



LANT DE POLICLINICI

Proiect la Baze de Date



SĂVIANU BOGDAN
PĂCURAR IRINA
UTCN, CTI, SERIA A, GRUPA 30223
ÎNDRUMĂTOR: IVAN COSMINA DANIELA
- ianuarie 2024 -

Cuprins

1. Tema Proiectului	3
2. Proiectarea conceptuală a bazei de date	5
3. Descrierea bazei de date relaționale	6
4. Argumentarea nivelului de normalizare	7
5. Codul în SQL	8
6. Funcționalitatea aplicației	12
A. ConnectionPackage	12
1. Metode pe proceduri	12
2. Metode pe interogări	12
B. GUIPackage	13
C. ModelsPackage	23
7. Manual de utilizare	24
8. Concluzii și dezvoltări ulterioare	25
9. Bibliografie	25

1. Tema Proiectului

Se dorește crearea unui sistem informatic pentru gestionarea activităților într-un lanț de policlinici, folosind baza de date MySQL și interfața grafică. Funcționalitățile includ gestionarea angajaților, serviciilor financiar-contabile și a operațiilor curente din policlinică, cum ar fi gestionarea pacienților programați, completarea rapoartelor medicale și emiterea bonurilor fiscale.

Utilizatorii se autentifică în sistem și pot aparține departamentelor de resurse umane, financiar-contabil sau medical. Fiecare utilizator are informații personale precum CNP, nume, prenume, adresă, număr de telefon, email, cont IBAN, număr de contract, dată angajare și funcția deținută în cadrul lanțului de policlinici. Utilizatorii pot vizualiza datele personale, dar nu le pot modifica. Programul oferă funcționalitatea de deautentificare pentru a permite accesul altor utilizatori fără a reporni aplicația.

Administratorii pot adăuga, modifica și șterge informații în baza de date referitoare la utilizatori. Există și un rol de super-administrator care poate opera inclusiv asupra administratorilor.

Angajaților li se rețin salariul negociat și numărul de ore lunare. Funcțiile disponibile în lanțul de policlinici includ inspector resurse umane, expert financiar-contabil, recepționar, asistent medical și medic. Asistenților medicali li se adaugă tipul și gradul, iar medicilor li se rețin specialitatea, gradul, codul de parafă, competențele pentru proceduri speciale, titlul științific și postul didactic. Fiecare medic are un procent negociat din veniturile generate de serviciile medicale.

Lanțul de policlinici furnizează pacienților diverse servicii medicale, unele fiind disponibile pentru toate specialitățile, iar altele specifice fiecărei specialități. Detaliile pentru un serviciu medical includ specialitatea, competențele necesare, prețul și durata acestuia.

Modulele includ gestionarea resurselor umane (programul de lucru și concediile angajaților), operațiile financiar-contabile (profitul operațional determinat între venituri și cheltuieli) și gestionarea activităților operaționale (programarea pacienților, emiterea bonurilor fiscale, completarea rapoartelor medicale).

Modulul pentru gestiunea resurselor umane permite inspectorilor să caute angajați după parametri precum nume, prenume, și funcție. Se specifică un orar de lucru și perioada de concediu pentru fiecare angajat. Inspectorii și experții financiar-contabili pot consulta orarul săptămânal și informațiile despre concedii. Recepționerii, asistenții medicali și medicii pot vizualiza doar informațiile proprii, fără posibilitatea de a le modifica.

Modulul de operații financiar-contabile oferă experților informații despre profitul lanțului de policlinici, calculat ca diferență între venituri și cheltuieli. Salariile sunt ponderate în funcție de numărul de ore lucrate și sunt disponibile rapoarte privind profitul pentru fiecare medic, locație sau specialitate. Orice angajat poate vizualiza salariile obținute anterior, iar medicii pot consulta profitul generat de serviciile medicale furnizate.

Modulul pentru gestiunea activităților operaționale permite recepționarilor să programeze pacienți și să înregistreze pacienți prezenți în clinică. Asistenții medicali completează rapoarte pentru analize medicale, iar medicii pot vizualiza lista programărilor zilnice și istoricul pacienților. Ei completează rapoarte medicale obligatorii cu informații administrative și secțiuni medicale. Rapoartele sunt parafate și nu pot fi modificate ulterior.

Toate modulele sunt integrate într-un singur sistem informatic cu meniuri personalizate pentru fiecare utilizator, având permisiuni limitate în funcție de drepturile lor.

