



PROIECT

Lant de policlinici

Disciplina “Introducere in Baze de Date”

Realizat de:

Pele Carmen-ioana

Stan Alexandra-Paula

Vîrnă Ioana

Facultatea de Automatică și Calculatoare

Anul II

Grupa 30222

Profesor coordinator: Ing. Cosmina Ivan

Cuprins

1. Introducere
2. Specificație proiect
3. Model de date
 - Tabele si attribute
 - Diagrama UML/EER
 - cod SQL
4. Aplicatia lantului de policlinici
 - mod de implementare
 - utilizare
5. Securitate si justificarea solutiei alese

1.Introducere

O bază de date este o colecție organizată de informații sau de date structurate, stocate electronic într-un computer. O bază de date este controlată, de regulă, de un [sistem de management al bazelor de date \(DBSM\)](#). Cumulat, datele, DBMS și aplicațiile asociate reprezintă un sistem de baze de date, denumit prescurtat bază de date.

Datele din cele mai obișnuite tipuri de baze de date sunt distribuite de regulă pe linii și coloane, în diferite tabele, pentru eficientizarea procesării și interogării datelor. Datele pot fi accesate, gestionate, modificate, actualizate, controlate și organizate cu ușurință. Majoritatea bazelor de date utilizează un limbaj structurat de interogare (SQL) pentru scrierea și interogarea datelor.

Proiectul nostru ce implementează un sistem informatic destinate gestiunii activităților dintr-un lanț de policlinici vine în ajutorul celor ce lucrează la acea policlinică, prin gestionarea mult mai ușoară a tuturor informațiilor.

Având în vedere sistemul actual, trăind într-o perioadă pandemică, acest sistem este esențial pentru buna desfășurare a tuturor activităților de la policlinică, orice personal neesențial putând lucra de acasă (ex: receptioner, administrator, etc.). Vechiul mod de gestionare a datelor se dovedește a fi ineficient, putându-se greși ușor în preluarea și rescrierea unor informații. Este instabil atât din punct de vedere al securității, cât și din punctul de vedere al gestiunii timpului.

Cu toate acestea, proiectul acesta, dezvoltă o interfață complexă implementată prin intermediul limbajului JAVA ce permite accesarea diferitelor ferestre oferite de aplicație. Aceasta comunică în permanență cu baza de date SQL, în care sunt reținute toate datele.

În ceea ce urmează, vor fi prezentate modelul de date cu implementarea în SQL, aplicația lanțului de policlinici care prezintă atât implementarea cât și utilizarea, cât și securitatea și justificarea soluției alese.

2. Specificatie

Se dorește implementarea unui sistem informatic destinat gestiunii activităților 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.

Aplicația va trebui să utilizeze un sistem de gestiune pentru baze de date MySQL, iar interacțiunea cu acesta va fi realizată doar prin interfața grafică. Funcționalitățile pe care le va oferi programul vizează operații ce țin de gestiunea angajaților, serviciul financiar-contabil și administrarea operațiilor curente din cadrul policlinicii (gestiunea pacienților programați, completarea unui raport medical, emiterea unui bon fiscal)¹.

Aplicația 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 reține informații precum CNP, nume, prenume, adresa, număr de telefon, email, cont IBAN, numărul de contract, data angajării, funcția deținută 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 solicit 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 și șterge informații în baza de date legate de utilizatori. De asemenea, va exista și 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, recepționar, 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 didactic⁶.

Totodată, fiecare medic are negociat un procent din serviciile medicale realizate care îi revine, adițional 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 angajați în funcție de drepturile pe care le dețin. Astfel, vor fi implementate un **modul pentru gestiunea resurselor umane** ce vizează gestiunea programului de lucru și al concediilor angajaților, un **modul pentru operații financiar-contabile** care determină profitul operațional ca diferență între venituri (sume încasate pentru serviciile medicale) și cheltuieli (plăți efectuate către angajați 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 receptioneri, 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 angajaților din diferite departamente la modulele sistemului sunt descrise în tabelul de mai jos:

modul	departament / tip angajat	resurse umane	economic	medical		
				receptioner	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 submodulele raport medical analize	doar submodulele istoric și rapori medical

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 parametri 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)⁷. 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.

Angajații de tip inspector resurse umane și expert financiar contabil au la dispoziție o secțiune în care pot consulta orarul săptămânal și informații despre concediile efectuate.

Angajații de tip receptioner, asistent medical și medic pot vizualiza informațiile furnizate de modulul pentru gestiunea resurselor umane doar în ceea ce privește propria persoană (orarul pentru luna în curs, pentru fiecare zi indicându-se intervalul orar și locația, tratând și situațiile în care nu există nici un program de lucru specificat sau angajatul se află în concediu). Ei nu vor avea posibilitatea de a modifica în nici un fel aceste informații.

Modulul II.

În cadrul modulului de operații 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 specialitate⁸.

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 recepționar 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 recepționar poate înregistra un pacient în momentul în care acesta se prezintă în clinică. Totodată, recepționarul 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 validă t, 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

Cod creare tabele:

```
drop database if exists Policlinica;  
CREATE DATABASE IF NOT EXISTS Policlinica;  
use Policlinica;
```

```
CREATE TABLE IF NOT EXISTS Angajat (  
Id_angajat INT NOT NULL,  
Functie VARCHAR(45) NOT NULL,  
Salar_neg INT NOT NULL,  
Nr_ore INT NOT NULL,  
PRIMARY KEY (Id_angajat));
```

```
CREATE TABLE IF NOT EXISTS Asistent_medical (  
Id_asis_medical INT NOT NULL,  
Tip VARCHAR(20),  
Grad VARCHAR(20),  
Id_ca_utilizator INT NOT NULL,  
id_de_la_medie int,  
PRIMARY KEY (Id_asis_medical));
```

```
CREATE TABLE IF NOT EXISTS Medic (  
Id_medie INT NOT NULL,  
Specialitate VARCHAR(20) NOT NULL,  
Grad VARCHAR(20) NOT NULL,  
Cod_Parafa INT UNIQUE NOT NULL,  
Competente_detinute VARCHAR(45) NOT NULL,  
Titlu_Stiintific VARCHAR(20), -- poate sa nu aiba  
Postul_Didactic VARCHAR(20), -- poate sa nu aiba  
Procent_neg_servicii FLOAT(4,1) NOT NULL,  
PRIMARY KEY(Id_medie),  
Id_ca_utilizator INT NOT NULL);
```

```
CREATE TABLE IF NOT EXISTS Servicii (  
Id_servicii INT NOT NULL,  
Secialitate varchar(20),  
Necesitate_competente VARCHAR(45),  
Pret INT,  
Durata INT,  
echipament_necesar varchar(255),  
PRIMARY KEY (Id_servicii));
```

```
drop table if exists Program;  
CREATE TABLE IF NOT EXISTS Program(  

