INT NULL DEFAULT NULL,
  `data_angajarii` DATE NULL DEFAULT NULL,
  `functie` VARCHAR(45) NULL DEFAULT NULL,
  `salar_neg` FLOAT(5,2) NULL DEFAULT NULL,
  `nr_ore` INT NULL DEFAULT NULL,
  `nume_utilizator` VARCHAR(45) NULL DEFAULT NULL,
```

```
  `parola` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idutilizator`))

CREATE TABLE IF NOT EXISTS asistent (
  `idasistent` INT NOT NULL,
  `tip` VARCHAR(45) NULL DEFAULT NULL,
  `grad` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idasistent`),
  CONSTRAINT `asistent`
    FOREIGN KEY (`idasistent`)
    REFERENCES `policlinica`.`utilizator` (`idutilizator`))

CREATE TABLE IF NOT EXISTS medic (
  `idmedic` INT NOT NULL,
  `cod_parafa` INT NULL DEFAULT NULL,
  `post_didactic` VARCHAR(45) NULL DEFAULT NULL,
  `titlu_stiintific` VARCHAR(45) NULL DEFAULT NULL,
  `procent` FLOAT(5,2) NULL DEFAULT NULL,
  `grad` VARCHAR(45) NULL DEFAULT NULL,
  `competente` TEXT NULL DEFAULT NULL,
  `specialitatea` VARCHAR(45) NULL DEFAULT NULL,
  `disponibil` TINYINT NULL DEFAULT NULL,
  PRIMARY KEY (`idmedic`),
  CONSTRAINT `medic`
    FOREIGN KEY (`idmedic`)
    REFERENCES `policlinica`.`utilizator` (`idutilizator`))

CREATE TABLE IF NOT EXISTS programare (
```

```
  `idprogramare` INT NOT NULL AUTO_INCREMENT,
  `id_medic` INT NULL DEFAULT NULL,
  `nume_pacient` VARCHAR(45) NULL DEFAULT NULL,
  `prenume_pacient` VARCHAR(45) NULL DEFAULT NULL,
  `ora` TIME NULL DEFAULT NULL,
  `zi` DATE NULL DEFAULT NULL,
  `durata` INT NULL DEFAULT NULL,
  PRIMARY KEY (`idprogramare`),
  INDEX `medic3_idx` (`id_medic` ASC) VISIBLE,
  CONSTRAINT `medic3`
    FOREIGN KEY (`id_medic`)
    REFERENCES `policlinica`.`medic` (`idmedic`))
```

CREATE TABLE IF NOT EXISTS bonuri (

```
  `idbonuri` INT NOT NULL AUTO_INCREMENT,
  `pret` FLOAT(5,2) NULL DEFAULT NULL,
  `idprogramare` INT NULL DEFAULT NULL,
  PRIMARY KEY (`idbonuri`),
  INDEX `idprogram_idx` (`idprogramare` ASC) VISIBLE,
  CONSTRAINT `idprogram`
    FOREIGN KEY (`idprogramare`)
    REFERENCES `policlinica`.`programare` (`idprogramare`))
```

CREATE TABLE IF NOT EXISTS concediu (

```
  `idconcediu` INT NOT NULL AUTO_INCREMENT,
  `iduser` INT NULL DEFAULT NULL,
  `inceput_concediu` DATE NULL DEFAULT NULL,
  `sfarsit_concediu` DATE NULL DEFAULT NULL,
```

```
PRIMARY KEY (`idconcediu`),
INDEX `user2_idx` (`iduser` ASC) VISIBLE,
CONSTRAINT `user2`
  FOREIGN KEY (`iduser`)
  REFERENCES `policlinica`.`utilizator` (`idutilizator`))
```

CREATE TABLE IF NOT EXISTS raport (

```
`idraport` INT NOT NULL AUTO_INCREMENT,
`nume_pacient` VARCHAR(45) NULL DEFAULT NULL,
`prenume_pacient` VARCHAR(45) NULL DEFAULT NULL,
`id_asistent` INT NULL DEFAULT NULL,
`id_medic_consultatie` INT NULL DEFAULT NULL,
`data_consultatiei` DATE NOT NULL,
`simptome` TEXT NULL DEFAULT NULL,
`diagnostic` TEXT NULL DEFAULT NULL,
`recomandari` TEXT NULL DEFAULT NULL,
`idprogramare` INT NULL DEFAULT NULL,
`parafat` TINYINT NULL DEFAULT NULL,
PRIMARY KEY (`idraport`, `data_consultatiei`),
INDEX `medic_consult_idx` (`id_medic_consultatie` ASC) VISIBLE,
INDEX `asistent_idx` (`id_asistent` ASC) VISIBLE,
INDEX `asistent2_idx` (`id_asistent` ASC) VISIBLE,
INDEX `data_consult_idx` (`data_consultatiei` ASC) VISIBLE,
INDEX `program_idx` (`idprogramare` ASC) VISIBLE,
INDEX `program1_idx` (`idprogramare` ASC) VISIBLE,
CONSTRAINT `asistent2`
  FOREIGN KEY (`id_asistent`)
```

```
  REFERENCES `policlinica`.`asistent` (`idasistent`),
 CONSTRAINT `medic_consult`
  FOREIGN KEY (`id_medic_consultatie`)
  REFERENCES `policlinica`.`medic` (`idmedic`),
 CONSTRAINT `program1`
  FOREIGN KEY (`idprogramare`)
  REFERENCES `policlinica`.`programare` (`idprogramare`))
```

CREATE TABLE IF NOT EXISTS investigatii (

```
 `idinvestigatii` INT NOT NULL,
 `id_raport` INT NULL DEFAULT NULL,
 `ziua` DATE NULL DEFAULT NULL,
 `serviciu_medical` VARCHAR(45) NULL DEFAULT NULL,
 `descriere` TEXT NULL DEFAULT NULL,
 PRIMARY KEY (`idinvestigatii`),
 INDEX `raportid_idx` (`id_raport` ASC) VISIBLE,
 CONSTRAINT `raportid`
  FOREIGN KEY (`id_raport`)
  REFERENCES `policlinica`.`raport` (`idraport`))
```

CREATE TABLE IF NOT EXISTS istoric_medical (

