Proiect la discipina baze de date

Aplicatie de gestiune a unui lanț de policlinici

Proiect realizat de studenții:

-PINZARIU TEODOR ALEXANDRU

-TULBURE ROBERT ALEXANDRU

-TIMIS PETRU GHEORGHE

Grupa 30223

Profesor indrumator : teodora tat

Cuprins:

1.Tema proiectului

2.Proiectarea conceptuala a bazei de date

3.Solutia de transformare in relational

4.Descrierea bazei de date relationale

5.Argumentarea nivelului de normalizare

6.Codul in SQL

7.Functionalitatea aplicatiei

8.Concluzii si dezvoltari ulterioare

1.Tema proiectului

Aplicatia oferita reprezinta un sistem informatic de gestiune a unui lant de policlinici prin intermediul unei interfete grafice.Aceasta gestioneaza atat informatiile cu privire la utilizatorii care se logheaza in aceasta baza de date cat si informatiile cu privire la angajati, care poti fi de mai multe tipuri:

-medic

-asistent

-receptioner

-expert-financiar-contabil

-inspector resurse umane

De asemenea baza de date stocheaza atat informatiile personale cat si detalii cu privire la orarul de lucru,concedii si salariile angajatiilor.Totodata clientii se pot programa astfel incat la finalul consultatiei acestia sa poata avea cate un raport medical in ceea ce priveste propria persoana pentru a se putea vedea parcursul pacientului pe perioada consultatiei.In acelasi timp fiecare policlinica are atat orarul sau propriu cat si servicii unice in ceea ce priveste prestatia cu privire la profesionalismul medical.

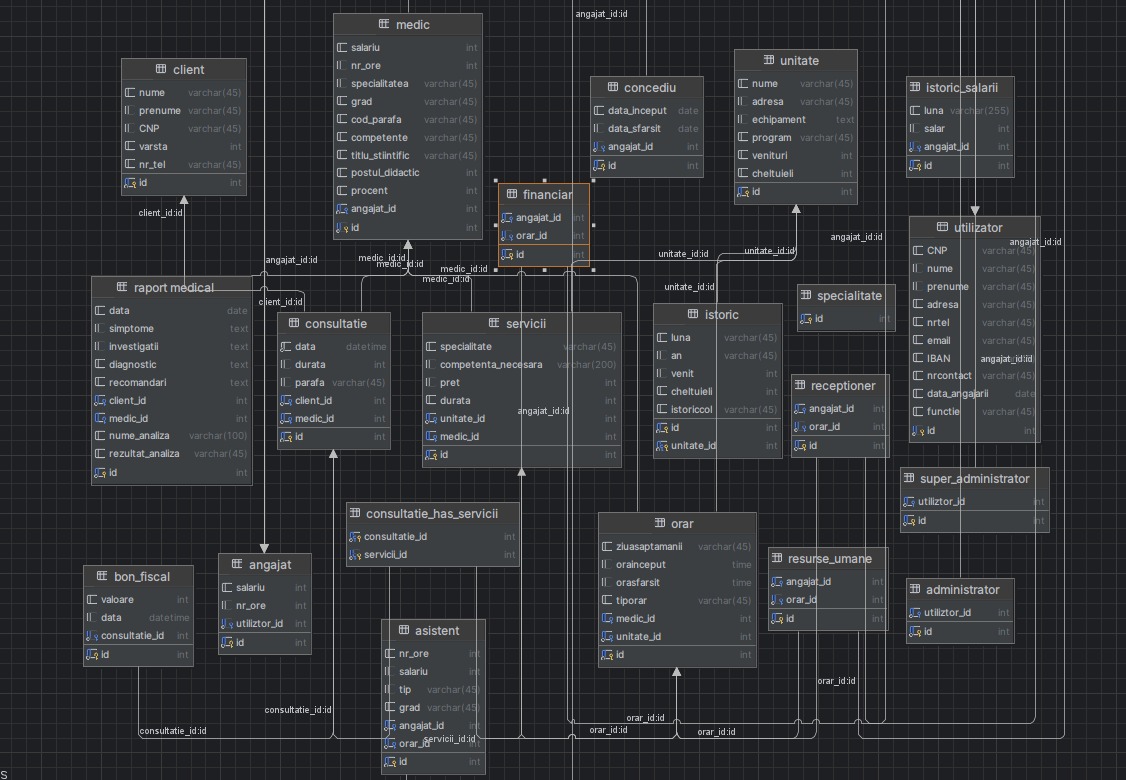
Utilizatorii de tip administrator pot adauga,modifica si sterge informatii cu privire la alti utilizatori.De asemenea exista si un super-administrator care poate opera inclusiv asupra utilizatorilor de tip administrator.