```

```
id_program int primary key,  
datapr date,  
ora_deschidere int,  
ora_inchidere int,  
unitate int);
```

```
drop table if exists Unitati_Medicale;  
CREATE TABLE IF NOT EXISTS Unitati_Medicale  
(id_unitate_medicala int primary key,  
denumire varchar(255),  
adresa varchar(255));
```

```
drop table if exists cabinete;  
CREATE TABLE IF NOT EXISTS cabinete(  
id_cabinet int primary key,  
id_unitate int,  
id_medic int,  
echipamente varchar(255));
```

```
drop table if exists Utilizatori;  
CREATE TABLE IF NOT EXISTS Utilizatori(  
id_utilizator int unique primary key, tip varchar(255),  
parola varchar(50),  
CNP int(13), nume varchar(255), prenume varchar(255),  
adresa varchar(255), nr_telefon varchar(255),  
mail varchar(255), cont varchar(255),  
nr_contract int, data_angajarii date,  
functie varchar(255), rol varchar(255) ,  
vaccin varchar(10),  
id_ca_angajat int);
```

```
drop table if exists Programari;  
CREATE TABLE IF NOT EXISTS Programari(  
id_programare int primary key,  
id_pacient int,  
id_serviciupr int,  
id_serviciupr2 int,  
id_serviciupr3 int,  
ora_inceperii int,  
ora_finalizarii int,  
datapacient date, #yyyy-mm-dd  
unitatea int,  
cabinetul int,  
medicul int);
```

```
drop table if exists ProgramAngajati;  
CREATE TABLE IF NOT EXISTS ProgramAngajati(  

```



```
id_prang int primary key,  
id_ca_angajat int,  
datapr date,  
ora_incepere int,  
ora_plecare int,  
unitate int,  
concediu bool);
```

```
drop table if exists Pacienti;  
CREATE TABLE IF NOT EXISTS Pacienti(  
id_pacient int primary key,  
alergii varchar(255),  
boli varchar(255),  
nume varchar(30),  
prenume varchar(50),  
datanasterii date,  
sexul char,  
domiciliu varchar(255),  
cetatenie varchar(30),  
asigurare varchar(10));
```

```
drop table if exists IstoricBonuri;  
CREATE TABLE IF NOT EXISTS IstoricBonuri(  
id_bon int primary key,  
id_de_la_programare int,  
sumaplatita varchar(255));
```

```
drop table if exists raport_medical;  
CREATE TABLE IF NOT EXISTS raport_medical(  
id_fisa int primary key,  
id_programare int,  
istoric varchar(255),  
simptome varchar(255),  
investigatii varchar(255),  
diagnostic varchar(255),  
recomandari varchar(255),  
validare int);
```

```
drop table if exists Istoric_Salariu;  
CREATE TABLE IF NOT EXISTS Istoric_Salariu(  
id_salariu int primary key,  
id_angajat int,  
data_aplicarii date,  
salariu int);  
Alter table Istoric_Salariu  
add FOREIGN KEY (id_angajat) REFERENCES angajat(id_angajat);
```

Alter table Asistent_medical

```
add FOREIGN KEY (Id_ca_utilizator) REFERENCES utilizatori(id_utilizator),  
add FOREIGN KEY (Id_de_la_medic) REFERENCES Medic(id_medic);
```

Alter table Medic

```
add FOREIGN KEY (Id_ca_utilizator) REFERENCES utilizatori(id_utilizator);  
alter table raport_medical  
-- add FOREIGN KEY (id_pacient) REFERENCES pacienti(id_pacient),  
add FOREIGN KEY (id_programare) REFERENCES Programari(id_programare);
```

Alter table Utilizatori

```
add FOREIGN KEY (id_ca_angajat) REFERENCES Angajat(id_angajat);
```

Alter table Program

```
add FOREIGN KEY (unitate) references Unitati_Medicale(id_unitate_medicala);
```

Alter table Programari

```
add FOREIGN KEY (unitatea) references Unitati_Medicale(id_unitate_medicala),  
add foreign key (id_serviciupr) references Servicii(id_servicii),  
-- add foreign key (id_serviciupr2) references Servicii(id_servicii),  
-- add foreign key (id_serviciupr3) references Servicii(id_servicii),  
add foreign key (medicul) references Medic(Id_medic),  
add foreign key (cabinetul) references cabinete(id_cabinet),  
add FOREIGN KEY (id_pacient) REFERENCES pacienti(id_pacient);
```

Alter table ProgramAngajati

```
add FOREIGN KEY (unitate) references Unitati_Medicale(id_unitate_medicala),  
add FOREIGN KEY (id_ca_angajat) references Angajat(Id_angajat);
```

alter table cabinete

```
add foreign key(id_unitate) references Unitati_Medicale(id_unitate_medicala),  
add foreign key(id_medic) references Medic(Id_medic);
```

Alter table Istoriconuri

```
add FOREIGN KEY (id_de_la_programare) REFERENCES Programari(id_programare);
```

View-uri folosite:

Pentru a afisa orarul programarilor am folosit view-ul:

```
DROP VIEW IF EXISTS Orar_programari;
```

```
CREATE VIEW Orar_Programari AS  
SELECT CONCAT(P.num, ' ', P.preume) AS 'Pacient', ora_inceperii, ora_finalizarii,  
CONCAT(U.num, ' ', U.preume) AS 'medic'  
FROM programari Pr, medic M, pacienti P, utilizatori U  
WHERE id_med = medicul AND Pr.id_pacient = P.id_pacient  
AND Id_ca_utilizator = id_utilizator;
```

Pentru a afisa programul angajatilor am folosit view-ul:

```
DROP VIEW IF EXISTS Program_Angajat;
```

```
CREATE VIEW Program_Angajat AS  
SELECT datapr AS 'Data', CONCAT(Pr.ora_incepere, ' - ', Pr.ora_plecure) AS 'Program',  
U.denumire AS 'Unitate medicala', U.adresa, utilizatori.num, id_angajat  
FROM programangajati Pr, angajat A, unitati_medicale U, utilizatori  
WHERE id_unitate_medicala = unitate  
AND id_angajat = Pr.id_ca_angajat AND id_angajat = utilizatori.id_ca_angajat;
```

Proceduri folosite:

Pentru a verifica daca un angajat e in concediu:

```
DELIMITER //
DROP PROCEDURE IF EXISTS verificare;
CREATE PROCEDURE verificare(n varchar(255), p varchar(255), t varchar(255), datav date)
BEGIN
SELECT concediu FROM programangajati Pr, utilizatori U, angajat A
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND U.num = n AND U.prenume = p AND tip = t AND datapr = datav;
END
// DELIMITER
```

Pentru a verifica daca e in concediu intr-o anumita data:

```
DELIMITER //
DROP PROCEDURE IF EXISTS verificareAconcediu;
CREATE PROCEDURE verificareAconcediu(id int, p varchar(50), datav date)
BEGIN
SELECT concediu FROM programangajati Pr, utilizatori U, angajat A
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND A.id_angajat = id AND U.parola = p AND datapr = datav;
END
// DELIMITER
```