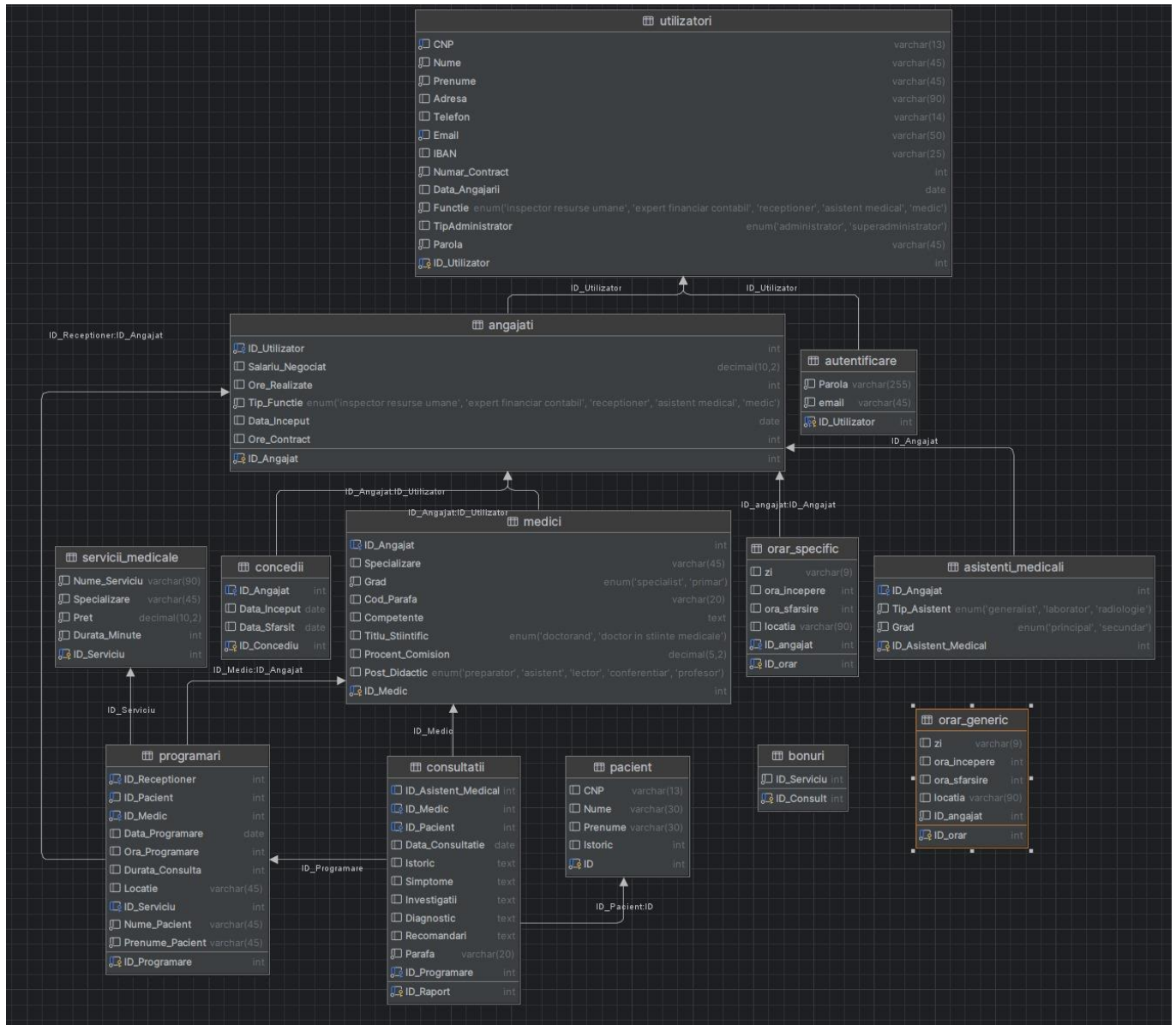
Sistemul trebuie să gestioneze orarele angajaților, având un tip generic pentru toți și unul specific pentru personalul medical. Asigurarea că orarul unui medic se potrivește cu programul locației în care oferă consultații este esențială, evitându-se suprapunerea programărilor.

Medicii pot oferi consultații în mai multe locații, cu condiția ca orarele acestora să fie disjuncte. Calculul profitului se realizează ținând cont doar de veniturile și cheltuielile pentru specialitatea respectivă, fără a impune restricții stricte privind zilele de concediu.

Salariile sunt calculate doar pentru zilele în care angajatul are definit un orar, fără remunerare în perioadele de concediu. Tariful consultației este determinat automat în funcție de gradul medicului. Situația în care un medic are mai multe specialități este gestionată prin posibilitatea variării a gradului pentru fiecare specialitate în parte.

În ceea ce privește programările, acestea sunt restricționate la serviciile unei singure specialități, iar programele de lucru diferențiate pe specialități nu sunt permise. Astfel, sistemul asigură o gestionare eficientă a orarelor, consultațiilor și salariilor, fără constrângeri stricte privind concediile și cu o administrare clară a diverselor specialități medicale.

2. Proiectarea conceptuală a bazei de date



3. Descrierea bazei de date relaționale

1. **Tabelul utilizatori:** Stochează informațiile despre utilizatori, inclusiv angajații policlinicii și administratorii. Atributele includ CNP, nume, prenume, adresa, telefon, email, IBAN, număr de contract, data angajării, funcția (rolul în organizație), tipul de administrator și parola pentru autentificare.
2. **Tabelul angajati:** Conține detalii despre angajații policlinicii, cu informații precum salariu negociat, ore realizate, tipul de funcție, data începerii contractului și orele contractuale. Se leagă de tabelul **utilizatori** prin **ID_Utilizator**.
3. **Tabelul asistenti_medicali:** Păstrează informații specifice asistenților medicali, inclusiv tipul de asistent (generalist, laborator, radiologie) și gradul (principal, secundar). Se leagă de tabelul **angajati** prin **ID_Angajat**.
4. **Tabelul autentificare:** Gestionează informațiile de autentificare ale utilizatorilor, inclusiv ID-ul utilizatorului, parola și adresa de email. Este conectat la tabelul **utilizatori** prin **ID_Utilizator**.
5. **Tabelul concedii:** Stocază informații despre concediile angajaților, inclusiv data de început și sfârșit. Este conectat la tabelul **angajati** prin **ID_Angajat**.
6. **Tabelul medici:** Conține detaliile despre medici, cum ar fi specializarea, gradul, codul parafă, competențele, titlul științific, procentul de comision și postul didactic. Se leagă de tabelul **angajati** prin **ID_Angajat**.
7. **Tabelul servicii_medicale:** Păstrează informații despre serviciile medicale oferite, precum numele, specializarea, prețul și durata. Este conectat la tabelul **medici** prin **ID_Medic**.
8. **Tabelul programari:** Gestionează programările pentru consultații, având informații precum ID-ul receptionerului, pacientului, medicului, data programării, ora, durata, locația și ID-ul serviciului medical. Este conectat la mai multe tabele, inclusiv **angajati**, **medici** și **servicii_medicale**.
9. **Tabelul pacient:** Conține detalii despre pacienți, inclusiv CNP, nume, prenume și istoric medical.
10. **Tabelul consultatii:** Stochează rapoartele de consultare, având informații despre asistentul medical, medic, pacient, data consultării, simptome, investigații, diagnostic, recomandări și parafă. Este conectat la multiple tabele, inclusiv **asistenti_medicali**, **medici**, **utilizatori** și **programari**.
11. **Tabelul orar_generic:** Definește orarul generic de lucru pentru angajați, precum zilele, orele de început și sfârșit și locația. Este conectat la tabelul **angajati**.
12. **Tabelul orar_specific:** Conține informații despre orarul specific al angajaților, inclusiv zilele, orele și locațiile. Este conectat la tabelul **angajati**.