Inspectorii de tip resurse umane pot cauta orice angajat,de orice tip,in functie de anumiti parametrii specifici care sunt numele,prenumele si functia.Totodata atat acest tip de inspector cat si expertul financiar contabil pot consulta orarul saptamanal si informatiile despre concediile efectuate.Expertul financiar contabil poate vizualiza informatii cu privire la profitul realizat de lantul de policlinici iar acest profit este calculat ca diferenta dintre venituri si cheltuieli

Angajatii de tip receptioner,asistent medical si medic pot vizualiza orarul si concediul doar in ceea ce priveste propria persoana.Orice angajat poate vizualiza detalii cu privire la salariile obtinute in lunile precedente.Angajatii de tip receptioner au aptitudinea de a realiza o programare pentru un anumit pacient.De asemenea pe baza consultatiei receptionerul emite un bon fiscal.Angajatii de tip asistent medical pot completa informatii in rapoartele medicale.Medicul poate vizualiza pacientii care sunt programati in ziua respectiva la el.Pentru fiecare pacient, medicul completeaza un raport medical.Medicul va putea gestiona lista serviciilor medicale.

De asememea,in proiect este inclusa si o sectiune in care se poate calcula salariul unui medic bazat pe consulatiile facute si nu pe salariu iar medicul are optiunea de a-si personaliza serviciile medicale.

2.Proiectarea conceptuala a bazei de date

**

#### 3.Solutia de transformare in relational

Relații de Tip Unu-la-Unu (One-to-One):

1. Tabela administrator cu Tabela utilizator:
   * Un administrator (administrator.id) corespunde unui utilizator (utilizator.id).
   * Un utilizator (utilizator.id) are cel mult un administrator asociat (administrator.id).
2. Tabela angajat cu Tabela utilizator:
   * Un angajat (angajat.id) corespunde unui utilizator (utilizator.id).
   * Un utilizator (utilizator.id) are cel mult un angajat asociat (angajat.id).
3. Tabela medic cu Tabela angajat:
   * Un medic (medic.id) corespunde unui angajat (angajat.id).
   * Un angajat (angajat.id) are cel mult un medic asociat (medic.id).
4. Tabela receptioner cu Tabela angajat:
   * Un receptioner (receptioner.id) corespunde unui angajat (angajat.id).
   * Un angajat (angajat.id) are cel mult un receptioner asociat (receptioner.id).
5. Tabela super-administrator cu Tabela utilizator:
   * Un super-administrator (super-administrator.id) corespunde unui utilizator (utilizator.id).
   * Un utilizator (utilizator.id) are cel mult un super-administrator asociat (super-administrator.id).

Relații de Tip Unu-la-Mulți (One-to-Many):

1. Tabela department cu Tabela employee:
   * Un departament (department.dnumber) poate avea mai mulți angajați (employee.dno).
2. Tabela employee cu Tabela dependent:
   * Un angajat (employee.ssn) poate avea mai mulți dependenți (dependent.essn).
3. Tabela department cu Tabela project:
   * Un departament (department.dnumber) poate fi asociat cu mai multe proiecte (project.dnum).
4. Tabela project cu Tabela works\_on:
   * Un proiect (project.pnumber) poate avea mai mulți angajați care lucrează la el (works\_on.pno).
5. Tabela utilizator cu Tabela administrator:
   * Un utilizator (utilizator.id) poate avea mai mulți administratori (administrator.utiliztor\_id).
6. Tabela utilizator cu Tabela angajat:
   * Un utilizator (utilizator.id) poate avea mai mulți angajați (angajat.utiliztor\_id).
7. Tabela utilizator cu Tabela financiar:
   * Un utilizator (utilizator.id) poate avea mai multe înregistrări financiare (financiar.utiliztor\_id).
8. Tabela unitate cu Tabela orar:
   * O unitate (unitate.id) poate avea mai multe programe (orar.unitate\_id).
9. Tabela medic cu Tabela orar:
   * Un medic (medic.id) poate avea mai multe programe (orar.medic\_id).
10. Tabela unitate cu Tabela istoric:
    * O unitate (unitate.id) poate avea mai multe înregistrări istorice (istoric.unitate\_id).
11. Tabela client cu Tabela consultatie:
    * Un client (client.id) poate avea mai multe consultații (consultatie.client\_id).
12. Tabela medic cu Tabela consultatie:
    * Un medic (medic.id) poate avea mai multe consultații (consultatie.medic\_id).
13. Tabela consultatie cu Tabela consultatie\_has\_servicii:
    * O consultație (consultatie.id) poate avea mai multe servicii asociate (consultatie\_has\_servicii.consultatie\_id).
14. Tabela unitate cu Tabela servicii:
    * O unitate (unitate.id) poate oferi mai multe servicii (servicii.unitate\_id).
15. Tabela utilizator cu Tabela resurse\_umane:
    * Un utilizator (utilizator.id) poate fi asociat cu mai multe înregistrări HR (resurse\_umane.utiliztor\_id).