Pentru a verifica daca o unitate are program in ziua respectiva:

```
DELIMITER //
DROP PROCEDURE IF EXISTS CautareProgramAzi;
CREATE PROCEDURE CautareProgramAzi(n varchar(255), p varchar(255), t varchar(255))
BEGIN
SELECT ora_incepere AS 'Ora incepere', ora_plecure AS 'Ora plecare', unit.denumire AS 'Unitate'
FROM Utilizatori U, Angajat A, Programangajati Pr, Unitati_medicale unit
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat AND Pr.unitate =
unit.id_unitate_medicala
AND U.num = n AND U.prenume = p AND tip = t AND datapr = current_date();
END
// DELIMITER
```

Pentru a modifica programul unui angajat:

```

DELIMITER //
DROP PROCEDURE IF EXISTS ModificareProgramAngajat;
CREATE PROCEDURE ModificareProgramAngajat(n varchar(255), p varchar(255), t varchar(255),
datav date, inceput int , final int)
BEGIN
UPDATE programangajati Pr, angajat A, utilizatori U
SET ora_incepere = inceput, ora_plecare = final
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND U.num = n AND U.prenume = p AND tip = t AND Pr.datapr = datav;
END
// DELIMITER

```

Pentru a adauga o zi de concediu la un anagajat:

```

DELIMITER //
-- DROP PROCEDURE IF EXISTS AdaugareConcediu;
CREATE PROCEDURE AdaugareConcediu(n varchar(255), p varchar(255), t varchar(255), inceput date
, final date)
BEGIN
DECLARE c int;
SELECT COUNT(id_programare) INTO c
FROM programari P, unitati_medicale UM, programangajati Pr, angajat A, utilizatori U
WHERE U.id_ca_angajat = A.id_angajat AND id_angajat = Pr.id_ca_angajat AND id_unitate_medicala
= P.unitatea
AND id_unitate_medicala = Pr.unitate AND num = n AND prenume = p AND tip = t;

IF(c = 0)
then UPDATE programangajati Pr, angajat A, utilizatori U
SET ora_incepere = null, ora_plecare = null, concediu = 1
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND U.num = n AND U.prenume = p AND tip = t AND datapr BETWEEN inceput AND final;
end if;
END

```

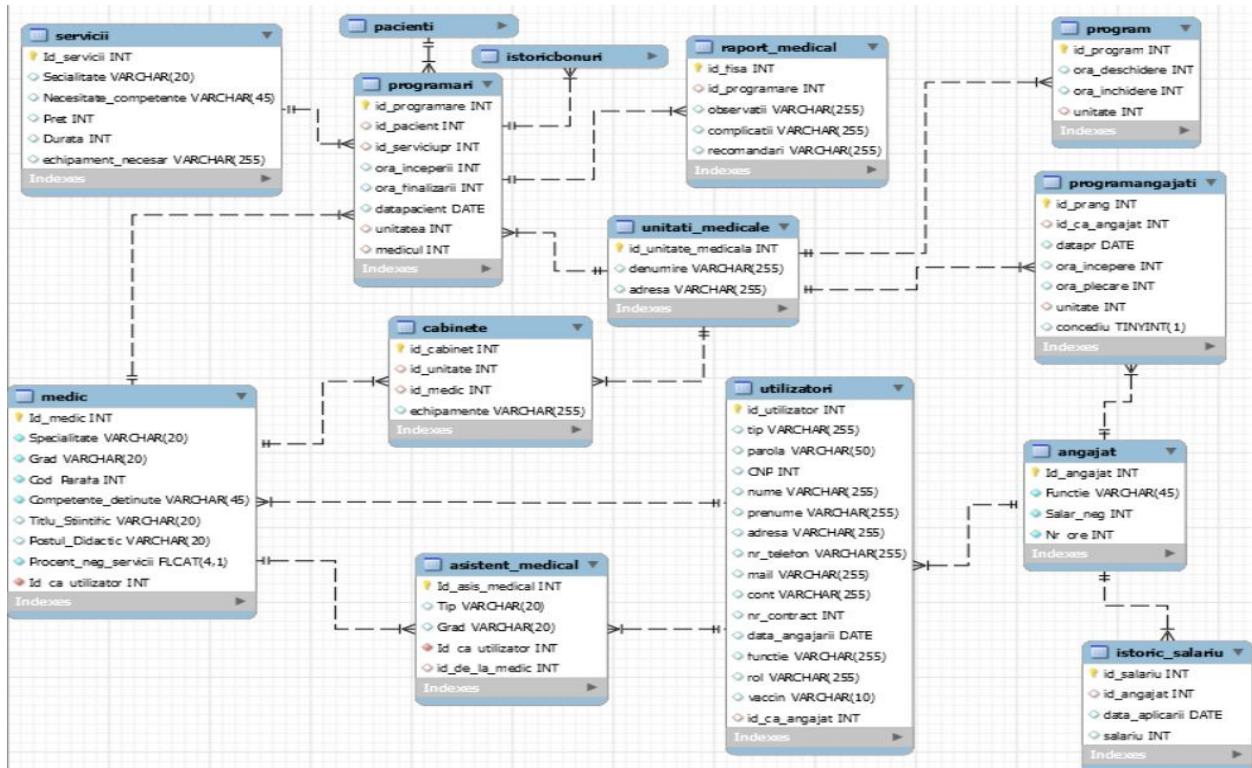
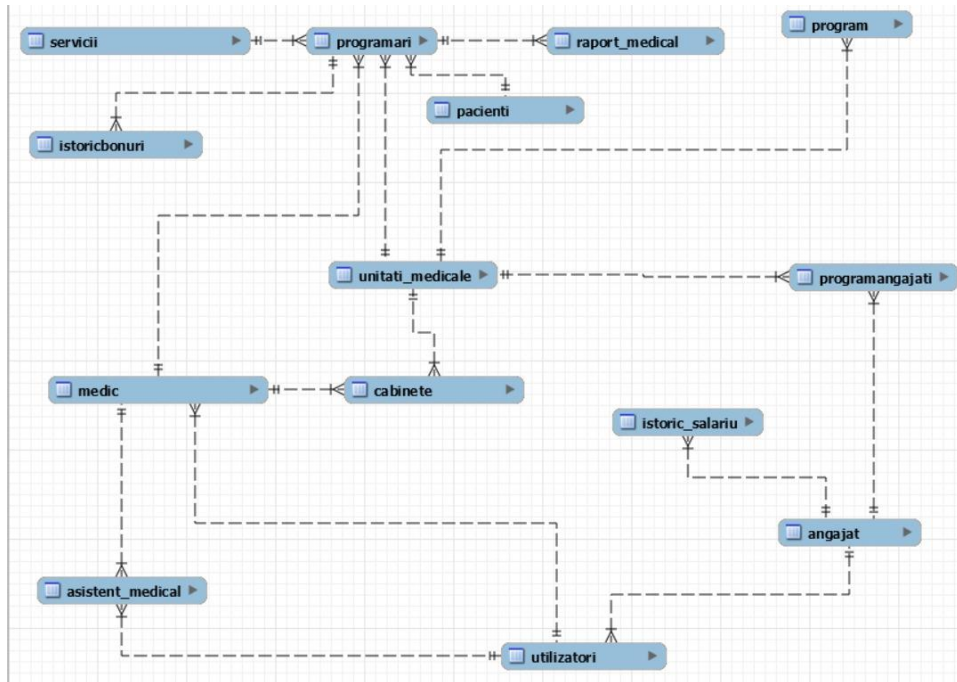
Pentru a verifica daca intervalul orar introdus pentru o consultatie e egal cu intervalul orar necesar unei consultatii:

```

// DELIMITER
DELIMITER $$
CREATE PROCEDURE verificare_or1( s1 int, s2 int, s3 int, oi int , orf int )
BEGIN
select if(((s1+s2+s3)/60 =orf-oi),1,0) as ifc ;
END $$
DELIMITER ;

```

Diagrama UML:



4. Implementarea aplicatiei:

1. Pagina principala



Acesta este prima pagina care se deschide atunci cand utilizatorul acceseaza aplicatia. Si acestuia i se ofera posibilitatea de a intra la sectiunea de gestiune a resurselor umane, de operatii financiar contabile, de gestiune a activitatilor operationale, de administrator, super-administrator, respectiv vaccine dand click pe butoanele disponibile pe interfata.