```
 `data` DATE NOT NULL,
 `nume_pacient` VARCHAR(45) NULL DEFAULT NULL,
 `prenume_pacient` VARCHAR(45) NULL DEFAULT NULL,
 PRIMARY KEY (`data`),
 CONSTRAINT `data_raport`
  FOREIGN KEY (`data`)
  REFERENCES `policlinica`.`raport` (`data_consultatiei`))
```

```
CREATE TABLE IF NOT EXISTS  program`(
  `nrProgram` INT NOT NULL AUTO_INCREMENT,
  `luni_s` TIME NULL DEFAULT NULL,
  `luni_f` TIME NULL DEFAULT NULL,
  `marti_s` TIME NULL DEFAULT NULL,
  `marti_f` TIME NULL DEFAULT NULL,
  `miercuri_s` TIME NULL DEFAULT NULL,
  `miercuri_f` TIME NULL DEFAULT NULL,
  `joi_s` TIME NULL DEFAULT NULL,
  `joi_f` TIME NULL DEFAULT NULL,
  `vineri_s` TIME NULL DEFAULT NULL,
  `vineri_f` TIME NULL DEFAULT NULL,
  `sambata_s` TIME NULL DEFAULT NULL,
  `sambata_f` TIME NULL DEFAULT NULL,
  `duminica_s` TIME NULL DEFAULT NULL,
  `duminica_f` TIME NULL DEFAULT NULL,
  PRIMARY KEY (`nrProgram`))

CREATE TABLE IF NOT EXISTS unitate_medicala(
  `idunitate_medicala` INT NOT NULL AUTO_INCREMENT,
  `denumire` VARCHAR(45) NULL DEFAULT NULL,
  `adresa` VARCHAR(45) NULL DEFAULT NULL,
  `descriere_servicii` TEXT NULL DEFAULT NULL,
  `nrProgram` INT NOT NULL,
  `idAngajat` INT NULL DEFAULT NULL,
  PRIMARY KEY (`idunitate_medicala`),
  INDEX `program_idx` (`nrProgram` ASC) VISIBLE,
  INDEX `angajatId_idx` (`idAngajat` ASC) VISIBLE,
```

```
  CONSTRAINT `angajatId`
    FOREIGN KEY (`idAngajat`)
    REFERENCES `policlinica`.`utilizator` (`idutilizator`),
  CONSTRAINT `program`
    FOREIGN KEY (`nrProgram`)
    REFERENCES `policlinica`.`program` (`nrProgram`))
```

CREATE TABLE IF NOT EXISTS orar_generic (

```
  `idorar_generic` INT NOT NULL AUTO_INCREMENT,
  `idUnitate` INT NULL DEFAULT NULL,
  `ziua` DATE NULL DEFAULT NULL,
  `ora_s` TIME NULL DEFAULT NULL,
  `ora_f` TIME NULL DEFAULT NULL,
  `idAngajat` INT NULL DEFAULT NULL,
  PRIMARY KEY (`idorar_generic`),
  INDEX `orar_unitate_idx` (`idUnitate` ASC) VISIBLE,
  INDEX `orar_angajat_idx` (`idAngajat` ASC) VISIBLE,
  CONSTRAINT `orar_angajat`
    FOREIGN KEY (`idAngajat`)
    REFERENCES `policlinica`.`utilizator` (`idutilizator`),
  CONSTRAINT `orar_unitate`
    FOREIGN KEY (`idUnitate`)
    REFERENCES `policlinica`.`unitate_medicala`
(`idunitate_medicala`))
```

CREATE TABLE IF NOT EXISTS orar_specific (

```
  `idorar_specific` INT NOT NULL AUTO_INCREMENT,
  `iduser` INT NULL DEFAULT NULL,
```

```
  `zi` DATE NULL DEFAULT NULL,
  `ora_s` TIME NULL DEFAULT NULL,
  `ora_f` TIME NULL DEFAULT NULL,
  `unitate_med` VARCHAR(45) NULL DEFAULT NULL,
  PRIMARY KEY (`idorar_specific`),
  INDEX `user_idx` (`iduser` ASC) VISIBLE,
  CONSTRAINT `user`
    FOREIGN KEY (`iduser`)
    REFERENCES `policlinica`.`utilizator` (`idutilizator`))
```

CREATE TABLE IF NOT EXISTS salariu (

```
  `idsalariu` INT NOT NULL AUTO_INCREMENT,
  `id_utilizator` INT NULL DEFAULT NULL,
  `salariu` INT NULL DEFAULT NULL,
  `data` DATE NULL DEFAULT NULL,
  PRIMARY KEY (`idsalariu`),
  INDEX `idutil_idx` (`id_utilizator` ASC) VISIBLE,
  CONSTRAINT `idutil`
    FOREIGN KEY (`id_utilizator`)
    REFERENCES `policlinica`.`utilizator` (`idutilizator`))
```

CREATE TABLE IF NOT EXISTS servicii_medicale (

```
  `id_servicii` INT NOT NULL AUTO_INCREMENT,
  `id_medic` INT NULL DEFAULT NULL,
  `denumire` VARCHAR(45) NULL DEFAULT NULL,
  `pret` INT NULL DEFAULT NULL,
  `durata` TIME NULL DEFAULT NULL,
  `competenta` TEXT NULL DEFAULT NULL,
```

```
PRIMARY KEY (`id_servicii`),
INDEX `medic2_idx` (`id_medic` ASC) VISIBLE,
CONSTRAINT `medic2`
  FOREIGN KEY (`id_medic`)
  REFERENCES `policlinica`.`medic` (`idmedic`))
```

Proceduri utilizate:

Procedura care adaugă concediu pentru un utilizator folosind id-ul acestuia și perioada concediului.