4. Argumentarea nivelului de normalizare

- **Normalizarea:**
Baza de date pare respectă până la Forma Normală a Treia (3NF). În mod evident, sunt respectate primele două forme normale, deoarece există chei primare în fiecare tabel, și toate coloanele depind funcțional de cheile primare.
- **Relații:** Relațiile dintre asigură integritatea referențială și permite interogări complexe.
- **Eliminarea redundanței de date:** Tabelele sunt structurate în așa fel încât să evite redundanța de date. De exemplu, informațiile despre angajați sunt separate în două tabele distincte: **utilizatori** și **angajati**, și sunt legate printr-un cheie străină (**ID_Utilizator**).
- **Utilizarea cheilor străine:** Cheile străine sunt folosite corespunzător pentru a stabili relații între tabele. Aceasta permite asigurarea integrității referențiale între diferitele părți ale bazei de date.
- **Evitarea duplicării datelor:** Datele, precum CNP-ul și email-ul utilizatorilor, sunt definite ca chei unice, prevenind astfel duplicarea acestora în tabel.
- **Folosirea tipurilor de date adecvate:** Tipurile de date pentru diferite câmpuri sunt alese corespunzător. De exemplu, se folosește **DATE** pentru stocarea datelor de angajare sau programare, iar **DECIMAL** pentru stocarea valorilor financiare.
- **Folosirea de tipuri de date enum:** Utilizarea tipului de date **ENUM** pentru câmpuri precum **Funcție** sau **TipAdministrator** poate contribui la asigurarea validării și a consistenței datelor.
- **Utilizarea indecșilor unici:** Indecși unici sunt folosiți pentru a asigura că nu există duplicări ale cheilor primare sau ale altor valori critice în baza de date.
- **Utilizarea cheilor primare:** Fiecare tabelă are o cheie primară unică, ceea ce este esențial pentru identificarea unică a înregistrărilor din tabel.
- **Folosirea constrângerilor pentru acțiuni în cazul ștergerii sau actualizării:** Constrângerile **ON DELETE CASCADE** și **ON UPDATE CASCADE** asigură coerența datelor între tabele în cazul ștergerii sau actualizării înregistrărilor.

5. Codul în SQL

AdaugaOrarPentruAngajat:

Adaugă un nou orar pentru un angajat în cadrul unei policlinici. Primește informații precum ID-ul angajatului, ziua săptămânii, orele de începere și sfârșire ale programului, locația și tipul de orar (generic sau specific). Inserează datele în tabelul corespunzător (orar_generic sau orar_specific) în funcție de tipul de orar ales și returnează ID-ul orarului adăugat.

AdaugaPacient:

Adaugă un nou pacient în sistemul policlinicii, primește informații precum numele și prenumele pacientului și inserează aceste date în tabelul "pacient" al bazei de date.

AdaugaProgramare:

Adaugă o nouă programare în sistemul policlinicii, primind informații precum ID-ul receptionerului, numele și prenumele pacientului, ID-ul medicului, data și ora programării, locația și ID-ul serviciului. Inserează un nou pacient în tabelul "pacient" al bazei de date și obține ID-ul pacientului. Totodată, inserează informațiile programării, inclusiv ID-ul pacientului obținut anterior, în tabelul "programări".

AdaugaRaportMedical

Această procedură stocată este creată pentru a adăuga un raport medical în baza de date a policlinicii. Parametrii de intrare includ informații despre pacient, asistentul medical, programarea asociată și detalii despre consultație. Procedura actualizează și CNP-ul pacientului.

AdaugaUtilizator:

Adaugă un nou utilizator în sistemul policlinicii, primeind informații precum CNP, nume, prenume, adresă, telefon, email, parolă, IBAN, număr de contract, data angajării, funcția (dintr-o listă predefinită), tipul administratorului, salariu și orele contractului. Inserează informațiile utilizatorului în tabelul "utilizatori" și obține ID-ul utilizatorului adăugat. Inserează informațiile angajatului în tabelul "angajati" și utilizează ID-ul utilizatorului pentru a stabili legătura între tabele. Inserează informațiile de autentificare (parolă și email) în tabelul "autentificare". Dacă funcția este "Asistent Medical", adaugă o intrare în tabelul "asistenti_medicali". Dacă funcția este "Medic", adaugă o intrare în tabelul "medici".

AdaugaUtilizatorSiAsistentMedical:

Adaugă un nou utilizator cu funcția de "Asistent Medical" în sistemul policlinicii, primind informații precum CNP, nume, prenume, adresă, telefon, email, parolă, IBAN, număr de contract, data angajării, tip de administrator (dacă este sau nu administrator), salariu și orele contractului.