2. Administrator

Pe aceasta pagina administratorului i se permite sa adauge, modifice, stearga utilizatori din tabelul Utilizatori, inasa doar daca acestia nu sunt de tip administrator sau super-administrator.

Acesta trebuie sa completeze casutele aflate in stanga cu datele care trebuie introduce in tabel, apoi sa apese butonul Inserirre/Update pentru a face operatiile respective.

Pentru a sterge e de ajuns sa completeze doar id-ul utilizatorului.

Acesta poate sa caute utilizatori in tabela in functie de id-ul lor sau dupa nume, selectand din ComboBox id-ul sau completand numele in casuta de text.

3. Super-Administrator

Pe aceasta pagina super-administratorului i se permite sa adauge, modifice, stearga utilizatori din tabelul Utilizatori.

Acesta trebuie sa completeze casutele aflate in stanga cu datele care trebuie introduce in tabel, apoi sa apese butonul Inserare/Update pentru a face operatiile respective.

Pentru a sterge e de ajuns sa completeze doar id-ul utilizatorului.

Acesta poate sa caute utilizatori in tabela in functie de id-ul lor sau dupa nume, selectand din ComboBox id-ul sau completand numele in casuta de text.

Modulul I. Resurse umane

In acest modul se pot vizualiza informatii despre propria persoana in ceea ce priveste orarul pentru angajti (receptioner, medic, asistent), pentru exectorul financiar exista sectiunea de program si concediu al tuturor angajatilor sectiuni asemanatoare cu cele ale angajatilor de tip resurse umane; pentru ei se adauga posibilitatea de modificare.

Prima fereastra este cea de alegere, fiecare utilizator isi selecteaza categoria potrivita.

NUME

PRENUME

FUNCTIA

Cautare angajat*

*Se va afisa programul pentru ziua curenta.

Nume	Prenume	Functie
Baum	Andreea	resurse
Chitos	Mara	receptioner
Coroian	Rebrca	economic
Costin	Ilana	asistent
Ferenc	Tudor	receptioner
Grecu	Alexandra	medic
Grigore	Maria	medic
Grigore	Stefana	asistent
Moldovan	Alex	admin
Muresan	Ioana	asistent
Peles	Ioan	receptioner

DATA

PERIOADA -

Cautare zi

Cautare perioada

Pentru a modifica programul va rugam sa introduceti intervalul orar

Or

Se

Introduceti datele mentionate pentru a afisa programul saptamanal al unei unitati medicale.

Data	Program
2021-01-01	8 - 16
2021-01-02	8 - 16
2021-01-03	9 - 23
2021-01-04	8 - 16
2021-01-05	8 - 16

PERIOADA -

UNITATE MEDICALA

Afisare unitati

Afisare program

Pentru a modifica programul va rugam sa introduceti intervalul orar

Ora de inceput - Ora de final

Si data

Modificare

BACK

Pentru logarea ca administrator resurse umane, daca datele au fost introduse bine, va aparea fereastra pentru care am atasat imaginea din stanga. Butonul de „Afisare angajati” listeaza in tabel datele posibile de introducere pentru a incepe cautarea.

„Cautare angajat*” va afisa programul din ziua curenta. Pentru a cauta o anumita data calendaristica se poate folosi „Cautare zi” dupa ce e introdusa data in campul liber „Data”. La fel se poate proceda si pentru cautarea unei perioade. Modificare se face dupa apasarea butonului daca sunt completate toate campurile necesare.

Tabul de „Orar saptamanal” este dedicat unitatilor medicale, inainte de a utiliza orice buton se introduce perioada, dupa putand fi afisat sau modificat programul.

Butonul de „Back” va face trimitere inapoi la logare.

Asemenea celor de la resurse umane, expertului financiar interfata va cuprinde atat sectiunea de cautare angajat cat si cele de orar al unitatilor si concediu. Diferenta consta in drepturile de scriere pe care doar departamentul resurse umane o au.

Pentru „Concediu” vine in ajutor butonul de afisare angajati, daca se apasa pe o linie din tabelul angajat afisat datele din stanga tabelului se vor completa automat cu cele din randul ales.

Este posibila afisarea concediilor din tot istoricul angajatului si modificarea programului in vederea adaugarii concediului. In spatele butonului de „Modificare” este apelata o procedura [call AdaugareConcediu](#). Concediul nu poate fi adaugat daca angajatul (medic sau asistent) are deja programari pentru data respectiva.

```
CREATE PROCEDURE AdaugareConcediu(n
varchar(255), p varchar(255), t varchar(255), inceput
date , final date)
BEGIN
```

NUME

PRENUME

FUNCTIA

Afisare concedii

Introduceti perioada pentru a adauga concediu *

Prima zi - Ultima zi

Modificare

Afisare angajati

BACK

In cazul existentei programarilor pentru data introdusa, concediul nu va fi adaugat.

```

DECLARE c int;
SELECT COUNT(id_programare) INTO c
FROM programari P, unitati_medicale UM, programangajati Pr, angajat A, utilizatori U
WHERE U.id_ca_angajat = A.id_angajat AND id_angajat = Pr.id_ca_angajat AND id_unitate_medicala
= P.unitatea
AND id_unitate_medicala = Pr.unitate AND nume = n AND prenume = p AND tip = t;

IF(c = 0)
then UPDATE programangajati Pr, angajat A, utilizatori U
SET ora_incepere = null, ora_plecare = null, concediu = 1
WHERE U.id_ca_angajat = A.id_angajat AND Pr.id_ca_angajat = A.id_angajat
AND U.nume = n AND U.prenume = p AND tip = t AND datapr BETWEEN inceput AND final;
end if;
END

```

Id	Specialitatea	Competente	Pret	Durata	Echipament
1	cardiologie	medic	100	60	ecograf
2	cardiologie	medic	200	60	holter
3	neurologie	medic	100	60	CT
4	pediatrie	medic	100	60	Stetoscop si tensi
5	ortopedie	medic	100	60	punere ghips
6	neurologie	asistent	80	60	CT

Pentru personalizarea de servicii medicale, tab accesibil doar departamentului resurse umane, se pot adauga date, pentru a completa mai usor se introduc automat cand este apasata o linie din tabel, se modifica informatiile din campurile dorite, iar butonul „Adauga” va duce la afisarea unui mesaj daca au fost introduse cu succes sau nu.