```
CREATE PROCEDURE adaugareConcediu( id INT, data_inc date, data_f date )

    BEGIN
        SELECT id
        FROM utilizator a
        WHERE
            a.idutilizator = id INTO @id;
        IF
            ( @id IS NOT NULL ) THEN
            INSERT                                                INTO
concediu(iduser,inceput_concediu,sfarsit_concediu)
            VALUES
                ( @id, data_inc, data_f );

        END IF;
      END
```

Procedura care afișează toate datele despre un utilizator folosind 3 parametri – nume, prenume și funcție

```
CREATE PROCEDURE cautaUtilizator(numeU VARCHAR ( 45 ),prenumeU VARCHAR
( 45 ),functieU VARCHAR ( 45 ))

BEGIN

    SELECT * FROM utilizator  WHERE numeU = nume AND prenumeU = prenume
AND functieU = functie ;

END
```

Procedura care se apelează atunci cand se înregistrează un utilizator folosind pagina de creare cont din interfață.

CREATE PROCEDURE creare_cont(CNP int,Nume VARCHAR(45),Prenume VARCHAR(45),Numar_telefon int,Email VARCHAR(45),IBAN VARCHAR(45),nr_contract int,data_angajarii date,functie VARCHAR(45),nume_utilizator VARCHAR(45),parola VARCHAR(45))

BEGIN

INSERT INTO utilizator(CNP,Nume,Prenume,Numar_telefon,Email,IBAN,nr_contract,data_angajarii,functie,nume_utilizator,parola)

VALUES(CNP,Nume,Prenume,Numar_telefon,Email,IBAN,nr_contract,data_angajarii,functie,nume_utilizator,parola);

END

Procedura care afișează informații despre un medic (ex competențe, titlu științific etc.) folosindu-se de email-ul acestuia

CREATE PROCEDURE date_medic(email1 varchar (100))

begin

select medic.idmedic, utilizator.Nume, utilizator.Prenume, medic.specialitatea, medic.grad, medic.competente, medic.titlu_stiintific,medic.post_didactic

from medic

join utilizator on medic.idmedic=utilizator.idutilizator

where utilizator.Email=email1;

end

Procedura care afișează informații despre concediul unei persoane folosind id-ul acesteia

```
CREATE PROCEDURE date_despre_concediu(id int)
BEGIN
        SELECT u.Nume as 'Nume', u.Prenume as 'Prenume', c.inceput_concediu as 'dataInceput', c.sfarsit_concediu
as 'dataSfarsit' from concediu c, utilizator u where c.iduser = id and u.idutilizator = id;
end;
```

Triggere

Acest trigger se activează după ce medicul și-a pus parafa pe un raport , asta însemnând că a terminat consultația și el devine disponibil. În tabela medic există o variabilă care poate lua doar valorile 0 sau 1 (0 – medicul este în timpul unei programări sau nu lucrează , 1 – medicul este disponibil)