Utilizează procedura "AdaugaUtilizator" pentru a adăuga informațiile generale ale utilizatorului și inserează informațiile specifice asistentului medical în tabelul "asistenti_medicali", inclusiv tipul asistentului (generalist, laborator sau radiologie) și gradul (principal sau secundar).

AdaugaUtilizatorSiMedic:

Adaugă un nou utilizator cu funcția de "Medic" în sistemul policlinicii, primind informații precum CNP, nume, prenume, adresă, telefon, email, parolă, IBAN, număr de contract, data angajării, tip de administrator (dacă este sau nu administrator), salariu și orele contractului. Utilizează procedura "AdaugaUtilizator" pentru a adăuga informațiile generale ale utilizatorului și atribuie un ID utilizator, apoi utilizează procedura "UpdateMedic" pentru a adăuga detaliile specifice medicului în tabelul "medici", inclusiv specializarea, gradul, codul parafă, competențele, titlul științific, procentul de comision, postul didactic. Nu conține o inserare în tabelul "autentificare" în comentariu, ceea ce înseamnă că poate fi necesar să fie adăugată ulterior, în funcție de cerințele aplicației.

CalculeazaProfitMedic:

Calculează profitul unui medic pentru un anumit interval de timp (lună și an).

Primește ID-ul medicului (p_medicID), luna (p_month) și anul (p_year) pentru care se dorește calculul profitului. Calculează profitul utilizând formula:

Profit = Suma totală a prețurilor serviciilor - (Salariul medicului * (Ore realizate de medic / Ore contractuale ale medicului)).

Se bazează pe tabelele consultatii, programari, servicii_medicale, și angajati pentru a obține datele necesare, rezultatul fiind stocat în parametrul de ieșire p_profit.

CalculeazaProfitPoliclinica:

Calculează profitul total al policlinicii, suma totală a salariilor angajaților (p_TotalSalaries) prin selectarea sumei salariilor negociate din tabelul angajati, suma totală a prețurilor serviciilor medicale (p_TotalServicePrices) prin combinarea datelor din tabelele consultatii, programari, și servicii_medicale. Totodată, calculează profitul policlinicii (p_Profit) prin scăderea sumei totale a salariilor din suma totală a prețurilor serviciilor. Rezultatele sunt stocate în parametrii de ieșire p_TotalSalaries, p_TotalServicePrices, și p_Profit.

GetLastMedic:

Această procedură returnează ultimul medic adăugat în tabelul medici. Variabila lastMedicId este utilizată pentru a stoca ID-ul ultimului medic. Se face o interogare pentru a obține ID-ul maxim al unui medic din tabelul medici și se stochează în variabila lastMedicId. Se selectează întreaga înregistrare a medicului din tabelul medici folosind ID-ul ultimului medic și se returnează rezultatul.

GetUserByIDAndPassword:

Această procedură are rolul de a autentifica un utilizator pe baza unui ID și a unei parole. Variabila NumRows este utilizată pentru a stoca numărul de rânduri găsite în tabelul autentificare care corespund ID-ului și parolei specificate. Se realizează o interogare pentru a număra câte rânduri îndeplinesc condițiile specificate în clauza WHERE în tabelul autentificare. Se verifică dacă numărul de rânduri (NumRows) este mai mare decât 0 (adică există o potrivire a ID-ului și parolei). În caz afirmativ, se selectează informațiile utilizatorului (căutând după ID-ul utilizatorului) din tabelul utilizatori și se returnează rezultatul. În caz contrar, se returnează un set de valori nule.

StergeUtilizatorSiRelatii:

Această procedură are rolul de a șterge un utilizator și toate relațiile asociate cu el din baza de date. Se utilizează parametrul p_ID_Utilizator pentru a specifica ID-ul utilizatorului care trebuie șters. Se șterg în ordine următoarele relații asociate utilizatorului:

- Toate consultațiile asociate pacientului din tabela consultatii.
- Toate programările asociate pacientului din tabela programari.
- Înregistrarea din tabela asistenti_medicali asociată angajatului (dacă este asistent medical).
- Înregistrarea din tabela medici asociată angajatului (dacă este medic).
- Înregistrarea din tabela angajati asociată utilizatorului.
- Înregistrarea din tabela autentificare asociată utilizatorului.
- Înregistrarea din tabela utilizatori asociată utilizatorului.

Prin această procedură se asigură ștergerea completă a datelor asociate unui utilizator în toate tabelele relevante.

UpdateMedic:

Această procedură are rolul de a actualiza informațiile asociate unui medic în baza de date. Parametrul p_ID_Medic reprezintă ID-ul unic al medicului care trebuie actualizat. Alți parametri, precum p_Specializare, p_Grad, p_Cod_Parafa, p_Competente, p_Titlu_Stiintific, p_Procent_Comision, p_Post_Didactic, reprezintă noile valori pe care se intenționează a fi actualizate în înregistrarea medicului cu ID-ul specificat. Se realizează un UPDATE asupra tabelului medici, actualizând informațiile cu cele furnizate prin parametri. Această procedură este utilă pentru menținerea și modificarea detaliilor despre medici în baza de date.

SalarMedic:

Această procedură are rolul de a calcula salariul total al unui medic pe baza prețurilor serviciilor medicale pe care le-a oferit în cadrul programărilor. Parametrul medicID reprezintă ID-ul unic al medicului pentru care se calculează salariul. Se utilizează un cursor (cur) pentru a itera prin identificatorii unici ai serviciilor medicale asociate programărilor medicului specificat.