Angajatii nu au nevoie de logare inaintede a vizualiza fereastra, deoarece pentru a cauta informatii legate de program sau concediu trebuie sa completeze aici id-ul si parola.

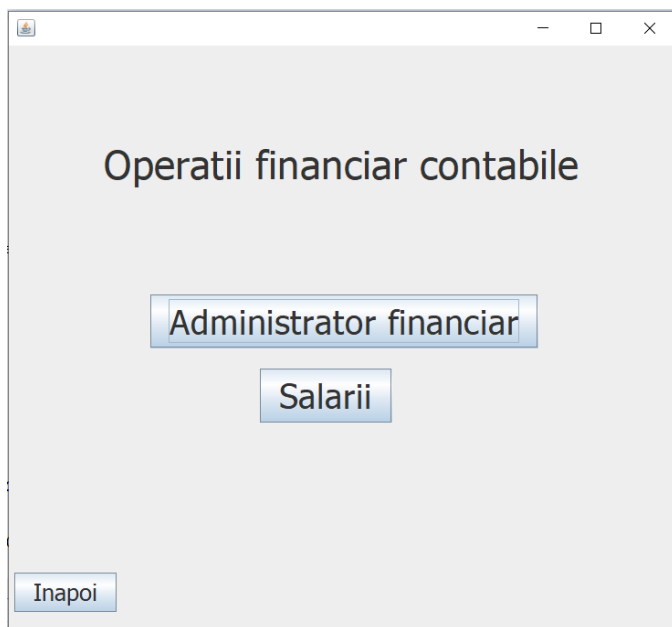
Pot cauta personalizat programul, cu alte cuvinte pot cauta o anume zi calendaristica sau perioada.

Inclusiv pentru o zi pot vizualiza programarile.

Data	Ora inceput	Ora final	Unitatea
2021-01-03	10	13	UnitASI
2021-01-04	8	16	UnitASI
2021-01-04	18	22	UnitASI
2021-01-05	8	16	UnitASI
2021-01-06	8	16	UnitASI
2021-01-07	8	16	UnitASI
2021-01-08	8	16	UnitASI
2021-01-09	8	16	UnitASI
2021-01-10	8	16	UnitASI
2021-01-11	8	16	UnitASI
2021-01-12	8	16	UnitASI
2021-01-13	8	16	UnitASI
2021-01-14	8	16	UnitASI
2021-01-15	8	16	UnitASI

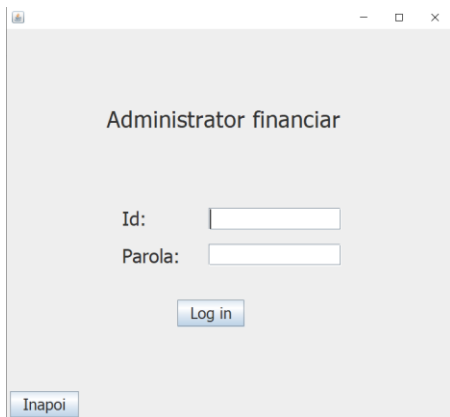
Modulul II. Operatii financiar contabile

In acest modul, un utilizator va putea sa isi verifice atat salariul, cat si istoricul salariilor. Mai mult, un medic va putea sa verifice si profitul generat in luna actuala, defalcat pe venituri si cheltuieli. La selectarea optiunii "Operatii dinanciar contabile", se va deschide urmatoarea fereastră:



1. Administrator financiar

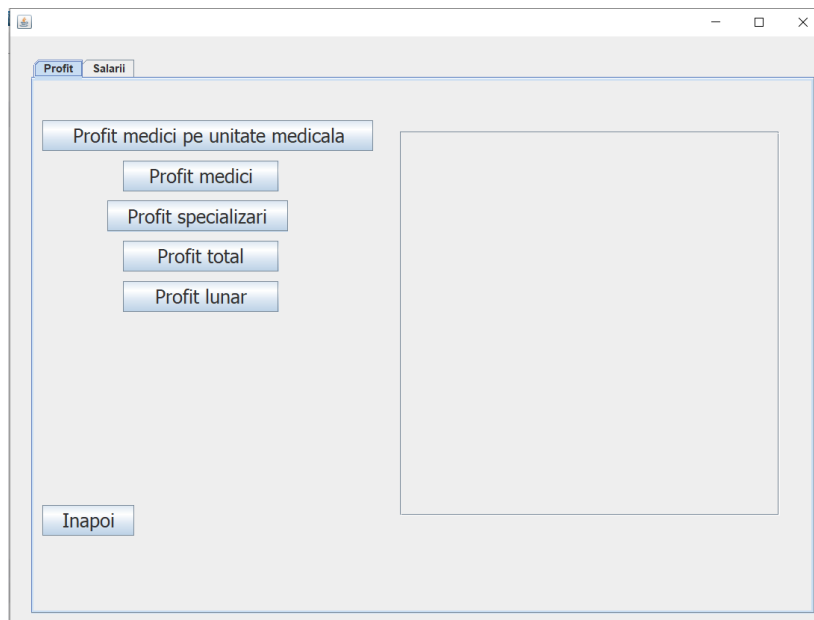
Inainte de a i se deschide fereastră de unde poate selecta orice doreste sa vada, acesta trebuie sa se conecteze.



este administrator financiar (tip='economic'), se va deschide urmatoarea fereastră de unde poate monitoriza fiecare angajat in parte. In caza contrar, se va afisa un mesaj "Wrong id and password".

Daca serverul detecteaza ca cel care doreste sa acceseze sectiunea nu

Fereastra pentru administratorul financiar ce se deschide dupa ce a reusit conectarea, prezinta 5 butoane, fiecare in parte lasandu-l pe administratorul financiar sa observe profitul generat de fiecare medic in parte, pe o anumita locatie sau specialitate medicala din luna precedenta. De asemenea, se poate vedea si profitul total adus de lantul de policlinici, dar si profitul lunar adus de asemenea de intreg lantul de policlinici. Profitul se poate vizualiza doar in cazul in care exista activitate in luna precedenta.