Relații de Tip Mulți-la-Mulți (Many-to-Many):

1. Tabela medic cu Tabela specialitate:
   * Un medic (medic.id) poate avea mai multe specializări (specialitate.id).
   * O specializare (specialitate.id) poate fi asociată cu mai mulți medici (medic.id).

#### 4.Descrirerea bazei de date relationale

 Tabelul unitate:

* Atribute: id (identificator unic al unității), nume (numele unității), adresa (adresa unității), echipament (informații despre echipamentul din unitate), program (programul de funcționare), venituri (informații despre veniturile unității), cheltuieli (informații despre cheltuielile unității).

 Tabelul servicii:

* Atribute: id (identificator unic al serviciului), specialitate (specialitatea serviciului medical), competenta\_necesara (competențele necesare pentru serviciu), pret (prețul serviciului), durata (durata serviciului), unitate\_id (cheie străină care leagă serviciul de o anumită unitate), medic\_id (cheie străină care leagă serviciul de un medic).

 Tabelul utilizator:

* Atribute: id (identificator unic al utilizatorului), CNP (Cod Numeric Personal), nume, prenume, adresa, nrtel (număr de telefon), email, IBAN (numărul de cont bancar), nrcontact (număr de contact), data\_angajarii (data angajării), functie (funcția utilizatorului).

 Tabelul angajat:

* Atribute: id (identificator unic al angajatului), salariu, nr\_ore (numărul de ore lucrate), utilizator\_id (cheie străină care leagă angajatul de un anumit utilizator).

 Tabelul administrator:

* Atribute: id (identificator unic al administratorului), utilizator\_id (cheie străină care leagă administratorul de un anumit utilizator).

 Tabelul super-administrator:

* Atribute: id (identificator unic al super-administratorului), utilizator\_id (cheie străină care leagă super-administratorul de un anumit utilizator).

 Tabelul resurse\_umane:

* Atribute: id (identificator unic al resurselor umane), angajat\_id (cheie străină care leagă resursele umane de un anumit angajat).

 Tabelul financiar:

* Atribute: id (identificator unic al aspectului financiar), angajat\_id (cheie străină care leagă aspectul financiar de un anumit angajat).

 Tabelul receptioner:

* Atribute: id (identificator unic al receptionerului), angajat\_id (cheie străină care leagă receptionerul de un anumit angajat), orar\_id (cheie străină care leagă receptionerul de un anumit orar).

 Tabelul concediu:

* Atribute: id (identificator unic al concediului), data\_inceput (data de început a concediului), data\_sfarsit (data de sfârșit a concediului), angajat\_id (cheie străină care leagă concediul de un anumit angajat).

 Tabelul asistent:

* Atribute: id (identificator unic al asistentului), nr\_ore (numărul de ore lucrate), salariu, tip (tipul de asistent), grad (gradul asistentului), angajat\_id (cheie străină care leagă asistentul de un anumit angajat), orar\_id (cheie străină care leagă asistentul de un anumit orar), orar\_medic\_id (cheie străină care leagă asistentul de un orar specific medicului).

 Tabelul medic:

* Atribute: id (identificator unic al medicului), salariu, nr\_ore (numărul de ore lucrate), specialitatea (specialitatea medicului), grad (gradul medicului), cod\_parafa (codul de parafă al medicului), competente (competențele medicului), titlu\_stiintific (titlul științific al medicului), postul\_didactic (postul didactic al medicului), procent (procentul de implicare), angajat\_id (cheie străină care leagă medicul de un anumit angajat).