Se utilizează un handler pentru a gestiona situația în care nu mai sunt găsite înregistrări (NOT FOUND) în cursor, setând done la TRUE. Într-o buclă, se realizează FETCH pentru a obține prețul

fiecărui serviciu medical asociat programărilor, adăugându-le la totalAmount. După ce toate înregistrările au fost procesate, cursorul este închis. Este returnat rezultatul calculului sub numele de TotalAmount. Procedura oferă o modalitate de calcul a salariului medicului în funcție de serviciile medicale furnizate în cadrul programărilor.

6. Funcționalitatea aplicației

A. ConnectionPackage

1. Metode pe proceduri

- adaugaOrarAngajat apelează procedura AdaugăOrarPentruAngajat.
- getLastMedicId apelează procedura GetLastMedic
- adaugaUtilizatorSiMedic apelează procedura AdaugaUtilizatorSiMedic
- adaugaUtilizatorSiAsistentMedical apelează procedura AdaugaUtilizatorSiAsistentMedical
- adaugaUtilizator apelează procedura AdaugaUtilizator
- getUser apelează procedura GetUserByIDAndPassword
- adaugaConsultatie apelează procedura AdaugaRaportMedical
- calculateProfit apelează procedura CalculeazaprofitPoliclinica
- calculateMedicProfit apelează procedura CalculeazaProfitMedic
- adaugaProgramare apelează procedura AdaugaProgramare
- calculateSalaryForMedic apelează procedura SalarMedic

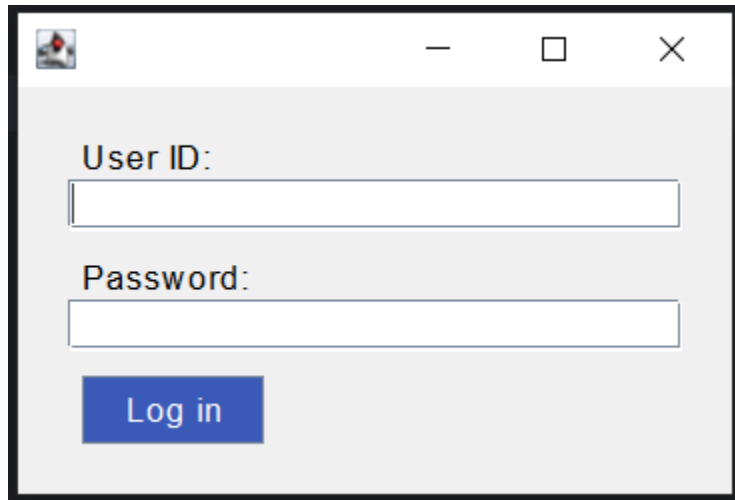
2. Metode pe interogări

- stergeAngajat execută un query care șterge angajatul respectiv și toate intrările din alte tabele, unde acesta face parte din cheia străină.
- getAllMedics executează un query care preia lista cu toți medicii
- getMediciByServiciu apelează metoda getAllMedics pentru a prelua lista cu toți medicii și a-i selecta pe cei care efectuează acel serviciu.
- getAllSpecializari executa un query care creează o listă cu toate specializările
- getAllAngajati execută un query care preia lista cu toți angajații
- searchAngajat apelează metoda getAllAngajati pentru a prelua lista și a căuta în ea.
- verificaDisponibilitateMedic execută un query care verifică dacă medicul este de gardă atunci când se dorește o programare
- getRomanianDayName execută un query care returnează zilele în română după cele în engleză pentru orar.
- isMedicOnVacation execută un query care determină când un medic e în concediu
- calculateSalariesForMonth execută un query care calculează salariul pe lună pentru fiecare angajat
- calculateSalaries execută un apelează metoda calculateSalariesForMonth și face tabel pe ultimele 6 luni pentru orice angajat
- getMonthName execută un query care extrage lunile anului.

B. GUIPackage

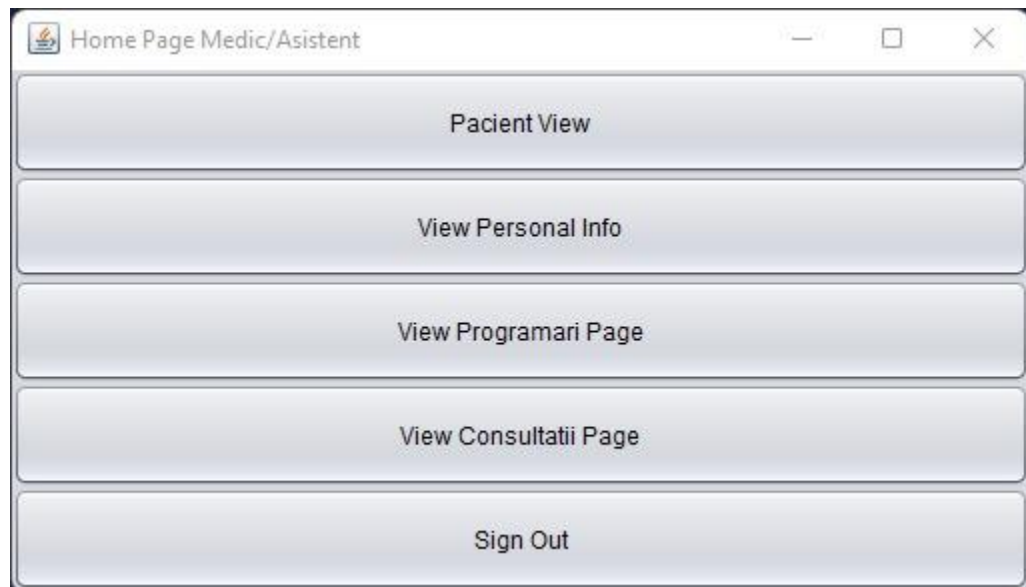
Ferestrele ce urmeaza sa fie afisate in cadrul aplicatiei sunt:

- Logare

A screenshot of a login window. It has a standard Windows-style title bar with a small icon on the left and minimize, maximize, and close buttons on the right. The main area is light gray and contains two text input fields. The first is labeled "User ID:" and the second is labeled "Password:". Below these fields is a blue button with the text "Log in" in white.