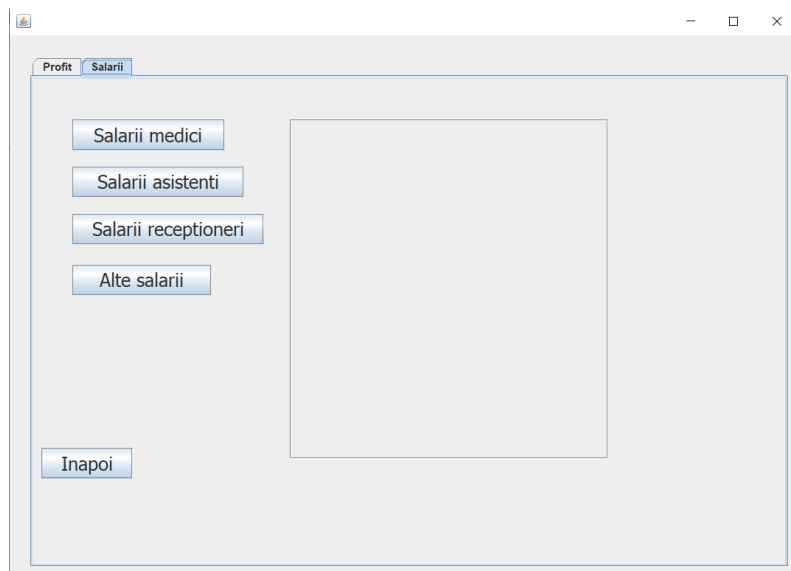
Exemplu:

- Selectarea profitului generat de medici pe unitate medicala

```
SELECT unitati_medicale.adresa, SUM(sumaplatita) as venituri,
IF(COUNT(programari.medicul),3000+SUM(medic.Procent_neg_servicii/100 * sumaplatita),3000) as
cheltuieli, SUM(sumaplatita)-3000-SUM(medic.Procent_neg_servicii/100 * sumaplatita) as profit
from angajat, utilizatori, istoricbonuri,medic,programari, unitati_medicale
where utilizatori.Id_ca_angajat=angajat.Id_angajat
and programari.medicul=medic.Id_medic
and medic.Id_ca_utilizator=utilizatori.id_utilizator
and istoricbonuri.id_de_la_programare=programari.id_programare
and programari.unitatea=unitati_medicale.id_unitate_medicala
and MONTH(programari.datapacient)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH)
group by unitati_medicale.adresa;
```

adresa	venituri	cheltuieli	profit
Tg. Mures	100.0	3060.0	-2960.0
Bucuresti	500.0	3427.5	-2927.5

Cea de a doua fila de la fereastra administratorului financiar ne ofera informatii despre salariile angajatilor din luna precedenta.



Salariile medicilor sunt calculate in functie de ce programari au efectuat. In cazul in care nu au activitate pe acea luna, ei vor avea un salariu (minim) de 3000 lei.

Butonul "Alte salarii" ne ofera informatii despre salariile altor angajati care nu sunt medici, asistenti sau receptioneri.

Exemplu:

- afisarea salariilor medicilor:

```
SELECT nume,prenume,
3000+SUM(medic.Procent_neg_servicii/100 * sumaplatita) as salariu
from istoricbonuri,utilizatori,medic,programari
WHERE medic.id_medic=programari.medicul
and programari.id_programare=istoricbonuri.id_de_la_programare
and utilizatori.id_utilizator=medic.id_ca_utilizator
and MONTH(programari.datapacient)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH)
group by medic.id_medic;
```

nume	prenume	salariu
Grigore	Maria	3060.0
Popa	Ion	3427.5

2. Salarii

Cea de a doua optiune din fereastra initiala "Operatii financiar contabile", ofera informatii despre angajatul ce o acceseaza pe luna precedenta. Pentru a putea selecta una din optiunile afisate, mai intai trebuie sa se inregistreze, pentru a i se putea furniza informatii doar despre el.

Optiunea "Profit generat" este disponibila doar celor care sunt medici.

Istoricul salariilor pentru medici se arata doar in momentul in care un medic are activitate pe luna respective, altfel se considera salariul de baza 3000 lei. Pentru restul personalului, cu un salariu fix, se va afisa data de la care a fost aplicat un salariu si salariul respective.

Exemplu:

- medic
 - salariul actual

```
SELECT MONTH(CURRENT_DATE - INTERVAL 1 MONTH) as luna,
IF(COUNT(programari.medicul),3000+SUM(medic.Procent_neg_servicii/100 * sumaplatita),3000) as salariu
FROM istoricbonuri, programari,medic
WHERE MONTH(programari.datapacient)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH)
AND istoricbonuri.id_de_la_programare=programari.id_programare
and programari.medicul=medic.id_medic
and medic.id_ca_utilizator=user.getText();
```

Id:

Parola:

Afiseaza salariu actual

Istoric salarii

Profit generat

luna	salariu
12	3000.0

Inapoi

➤ istoric salarii

```
SELECT m.luna, IF(COUNT(programari.medicul),3000+SUM(medic.Procent_neg_servicii/100 *
sumaplatita),3000) as salariu
FROM (SELECT 'January' AS luna
UNION SELECT 'February' AS luna
UNION SELECT 'March' AS luna
UNION SELECT 'April' AS luna
UNION SELECT 'May' AS luna
UNION SELECT 'June' AS luna
UNION SELECT 'July' AS luna
UNION SELECT 'August' AS luna
UNION SELECT 'September' AS luna
UNION SELECT 'October' AS luna
UNION SELECT 'November' AS luna
UNION SELECT 'December' AS luna) AS m, istoricbonuri, programari, medic
WHERE istoricbonuri.id_de_la_programare=programari.id_programare
and programari.medicul=medic.id_medic
and medic.id_ca_utilizator= + user.getText() +
and m.luna=monthname(programari.datapacient)
group by m.luna;
```

Afiseaza salariu actual

Istoric salarii

Profit generat

luna	salariu
October	3140.0
January	3280.0

Inapoi

➤ profit generat

```
SELECT MONTH(CURDATE()) as luna,
IF(COUNT(programari.medicul),SUM((100-medic.Procent_neg_servicii)/100 * sumaplatita)-3000,-3000) as profit
FROM istoricbonuri, programari,medic
WHERE MONTH(programari.datapacient)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH)
AND istoricbonuri.id_de_la_programare=programari.id_programare
and programari.medicul=medic.id_medic
and medic.id_ca_utilizator=user.getText();
```

Id: 1

Parola:

Afiseaza salariu actual

Istoric salarii

Profit generat

luna	profit
1	-3000.0

Inapoi

- orice angajat in afara medicilor

➤ salariu actual

```
SELECT nume, prenume, salar_neg*nr_ore*(28-SUM(IF(programangajati.concediu=1,1,0))) as salariu FROM
Angajat, utilizatori, programangajati
WHERE angajat.id_angajat=utilizatori.id_ca_angajat
AND MONTH(programangajati.datapr)=MONTH(CURRENT_DATE - INTERVAL 1 MONTH)
AND utilizatori.id_utilizator= user.getText() ;
```

Id: 2

Parola:

Afiseaza salariu actual

Istoric salarii

Profit generat

nume	prenume	salariu
Grigore	Stefana	8960

Inapoi

➤ istoric angajat

```
SELECT salariu, data_aplicarii  
FROM Istoric_Salariu, angajat, utilizatori  
WHERE angajat.id_angajat=Istoric_Salariu.id_angajat  
AND angajat.id_angajat=utilizatori.id_ca_angajat  
AND utilizatori.id_utilizator=+ user.getText() +;
```