```
DELIMITER //
CREATE TRIGGER medic_disponibil AFTER UPDATE ON raport
    FOR EACH ROW
        BEGIN
    IF parafat = 1 AND id_medic_consultatie = medic.idmedic THEN
  UPDATE medic ON disponibil
    SET disponibil = 1;
        END IF;
    END;
DELIMITER ;
```

VIEW-uri

Având în vedere nevoia de a se putea vizualiza programul de lucru al unui angajat, am creat un view care sa afișeze numele, prenumele, data, ora de început a zilei de lucru și ora de sfârșit.

CREATE VIEW Program_Angajat AS

SELECT U.Nume as 'Nume', U.Prenume as 'Prenume', O.ziua as 'Data', O.ora_s as 'Ora_Inceput', O.ora_f as 'Ora_Sfarsit'

FROM orar_generic O, utilizator U

where U.idutilizator = O.idAngajat;

Apoi am mai creat un view pentru a afișa datele despre progaramarea unui pacient, și anume numele și prenumele pacientului, data și ora programării și durata. De asemena se va afișa și numele si prenumele medicului la care s-a făcut programarea.

CREATE VIEW Orar_Programari AS

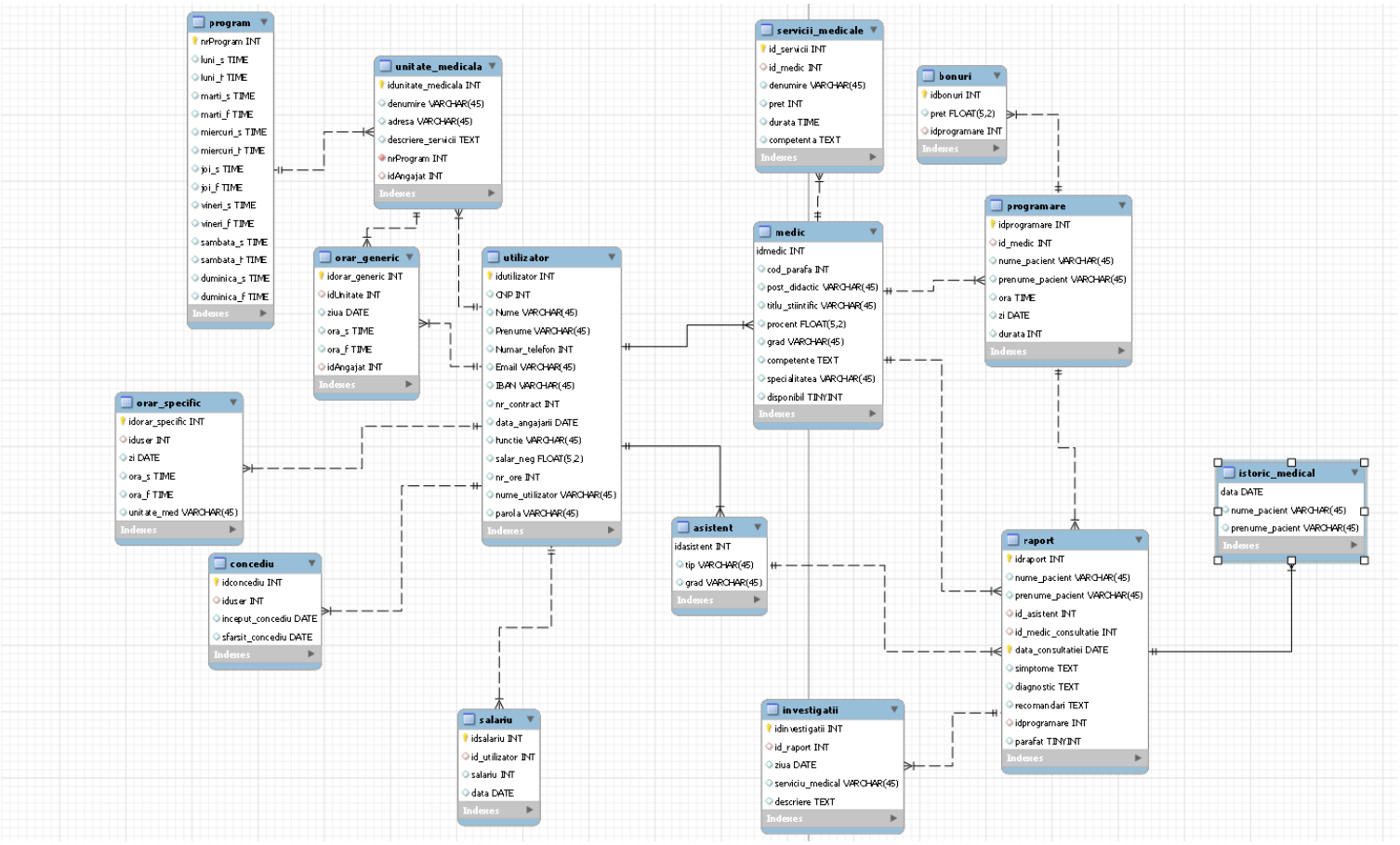
SELECT CONCAT(P.nume_pacient, ' ', P.prenume_pacient) AS 'Pacient', P.zi AS 'Data', P.ora AS 'Ora', P.durata AS 'Durata',

CONCAT(U.Nume, ' ', U.Prenume) AS 'medic'

FROM programare P, medic M, utilizator U

WHERE M.idmedic= U.idutilizator AND P.id_medic = M.idmedic;

Diagrama UML



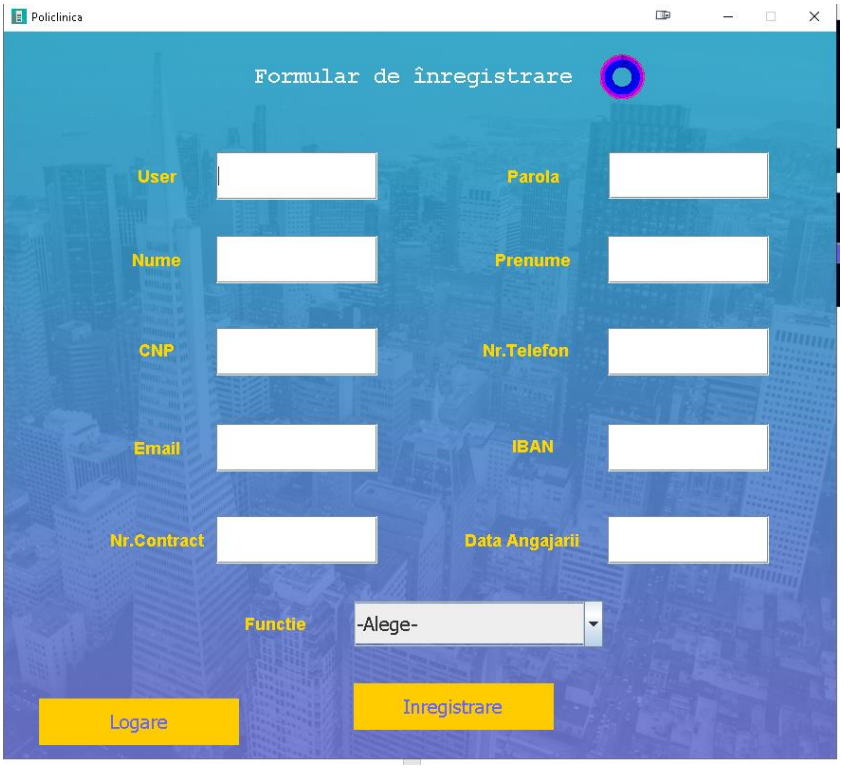
4. Mod de implementare

1.Pagina principală

Aceasta este prima pagina care apare atunci când deschidem aplicația.