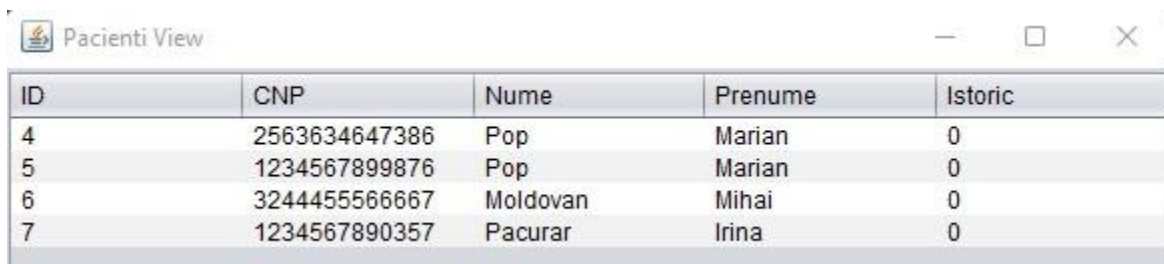
Aceasta fereastră apare în momentul rularii aplicației și are ca scop logarea unui utilizator deja existent în baza de date.

- Home Page pentru Medic sau Asistent

A screenshot of a web application window titled "Home Page Medic/Asistent". The window has a title bar with standard Windows controls. The main content area consists of five stacked, light gray buttons with rounded corners. From top to bottom, the buttons are labeled: "Pacient View", "View Personal Info", "View Programari Page", "View Consultatii Page", and "Sign Out".

Dacă utilizatorul e medic sau asistent, acesta poate vedea toți pacienții, informațiile personale, programările și consultațiile. De asemenea, are posibilitatea de a se deloga.

- Pacienții

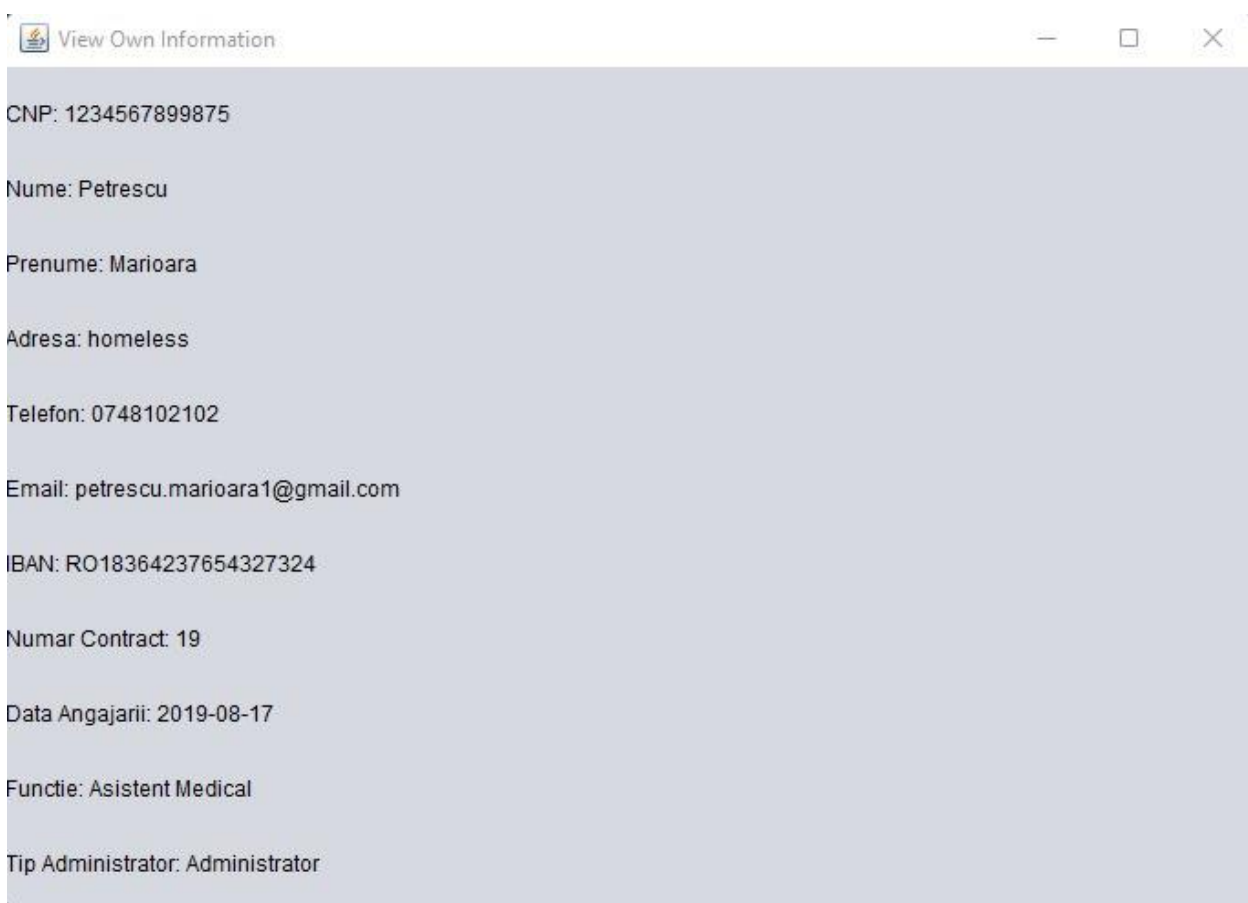


The screenshot shows a window titled "Pacienti View" with a table containing five columns: ID, CNP, Nume, Prenume, and Istoric. There are four rows of data.