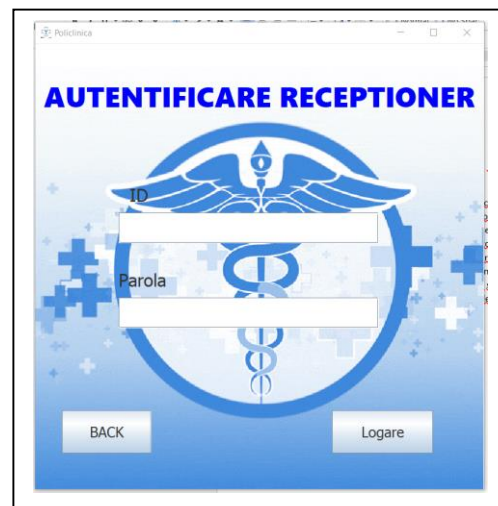
The screenshot shows a web application window with a light gray background. On the left, there are two input fields: 'Id:' with the value '2' and 'Parola:' with four black dots. To the right of these fields are three blue buttons: 'Afiseaza salariu actual', 'Istoric salarii', and 'Profit generat'. Below the input fields is a table with two columns: 'salariu' and 'data_aplicarii'. The table contains three rows of data. At the bottom left, there is a blue button labeled 'Inapoi'.

salariu	data_aplicarii
1500	2007-10-11
3000	2014-10-11
4000	2020-10-11

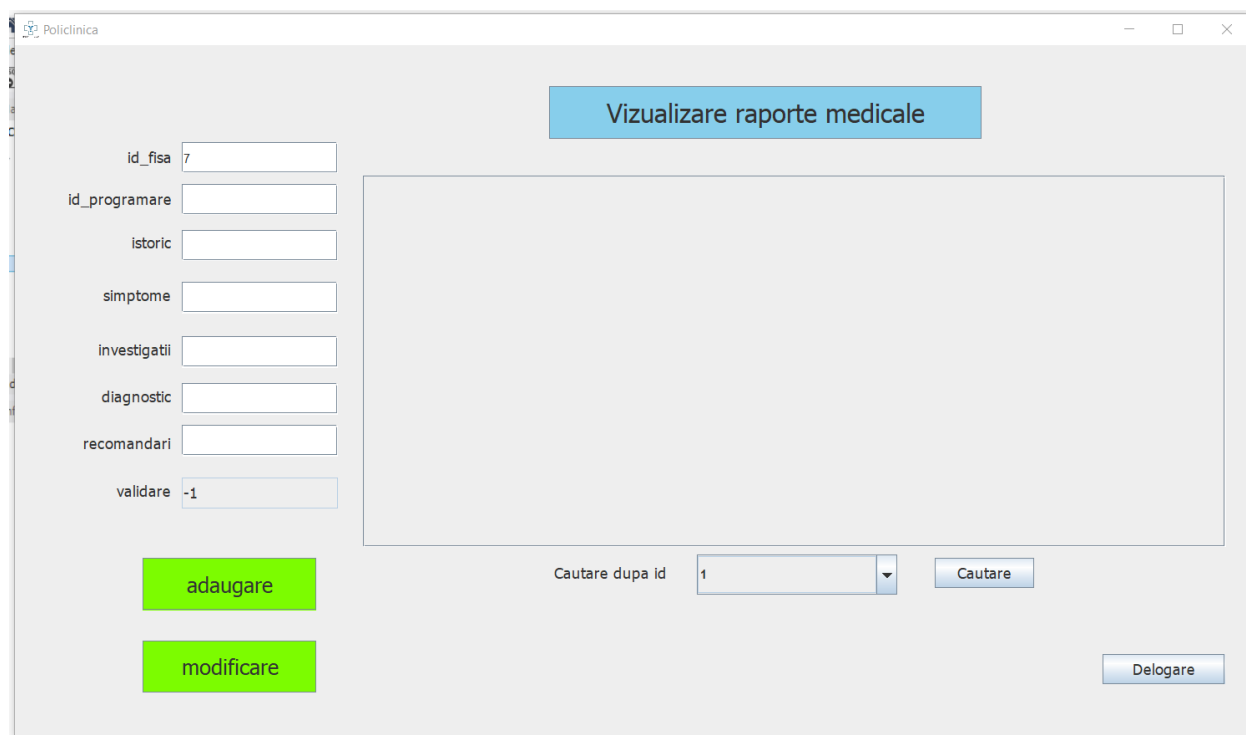
Modul III: Gestiunea activitatilor operationale



Atunci cand acceseaza aceasta pagina, utilizatorul trebuie sa isi aleaga mai intai drept ce se logheza, apasand in continuare butoanele corespunzatoare, deschizandu-se mai apoi o pagina in care trebuie sa-si introduca id-ul si parola pentru a putea continua mai departe. Daca a introdus datele corecte i se deschide urmatoarea pagina corespunzatoare, altfel primeste o atentionare „ Wrong id or password”.



Daca s-a autentificat drept asistent medical:



I se deschide fereastra de sus, astfel ca acesta poate vizualiza rapoartele medicale, poate adauga si modifica rapoarte medicale, completand casutele text de langa tabel si poate cauta rapoare dupa id. Acesta mai poate accesa datele din tabel care se copiaza automat in casute in momentul cand acesta da click pe un rand din tabel.

Daca s-a autentificat ca medic:

The image displays four screenshots of a medical application interface, arranged in a 2x2 grid. Each screenshot shows a different module of the application, with a sidebar on the left containing tabs for 'Pacienti', 'Programari pacienti', 'Raportare Medica', and 'Adaugare servicii'.

- Top Left Screenshot (Pacienti):** This module allows viewing and searching for patients. It features a list of patient fields (id_pacient, alergii, boli, nume, prenume, data_nasterii, sexul, domiciliu, cetatenie, asigurare) and a large table area. A button 'Vizualizarea tuturor pacientilor' is present. Search filters include 'Cautare pacient dupa id:', 'Cautare pacient dupa nume:', 'Cautare pacient dupa prenume:', and 'Cautare pacient dupa id-ul programarii:'. A 'Delogare' button is at the bottom right.
- Top Right Screenshot (Programari pacienti):** This module displays patient appointments. It includes fields for appointment details (id_programare, id_pacient, id_serviciu, id_serviciu2, id_serviciu3, ora_inceperii, ora_finalarii, data, unitatea, medicul) and a large table area. A button 'Vizualizare programari' is present. A section 'Afiseaza doar programările cu:' includes filters for 'id_programare', 'id_pacient', 'id_serviciu', 'ora_inceperii', 'ora_finalarii', 'unitatea', and 'medicul'. A 'Delogare' button is at the bottom right.
- Bottom Left Screenshot (Raportare Medica):** This module is for managing medical reports. It includes fields for report details (id_fisa, id_programare, istoric, simptome, investigatii, diagnostic, recomandari, validare) and a large table area. Buttons 'Vizualizare rapoarte medicale', 'Inserare raport medical', and 'Validare' are present. Search filters include 'Cautare raport medical cu id-ul fisei' and 'Cautare raport medical cu id-ul pacientului'. A 'Delogare' button is at the bottom right.
- Bottom Right Screenshot (Adaugare servicii):** This module is for managing services. It includes fields for service details (id_serviciu, specialitate, necesitate_competente, pret, durata, echipament_necesar) and a large table area. Buttons 'Adaugare servicii', 'Vizualizare servicii', and 'Stergere servicii' are present. A 'Delogare' button is at the bottom right.