 Tabelul client:

* Atribute: id (identificator unic al clientului), nume, prenume, CNP, varsta, nr\_tel (numărul de telefon al clientului).

 Tabelul raport\_medical:

* Atribute: id (identificator unic al raportului medical), data (data raportului medical), simptome, investigatii, diagnostic, recomandari, client\_id (cheie străină care leagă raportul medical de un anumit client), medic\_id (cheie străină care leagă raportul medical de un anumit medic).

 Tabelul orar:

* Atribute: id (identificator unic al orarului), ziuasaptamanii (ziua săptămânii), orainceput (ora de început a programului), orasfarsit (ora de sfârșit a programului), tiporar (tipul de orar), medic\_id (cheie străină care leagă orarul de un anumit medic), unitate\_id (cheie străină care leagă orarul de o anum

#### 4.Argumentarea nivelului de normalizare

 Tabela concediu:

* Este în 3NF, având o cheie primară (id) și o cheie străină către tabela angajat.
* Asigură gestionarea corespunzătoare a informațiilor despre concedii, fără redundanțe.

 Tabela servicii:

* Este în 3NF, cu o cheie primară bine definită (id) și o cheie străină corespunzătoare spre tabela unitate.
* Oferă o structură clară pentru informațiile despre servicii, evitând dependențe inutile.

 Tabela consultatie\_has\_servicii:

* Este în 3NF, având o cheie primară compusă și chei străine către tabelele consultatie și servicii.
* Menține coerența datelor despre serviciile asociate fiecărei consultații.

 Tabela financiar:

* Este în 3NF, având o cheie primară (id) și o cheie străină corespunzătoare spre tabela utilizator.
* Asigură o gestionare eficientă a informațiilor financiare în legătură cu utilizatorii.

 Tabela istoric:

* Este în 3NF, având o cheie primară compusă și o cheie străină către tabela unitate.
* Oferă un istoric detaliat al veniturilor și cheltuielilor asociate fiecărei unități.

 Tabela raport medical:

* Este în 3NF, având o cheie primară (id) și chei străine corespunzătoare spre tabelele client și medic.
* Asigură o structură clară pentru informațiile medicale și recomandările asociate fiecărui raport.

 Tabela receptioner:

* Este în 3NF, având o cheie primară (id) și chei străine către tabelele angajat și orar.
* Menține integritatea referențială pentru informațiile despre receptioneri și orarele lor.

 Tabela resurse\_umane:

* Este în 3NF, având o cheie primară (id) și o cheie străină corespunzătoare spre tabela utilizator.
* Asigură o gestionare eficientă a resurselor umane în cadrul sistemului.

 Tabela specialitate:

* Este în 3NF, având o cheie primară bine definită (id).
* Oferă o reprezentare clară a specialităților medicale fără redundanțe.

 Tabela super-administrator:

* Este în 3NF, având o cheie primară (id) și o cheie străină corespunzătoare spre tabela utilizator.
* Menține coerența datelor pentru super-administratori

În concluzie, baza de date, a fost proiectată cu atenție pentru a respecta principiile normalizării până la nivelul 3NF (Formă Normală a Treia). Aceasta înseamnă că datele sunt organizate în tabele astfel încât să minimizeze redundanța și să asigure o structură coerentă și eficientă.

#### 6.Codul SQL

 CalculSumaValoriBonFiscalPentruMedic(id\_medic\_param INT):

* Descriere: Această procedură calculează suma totală a valorilor bonurilor fiscale pentru un medic specific.
* Parametri:
  + id\_medic\_param - ID-ul medicului pentru care se calculează suma.
* Funcționalitate:
  + Se deschide un cursor pentru a itera prin consultările medicului specific.
  + Se folosește o buclă WHILE pentru a parcurge consultările și a aduna valorile bonurilor fiscale pentru fiecare.
  + Suma finală este returnată prin SELECT la închiderea cursorului.

 datepersonale(cnp\_param VARCHAR(13)):

* Descriere: Această procedură returnează informațiile personale ale unui utilizator pe baza CNP-ului său.
* Parametri:
  + cnp\_param - CNP-ul utilizatorului căruia i se caută informațiile.
* Funcționalitate:
  + Se realizează o interogare pentru a selecta informațiile personale ale utilizatorului cu CNP-ul dat.