ID	CNP	Nume	Prenume	Istoric
4	2563634647386	Pop	Marian	0
5	1234567899876	Pop	Marian	0
6	3244455566667	Moldovan	Mihai	0
7	1234567890357	Pacurar	Irina	0

În această pagină se pot vedea pacienții cu attributele lor.

- Informațiile personale ale medicului sau asistentului



The screenshot shows a window titled "View Own Information" with a light blue background. It displays various personal and professional details of a user.

CNP: 1234567899875

Nume: Petrescu

Prenume: Marioara

Adresa: homeless

Telefon: 0748102102

Email: petrescu.marioara1@gmail.com

IBAN: RO18364237654327324

Numar Contract: 19

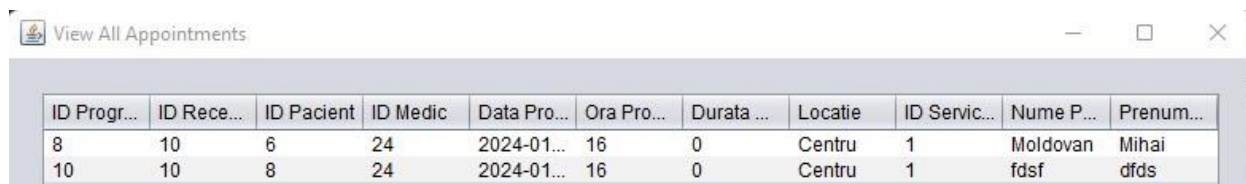
Data Angajarii: 2019-08-17

Functie: Asistent Medical

Tip Administrator: Administrator

În această pagină se afișează datele personale corespunzătoare utilizatorului care s-a logat, acesta fiind medic sau asistent.

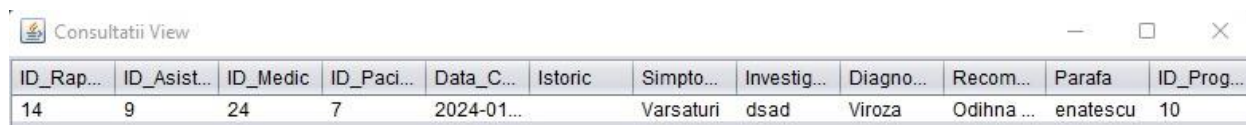
- Programările



ID Progr...	ID Rece...	ID Pacient	ID Medic	Data Pro...	Ora Pro...	Durata ...	Locatie	ID Servic...	Nume P...	Prenum...
8	10	6	24	2024-01...	16	0	Centru	1	Moldovan	Mihai
10	10	8	24	2024-01...	16	0	Centru	1	fsdf	dfds

În această pagină, utilizatorul poate urmări toate programările pacienților.

- Consultațiile



ID_Rap...	ID_Asist...	ID_Medic	ID_Paci...	Data_C...	Istoric	Simpto...	Investig...	Diagno...	Recom...	Parafa	ID_Prog...
14	9	24	7	2024-01...		Varsaturi	dsad	Viroza	Odiha ...	enatescu	10

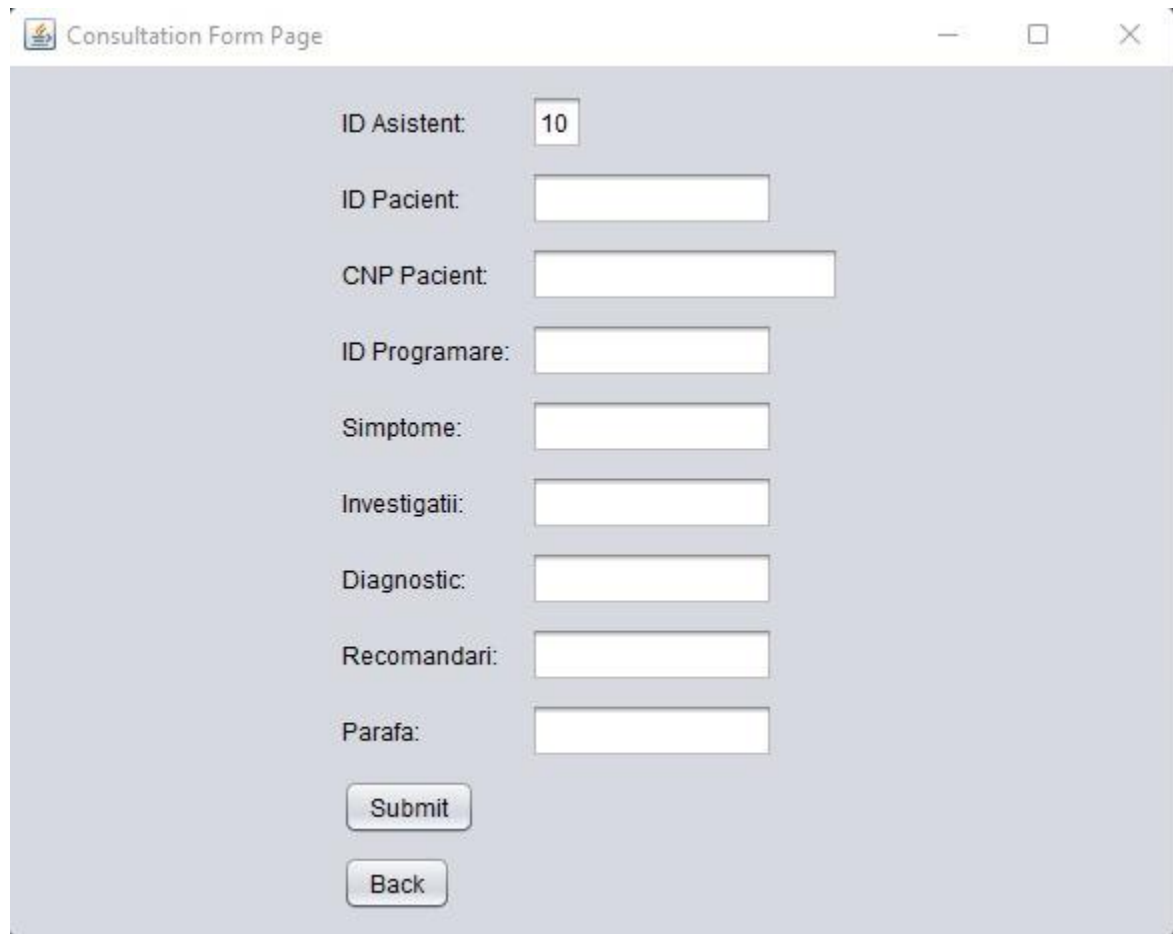
Aici se pot vizualiza consultațiile pacienților.