Pe primul panel, cel cu Pacienti, acesta poate vizualiza informatiile despre pacienti, dar poate si cauta pacienti in functie de id, nume, prenume, programare.

Pe al doilea panel, cel cu Programari Pacienti, acesta poate vedea programarile din ziua respectiva si sa caute programarile in functie de anumite date in cazul in care are nevoie de una anume.

Pe al treilea, acesta poate vizualiza, adauga si valida rapoartele medicale, dar poate si sa caute in functie de id-ul fisei, id-ul programarii sau dupa id-ul pacientului.

Pe ultimul panel, acesta poate vizualiza, adauga sau sterge un serviciu.

Utilizatorul poate accesa mai usor datele din tabele, dand click pe un anumit rand din tabel, iar datele din acel rand vor aparea in casutele din stanga tabelului.

Daca s-a autentificat ca receptioner:

Pe primul panel acesta poate vizualiza, adauga, sterge utilizatorii, poate modifica datele lor dar poate si cauta un anume pacient dupa nume sau prenume.

Polyclinic

Pacienti Programari Bon Fiscal Programare speciala

id_pacient 11

alergii

boli

nume

prenume

data_nasterii

sexul

domiciliu

cetatenie

asigurare

Vizualizare Pacienti Inserare Pacient

Modificare date de la un pacient Stergere Pacient

Cautare dupa nume

Cautare dupa prenume

Delogare

Pe aceasta pagina receptionerul poate vizualiza atat programari, medici, serviciile, cabinetele si unitatile medicale, pentru a-l ajuta sa completeze datele in casutele din stanga tabelului. Pentru a adauga/modifica/sterge o programare, aceasta trebuie sa fie verificata. Dand click pe butonul de verificare, se verifica daca in acelasi timp mai are loc o alta programare, la acelasi cabinet de la

aceasi unitate si cu acelasi doctor, intre programari trebuind si sa existe o pauza de o ora de odihna pentru medici. Poate cauta eventual si programari in functie de cateva date, pentru a-i simplifica alegerea datelor pe care vrea sa le introduca in tabel. Daca o programare nu este verificata si validata in urma verificarii aceasta nu poate fi modificata, adaugata sau stearsa.

Pe cel de al treilea panel acesta poate vizualiza bonurile fiscale introduse pana acum, dar poate adauga, modifica si sterge un anumit bon. La completarea casutei cu id-ul programarii se calculeaza automat casuta de la suma platita cu valoare corespunzatoare serviciilor efectuate.

Polyclinic

Pacienti Programari Bon Fiscal Programare speciala

id_bon	id_de_la_programare	sumaplatita
1	2	300
2	2	300
3	3	100
4	4	100
5	5	200

id_bon 6

id_de_la_programare

sumaplatita

Vizualizare istoric bonuri

Adaugare bon

Modificare bon

Stergere bon

Delogare

Policlinica

Pacienti Programari Bon Fiscal Programare speciala

Selectati din tabele serviciul, unitatea si medicul

serviciul 1 NULL 0
serviciul 2 NULL 0
serviciul 3 NULL 0

id_serviciu timp

echipament necesar

unitatea cabinetul

medic id_angajat al medicului

data

Introduceti manual aceste date:

id_programare 6
id_pacient
ora_inceperii
ora_finalizarii

Dupa ce ati introdus datele, verificati daca e valabila programarea si apoi adaugati-o

Verificare programarea
Adaugare programare

Serviciu 1 Serviciu 2 Serviciu 3 Vizualizeaza serviciile

Vizualizeaza programari Vizualizeaza unitati medicale

Vizualizeaza medicii pentru acel serviciu

Vizualizeaza zilele cand e medicul e la unitate

Delogare

Cel de-al patrulea panel este dedicat unei modalitati de programare mult mai usoare, ne mai fiind necesar sa tot accesam diferite tabele pentru a adauga o programare valida. Mai intal trebuie sa apasam vizualizare servicii, bifand cate un serviciu de langa si dand click in tabel pentru a alege serviciul dorit, completandu-se automat casutele din stanga. Dupa ce ne-am ales unul sau mai multe servicii (maxim 3) dand click pe vizualizare medici pentru acel serviciu putem vizualiza medicii care ar putea efectua serviciul dorit de pacient. Dand click pe un anumit medic se completeaza automat datele acestuia in stanga. Iar apoi trebuie dat click pe vizualizare zile cand medicul e la unitate pentru a vedea zile cand medicul e prezent in unitatea si cabinetul respectiv serviciilor. Alegand un rand din tabel se completeaza automat si data dorita, astfel ca mai ramane de adaugat doar id-pacient, ora_inceperii si ora_finalizarii. Dupa ce am completat toate campurile pentru a verifica daca programarea e corecta si a o adauga apasam pe verificare, astfel verificandu-se daca nu mai exista programari in acelasi timp, la acelasi doctor, cabinet si unitate, dar si daca diferenta dintre ora_inceperii si ora_finalizarii e egala cu timpul care trebuie alocat serviciilor alese. Daca toate datele sunt corecte se poate adauga programarea, altfel vor aparea mesaje corespunzatoare erorilor intampinate.

Vaccinare

Policlinica - VACCINARE

Nota: sunt acceptate modificari ulterioare.

Pentru a va inregistra in vederea efectuării vaccinului "Pfizer"
va rugam sa completati campurile urmatoare:

ID

Nume

Prenume

Sunt de acord cu vaccinarea


Parola

Trimite

Va multumim pentru alegerea facuta!

Pfizer

BACK



Pe aceasta pagina utilizatorul isi poate introduce id-ul, numele, prenumele, acordul si parola de utilizator pentru a-si completa formularul referitor la sectiunea de vaccin.

Dupa ce apasa trimitere se completeaza automat acordul i tabelul utilizatori la coloana „vaccin”.

5. Securitate si justificarea solutiei alese

Am ales aceasta forma de implementare deoarece este accesibila atat utilizatorului cat si programatorului. Este structurata pe module, fiecare modul are tipul de utilizatori care are acces si logarea prin id si parola.

Pentru cautare, inserare sau stergere (data, angajat, sau alte servicii) avem si optiunea de a afisa datele existente pentru a ne orienta si a gasi mai rapid ceea ce dorim. Si avand in vedere situatia actuala am avut grija sa pastram si o evidenta a persoanelor care doresc sa se vaccineze.

Pe partea financiara, ca dezvoltare ulterioara administratorul financiar ar putea modifica salariul oricarui angajat. Mai mult, acesta ar putea sa caute doar in functie de nume, adresa sau policlinica pentru a vedea pentru o singura persoana. De asemenea, un angajat ar putea sa vada pentru o singura luna ce salariu a avut, fara a mai cauta in istoricul salariilor.

O posibila dezvoltare ulterioara pentru modulul de resurse este adăugarea unui concediu medical "de urgență" programările fiind reatribuite altui angajat.

Ca securitate am utilizat frame-uri de login, in care utilizatorul trebuie sa isi introduca id-ul de utilizator si parola, astfel incat daca acesta introduce o parola gresita nu are acces la paginile urmatoare din aplicatie, altfel se vor deschide urmatoarele pagini in functie de pagina de pe care provine.