 Ex1(in\_nume VARCHAR(45), in\_prenume VARCHAR(45), in\_functie VARCHAR(45)):

* Descriere: Această procedură folosește funcția GetAngajatId pentru a obține ID-ul unui angajat și apoi apelează procedura idj pentru a afișa detaliile despre angajat.
* Parametri:
  + in\_nume, in\_prenume, in\_functie - Informații despre angajat pentru a obține ID-ul.
* Funcționalitate:
  + Se obține ID-ul angajatului apelând GetAngajatId.
  + Se apelează idj cu ID-ul obținut pentru a afișa detaliile angajatului.

 Ex2.AfisareInformatii():

* Descriere: Această procedură afișează toate informațiile din tabelele orar și concediu.
* Parametri: Nu are parametri.
* Funcționalitate:
  + Se realizează două interogări separate pentru a selecta toate înregistrările din tabelele menționate.

 Ex2.InserareConcediu(id\_resurse\_inspector INT, in\_data\_inceput DATE, in\_data\_sfarsit DATE, in\_angajat\_id INT):

* Descriere: Această procedură inserează o nouă înregistrare în tabela concediu dacă ID-ul resurselor umane/inspector este valid.
* Parametri:
  + id\_resurse\_inspector - ID-ul resurselor umane/inspector.
  + in\_data\_inceput, in\_data\_sfarsit - Perioada de început și sfârșit a concediului.
  + in\_angajat\_id - ID-ul angajatului pentru care se adaugă concediul.
* Funcționalitate:
  + Se verifică validitatea ID-ului resurselor umane/inspector folosind funcția new\_function.
  + Dacă este valid, se determină un nou ID și se inserează o nouă înregistrare în tabela concediu.

 Ex2.InserareOrar(id\_resurse\_inspector INT, ...):

* Descriere: Această procedură inserează o nouă înregistrare în tabela orar dacă ID-ul resurselor umane/inspector este valid.
* Parametri: Similar cu Ex2.InserareConcediu, cu informații specifice pentru orar.
* Funcționalitate:
  + Se verifică validitatea ID-ului resurselor umane/inspector folosind funcția new\_function.
  + Dacă este valid, se determină un nou ID și se inserează o nouă înregistrare în tabela orar.

 Ex2.StergereConcediu(id\_inspector\_resurse\_umane INT, in\_id\_concediu INT):

* Descriere: Această procedură șterge o înregistrare din tabela concediu dacă ID-ul resurselor umane/inspector este valid.
* Parametri:
  + id\_inspector\_resurse\_umane - ID-ul resurselor umane/inspector.
  + in\_id\_concediu - ID-ul concediului care trebuie șters.
* Funcționalitate:
  + Se verifică validitatea ID-ului resurselor umane/inspector folosind funcția new\_function.
  + Dacă este valid, se șterge înregistrarea corespunzătoare din tabela concediu.

 Ex2.StergereOrar(id\_inspector\_resurse\_umane INT, in\_id\_orar INT):

* Descriere: Această procedură șterge o înregistrare din tabela orar dacă ID-ul resurselor umane/inspector este valid.
* Parametri: Similar cu Ex2.StergereConcediu, cu informații specifice pentru orar.
* Funcționalitate:
  + Se verifică validitatea ID-ului resurselor umane/inspector folosind funcția new\_function.
  + Dacă este valid, se șterge înregistrarea corespunzătoare din tabela orar.

 Ex3(cnp\_ut varchar(60)):

* Descriere: Această procedură afișează informațiile despre concediul unui utilizator dat CNP-ul său.
* Parametri:
  + cnp\_ut - CNP-ul utilizatorului.
* Funcționalitate:
  + Se obține ID-ul utilizatorului și se determină ID-ul angajatului asociat.
  + Se apelează GetOrarForAngajat pentru a afișa informațiile despre orarul angajatului și se selectează informațiile despre concediu.

 GetConcediuDataForUtilizator(in\_utilizator\_id INT):

* Descriere: Această procedură obține informațiile despre concediu pentru un utilizator dat ID-ul său.
* Parametri:
  + in\_utilizator\_id - ID-ul utilizatorului.
* Funcționalitate:
  + Se obține ID-ul angajatului asociat utilizatorului și se apelează GetOrarForAngajat pentru a afișa informațiile despre orarul angajatului și se selectează informațiile despre concediu.