Aici avem 2 opțiuni : fie ne logăm cu nume de utilizator și parola dacă avem cont , fie apăsăm pe Creare Cont și mergem la formularul de înregistrare.

2.Formularul de înregistrare

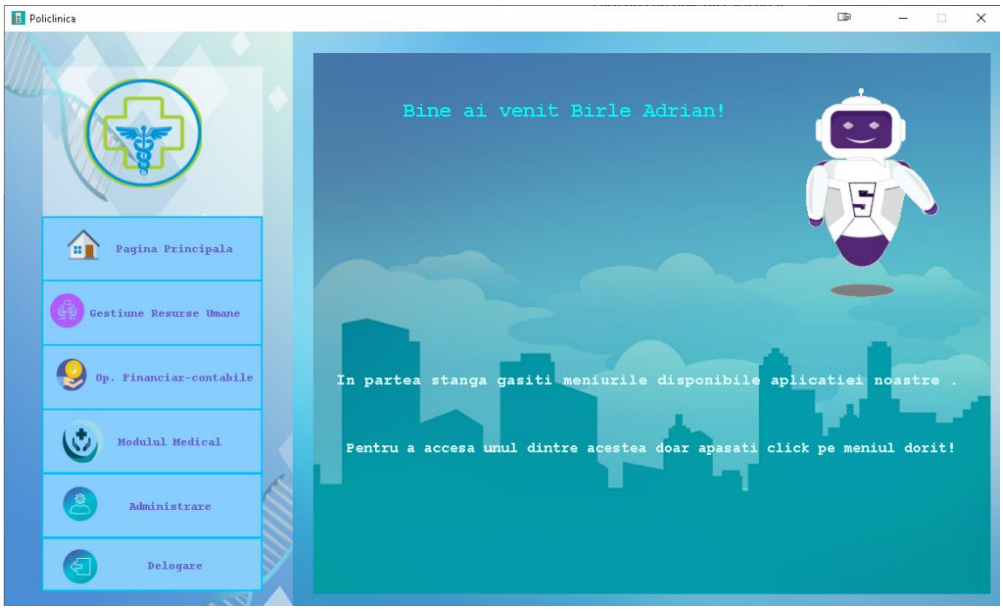


În această pagină se poate face o înregistrare a unei persoane.

Este de precizat că de aici se poate înregistra și un administrator sau Super Administrator , dar cu o condiție. Dacă se încearcă înregistrarea ca Super Administrator programul va cere un cod unic pe care doar noi (directorii /managerii) îl avem , iar la înregistrarea ca Administrator se va cere un alt

cod unic pe care îl vom avea noi + SuperAdministratorii.

3.Pagina dupa Logarea cu succes



Această fereastră va apărea după ce logarea utilizatorului a avut success.

În partea de sus avem un roboțel care face cu mâna și spune “Hi” și lângă el avem un mesaj de bun venit care preia din baza de date numele și prenumele utilizatorului și le afișează.

În partea stângă avem meniurile disponibile. Meniul administrare se va deschide doar dacă utilizatorul este de tip Administrator sau Super Administrator , altfel va aparea o eroare de genul “Accesul este interzis”. Tot în meniu mai avem și un buton de delogare.

Meniurile “Gestiune resurse umane” , “Op.Financiar-contabile”, “Modulul medical” reprezintă cele 3 module prezentate în specificația proiectului.

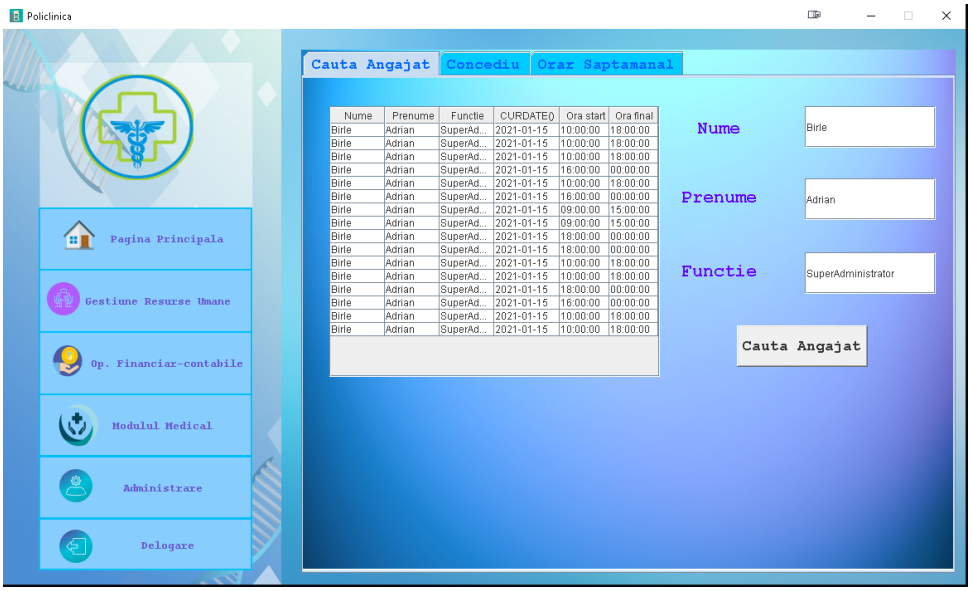