 GetOrarForAngajat(in\_angajat\_id INT):

* Descriere: Această procedură obține informațiile despre orarul pentru un angajat dat ID-ul său.
* Parametri:
  + in\_angajat\_id - ID-ul angajatului.
* Funcționalitate:
  + Se determină ID-ul orarului pentru angajat și se selectează informațiile despre orar.

 idj(IN employeeId INT):

* Descriere: Această procedură afișează detaliile despre un angajat dat ID-ul său.
* Parametri:
  + employeeId - ID-ul angajatului.
* Funcționalitate:
  + Se determină tipul de angajat (medic, asistent, receptioner, financiar, resurse umane) folosind ID-ul angajatului și se afișează detaliile respective.

 modificare\_serviciu(IN p\_id INT, IN p\_nou\_pret INT, IN p\_noua\_durata INT):

* Descriere: Această procedură actualizează prețul și durata unui serviciu în funcție de ID-ul său.
* Parametri:
  + p\_id - ID-ul serviciului care trebuie modificat.
  + p\_nou\_pret - Noul preț al serviciului.
  + p\_noua\_durata - Noua durată a serviciului.
* Funcționalitate:
  + Se realizează o actualizare a prețului și duratei serviciului în tabela servicii

calcul\_profit\_per\_medic

* + Scop: Calculează și returnează un rezumat text al informațiilor de profit pentru fiecare medic.
  + Detalii:
    - Folosește un cursor pentru a itera prin fiecare medic și calculează venitul total și cheltuielile unității asociate fiecărui medic.

InserareRaportMedical

* + Scop: Inserează rapoarte medicale într-o tabelă.
  + Detalii:
    - Verifică existența unui client cu numele și prenumele specificate și inserează raportul medical în caz afirmativ.
    - Dacă clientul nu este găsit, emite o excepție cu mesajul "Clientul nu a fost găsit".

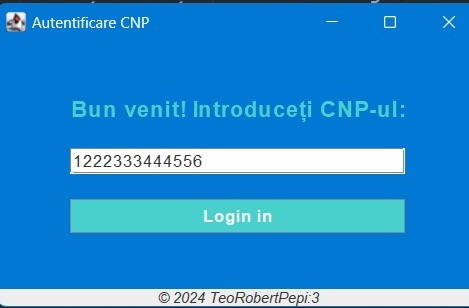
lunar\_event

* + Scop: Declanșează un eveniment lunar care rulează o procedură stocată la fiecare lună.
  + Detalii:
    - Rulează un script Bash specificat în calea @script\_path la fiecare lună.
    - Pentru a rula scriptul, se folosește mysql.ureport\_cmdline('bash ' + @script\_path).

1. Funcția: FindAngajatIdByUtilizatorId
   * Descriere: Această funcție caută și returnează ID-ul unui angajat bazat pe ID-ul utilizatorului asociat.
   * Parametri:
     + in\_utilizator\_id - ID-ul utilizatorului.
   * Funcționalitate:
     + Caută în tabela angajat și returnează ID-ul angajatului asociat utilizatorului dat. Dacă nu găsește nicio potrivire, returnează NULL.
2. Funcția: gaseste\_id\_orar\_angajat
   * Descriere: Această funcție caută și returnează ID-ul orarului asociat unui angajat cu un anumit ID.
   * Parametri:
     + angajat\_id\_param - ID-ul angajatului.
   * Funcționalitate:
     + Determină ID-ul medicului asociat angajatului și apoi găsește ID-ul orarului corespunzător în tabela orar.
3. Funcția: gaseste\_orar\_id
   * Descriere: Această funcție caută și returnează ID-ul orarului pentru un angajat în funcție de funcția acestuia (asistent, receptioner, financiar sau resurse umane).
   * Parametri:
     + angajat\_id\_param - ID-ul angajatului.
   * Funcționalitate:
     + Caută în tabelele asistent, receptioner, financiar și resurse\_umane pentru a găsi ID-ul orarului asociat angajatului dat. Returnează ID-ul găsit sau NULL dacă nu există nicio potrivire.
4. Funcția: GetAngajatId
   * Descriere: Această funcție caută și returnează ID-ul unui angajat în funcție de nume, prenume și funcție.
   * Parametri:
     + in\_name - Numele angajatului.
     + in\_surname - Prenumele angajatului.
     + in\_function - Funcția angajatului.
   * Funcționalitate:
     + Caută în tabela utilizator pentru a obține ID-ul utilizatorului în funcție de nume, prenume și funcție.
     + Caută în tabela angajat pentru a obține ID-ul angajatului asociat utilizatorului găsit. Returnează ID-ul găsit sau -1 dacă nu există nicio potrivire.
5. Funcția: getValoareBonFiscalByIdMedic
   * Descriere: Această funcție caută și returnează valoarea unui bon fiscal pentru o consultație specifică asociată unui medic.
   * Parametri:
     + id\_medic - ID-ul medicului.
     + id\_consultatie - ID-ul consultatiei.
   * Funcționalitate:
     + Verifică dacă există o consultatie asociata medicului cu ID-ul specificat și returnează valoarea bonului fiscal asociat acelei consultatii. Dacă nu există nicio potrivire, returnează 0.
6. Funcția: new\_function
   * Descriere: Această funcție verifică existența unui angajat în tabelele financiar sau resurse\_umane în funcție de ID-ul utilizatorului.
   * Parametri:
     + userId - ID-ul utilizatorului.
   * Funcționalitate:
     + Apelează funcția FindAngajatIdByUtilizatorId pentru a obține ID-ul angajatului asociat utilizatorului.
     + Verifică existența angajatului în tabela financiar și, dacă nu există, verifică în tabela resurse\_umane.
     + Returnează TRUE dacă angajatul este găsit în oricare dintre cele două tabele și FALSE în caz contrar.