Folosindu-ne de clasele din JAVA am făcut în așa fel încât , la logare , programul reține tipul de utilizator (ex Medic, Asistent etc.) iar la apăsarea pe unul din Meniurile care reprezintă cele 3 module, se va deschide o fereastră unică în care le apar opțiunile permise lor. Spre exemplu dacă eu sunt Medic și apăs pe meniul “ Gestiunea resurselor umane” îmi va apărea o fereastră doar cu operațiile permise funcției de medic spre exemplul vizualizarea orarului, dar nu voi avea drepturi de acces. Dacă sunt Inspector și apăs pe meniul “ Gestiunea resurselor umane” mi se va deschide o fereastră cu operațiile permise Inspectorului, de

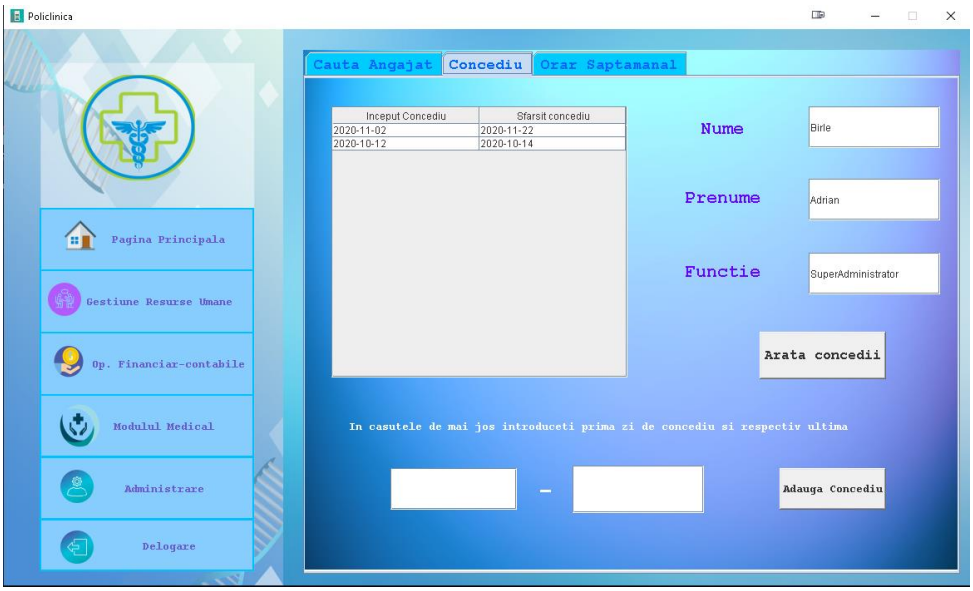
exemplu adăugarea concediului și voi avea drepturi atât de citire , cât și de scriere în acest modul.

Procedeul este analog pentru celelalte 2 meniuri/module , cu observația că în Modulul III (cel Medical) mi se va deschide o fereastră doar dacă aparțin acelui departament , altfel accesul este interzis.

MODULUL I (Gestionarea resurselor Umane)

Dacă ne logăm ca inspector sau ca expert contabil vom avea acces să vedem orarul pe ziua curentă a unui angajat căutându-l după nume, prenume și funcție.



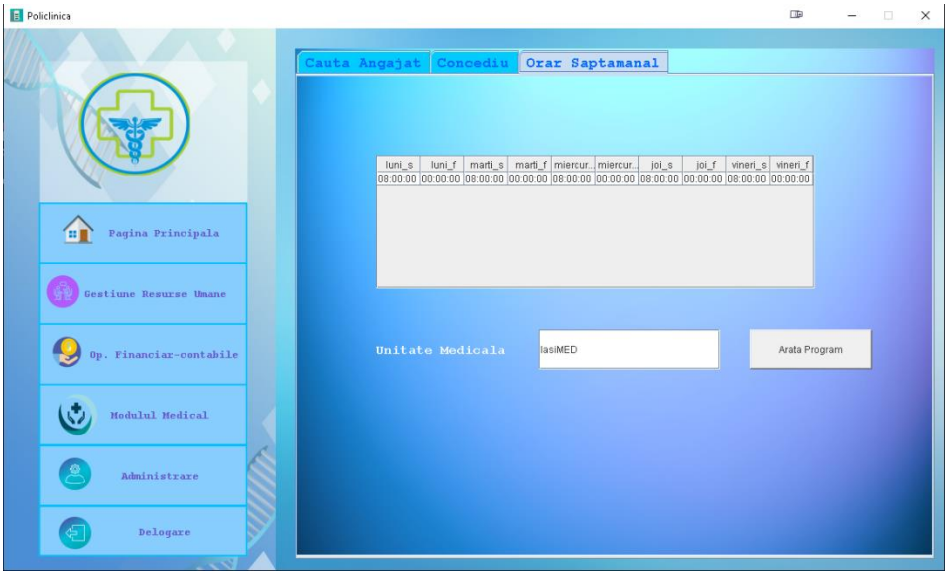


Când deschidem această fereastră ne apar toți utilizatorii din baza de date.

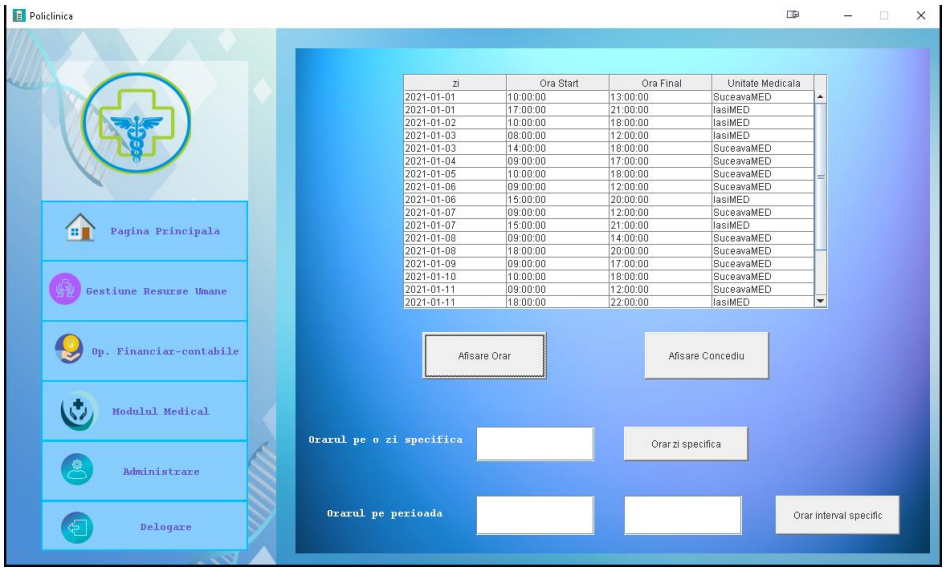
Aici putem vedea concediile pentru anumiți utilizatori sau să adăugăm concedii dacă suntem inspectori.

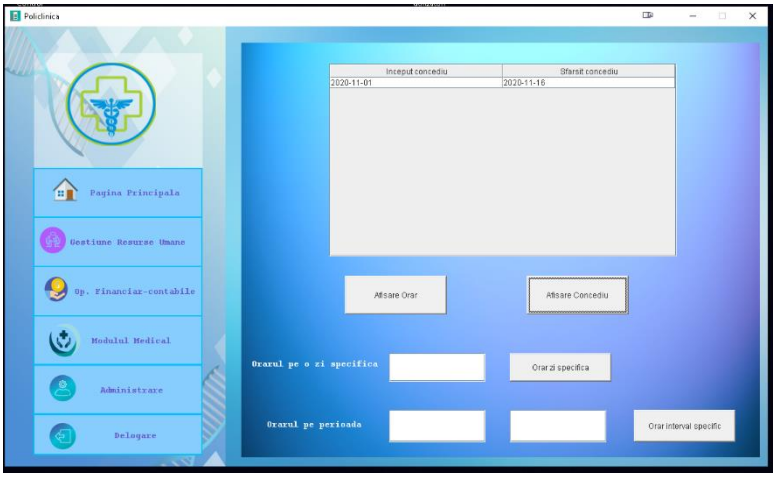
Aici pot vedea orarul

pentru o anumită Unitate Medicală atât Inspectorii cât și Experții economici.