# 7.Functionalitatea aplicatiei

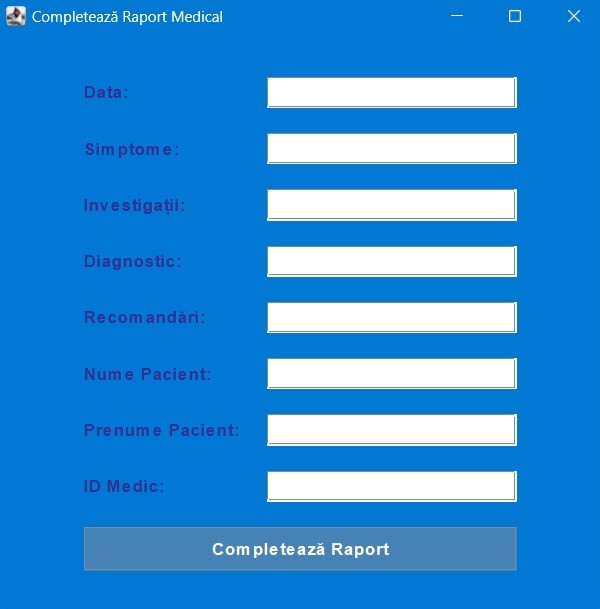
1.Modalitatea de logare

**

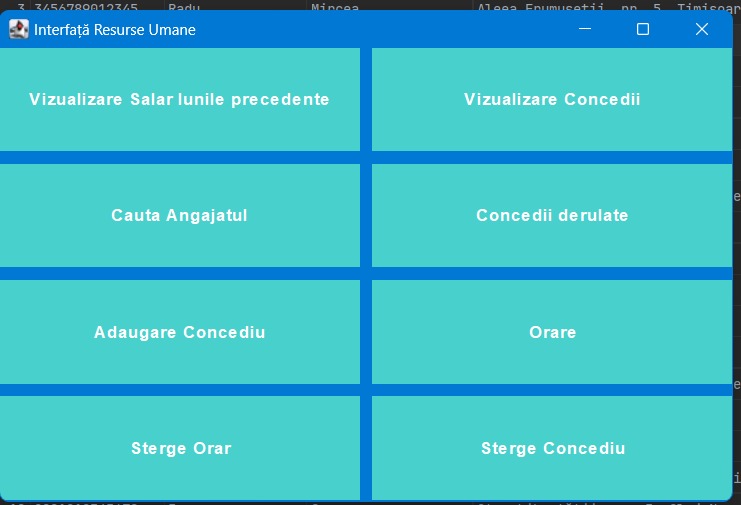
2.Meniu pentru medic

**

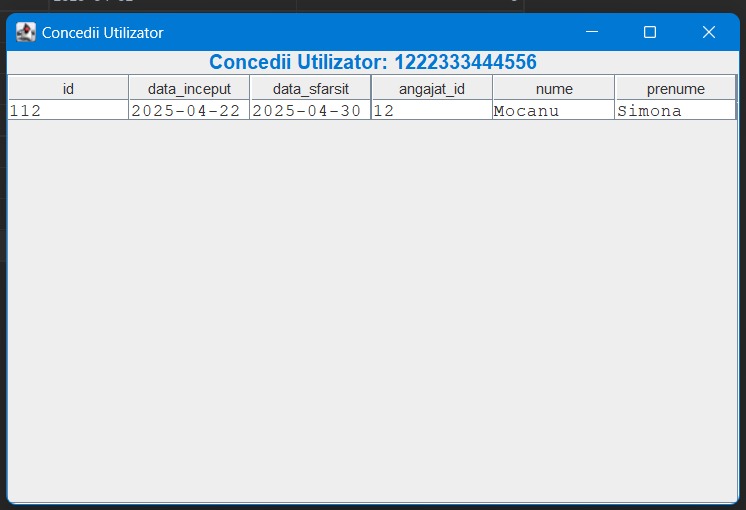
3.Adaugare raport medic

**

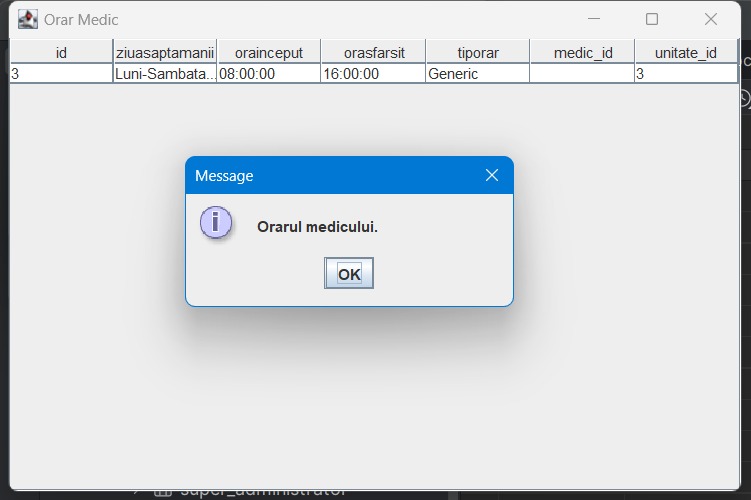
4.Meniu inspector resurse umane

**

5.Adaugare concediu pentru angajati

**

6.Optiune de vizualizare a orarului

**

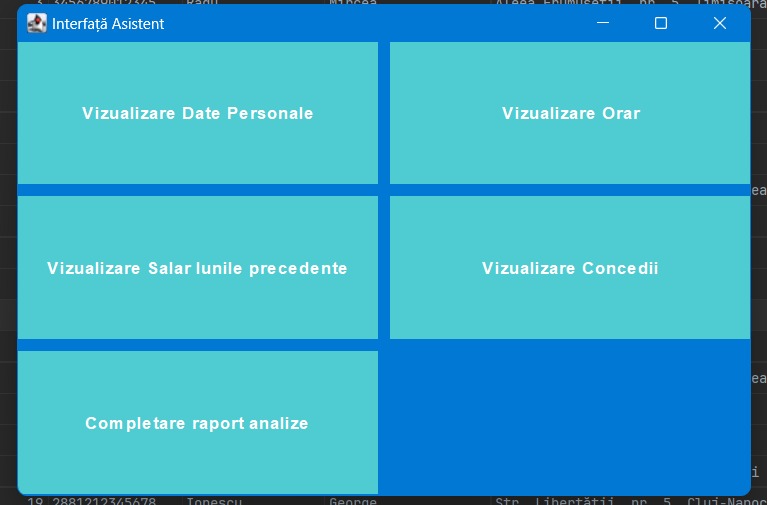
7.Interfata Expert financiar contabil



8.Interfata Receptioner



9.Interfata asistent

**

# 8.Concluzii

Proiectul reprezintă o aplicație de gestionare a unui lanț de policlinici, oferind funcționalități extinse pentru administrare, resurse umane, servicii medicale și rapoarte. Utilizatorii, în funcție de rolul lor, au acces la diferite module și acțiuni. De la administrarea utilizatorilor și a informațiilor personale ale angajaților la gestionarea orarelor, concediilor și salariilor, până la programarea pacienților, emiterea de rapoarte medicale și calcularea profitului, proiectul acoperă o gamă largă de nevoi.

Este notabilă inclusiv secțiunea de personalizare a serviciilor medicale și calculul salariilor medicilor în funcție de consultări, evidențiind atenția acordată detaliilor și adaptabilității sistemului. De asemenea, există un echilibru între accesul la informații, astfel încât fiecare tip de angajat să aibă acces doar la datele relevante pentru rolul său.