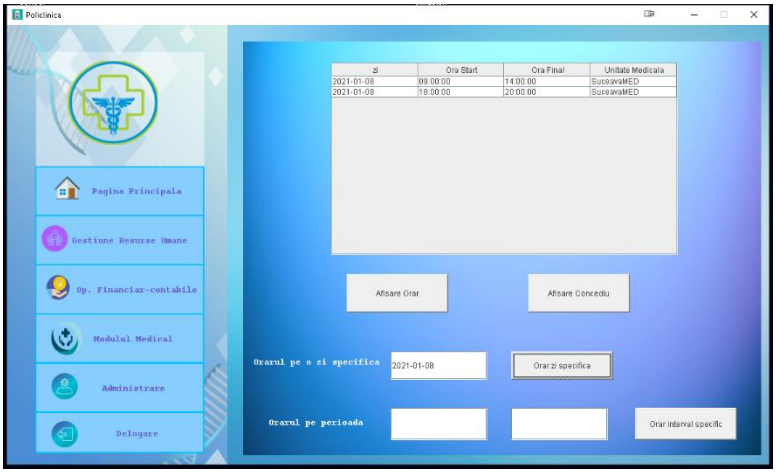


În imaginea din dreapta suntem logați ca Medic și avem opțiunile de Afișare Orar și Unitatea Medicală, afișare concediu , orar pe o anumită zi și orar pe o perioadă specifică.

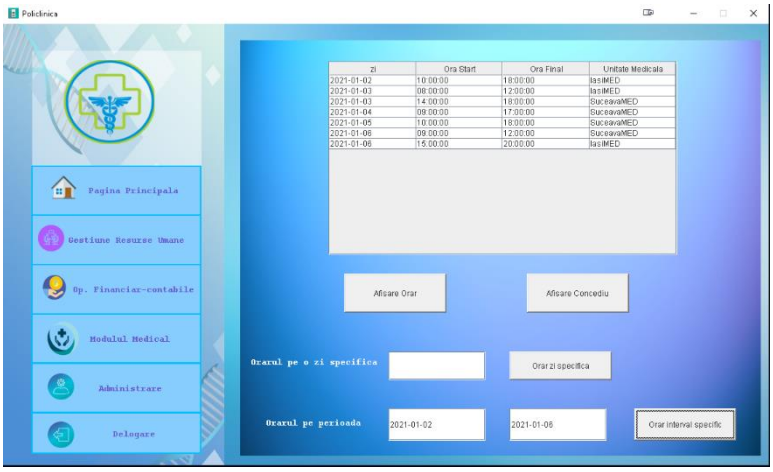




Afișarea concediului



Afișarea Orarului pe o zi specifică



Afișarea Orarului pe o anumită perioadă specifică

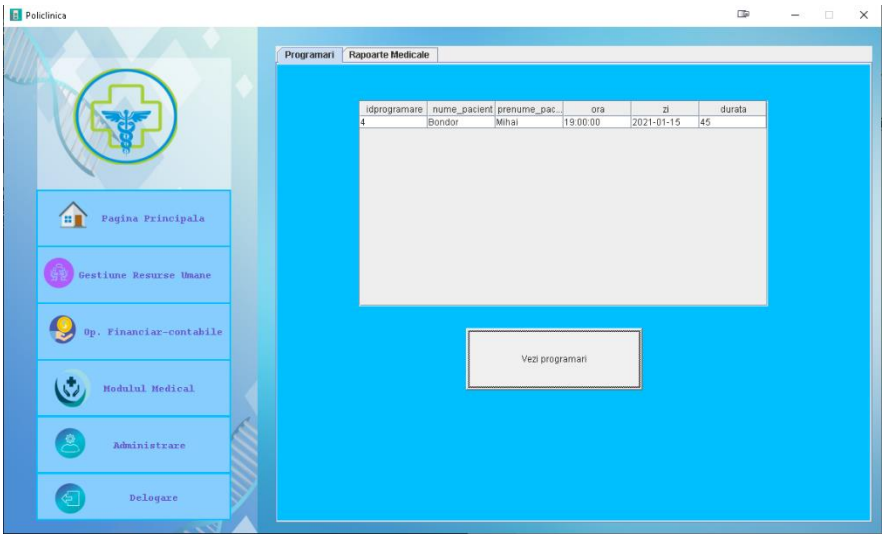
Modulul III (Modulul Medical)

În acest modul vom avea 3 cazuri adică 3 ferestre diferite , câte una pentru fiecare funcție din departamentul medical , adică câte una pentru Medic, una pentru Asistent Medical și una pentru Recepționer.

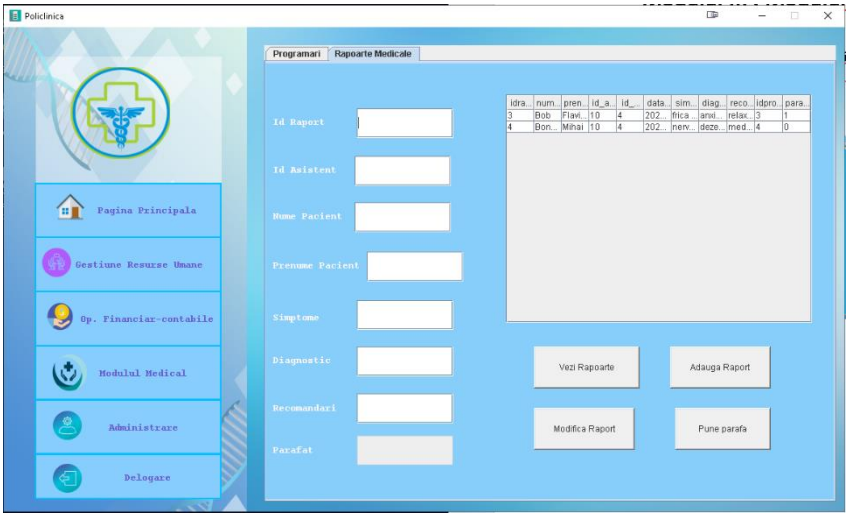
Logat ca medic

Vom avea 2 opțiuni :

- Să vedem programările din ziua curentă așa cum se vede în imaginea din dreapta
- Să vedem rapoartele medicale din ziua curentă

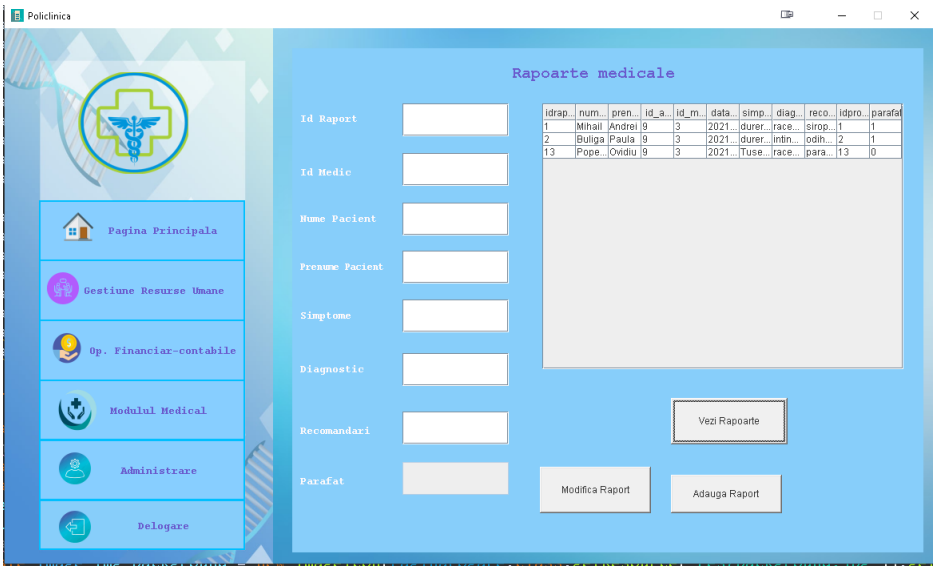


Aici putem vedea rapoartele din ziua curentă, să adăugăm un raport, să îl modificăm și dacă ne punem parafa pe el atunci el nu se mai poate modifica.



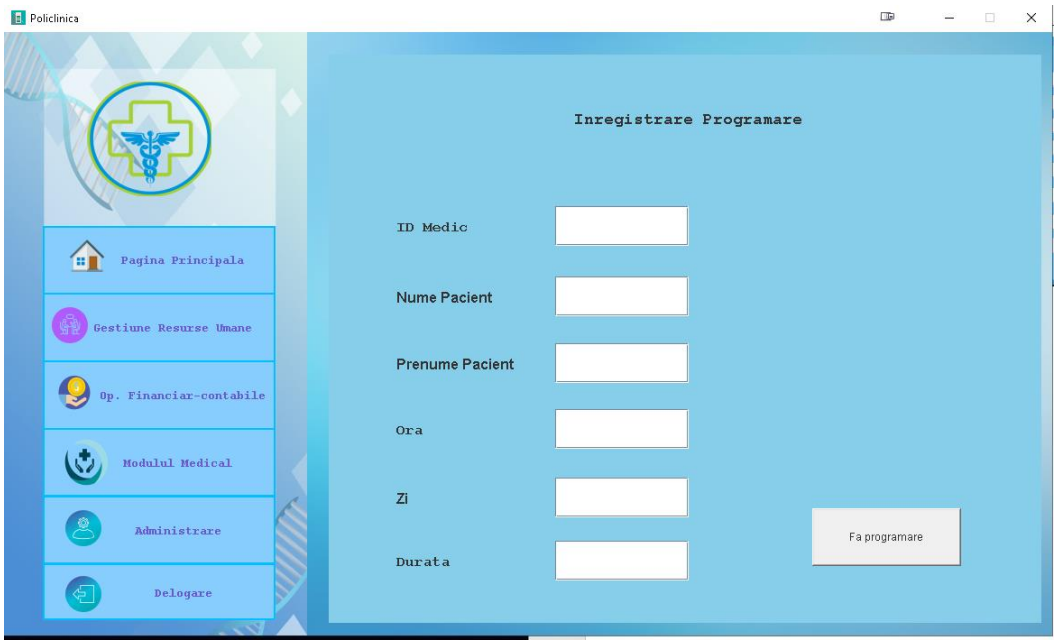
Logat ca asistent

Dacă suntem logați ca asistent medical putem vedea rapoartele din ziua curentă pentru pacienții la care am participat ca asistenți medicali și putem adăuga un raport nou sau sa modificam un raport daca el nu este parafat.



Logat ca receptioner

Dacă suntem logați ca receptioneri vom putea înregistra o programare pentru un anumit medic.



5.Securitatea și justificarea soluției alese

Am ales această forma de implementare deoarece este user-friendly și este ușor de utilizat de către oricine , chiar și de cei care nu prea se pricep la utilizarea calculatoarelor.

Pentru deschiderea meniurilor dorite (modulele) am folosit mai multe clase care functioneaza ca niște VIEW-uri pentru a se deschide fiecarui utilizator opțiunile disponibile conform funcției pe care o deține.

În ceea ce privește partea de gestiunea resurselor umane am implementat o modalitate de a căuta un utilizator și de a afișa date despre acesta inclusiv informații referitoare la programul de lucru la orarul angajatului și la concediu

O posibila dezvoltare ulterioara ar fi pentru modulele resurse și medical. Pentru modulul resurse umane ar fi potrivită predarea atribuțiilor unui angajat în cazul în care el este în concediu altui angajat.

Pentru modulul medical ar fi potrivită adăugarea unor Medici de gardă care să fie și pe perioada nopții în caz de urgențe.

Pe parte de Securitate , am folosit un formular de logare care necesită un nume de utilizator și o parolă (de minim 8 caractere), iar la introducerea greșită a datelor se va afișa un mesaj de eroare.

Totodată și în formularul de înregistrare în cazul în care se încearcă crearea unui cont de tip Super Administrator sau Administrator se va cere câte un cod unic din 8 caractere, iar în cazul introducerii greșite a codului se va afișa un mesaj de eroare.