- Home Page pentru Receptor



Acesta are acces la rapoartele medicale și programări, putând să adauge unele noi. De asemenea, poate vizualiza pacienți, salariile și datele personale. În plus, are posibilitate de delogare.

- Adăugare Raport Medical



Consultation Form Page

ID Asistent: 10

ID Pacient:

CNP Pacient:

ID Programare:

Simptome:

Investigatii:

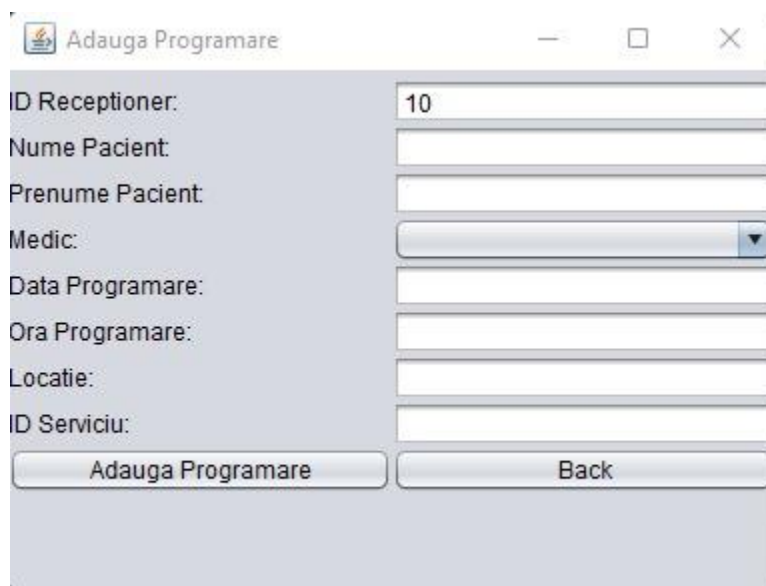
Diagnostic:

Recomandari:

Parafa:

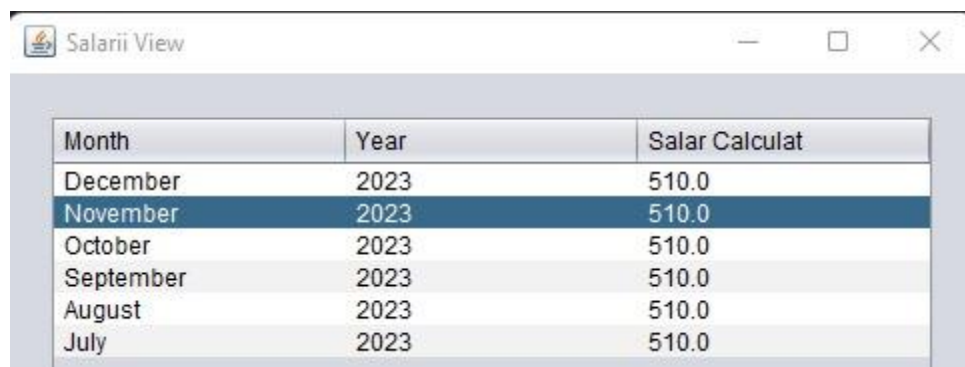
În această pagină, recepționarul poate adăuga raport medical pentru un pacient, specificând datele necesare.

- Adăugarea Programării



Fiecare programare se adaugă din această pagină, unde se introduc datele cerute, medicul putând fi selectat din listă drop-down.

- Salariile



Month	Year	Salar Calculat
December	2023	510.0
November	2023	510.0
October	2023	510.0
September	2023	510.0
August	2023	510.0
July	2023	510.0

Este afișat tabelul salariilor pe ultimele 6 luni ale recepționarului.

- Home Page pentru Inspector de Resurse Umane



Acesta are funcția de a căuta, a adăuga și a șterge un angajat, de a-și vedea datele personale, a adăuga concediu și de a se deloga din aplicație.

- Căutarea angajatului



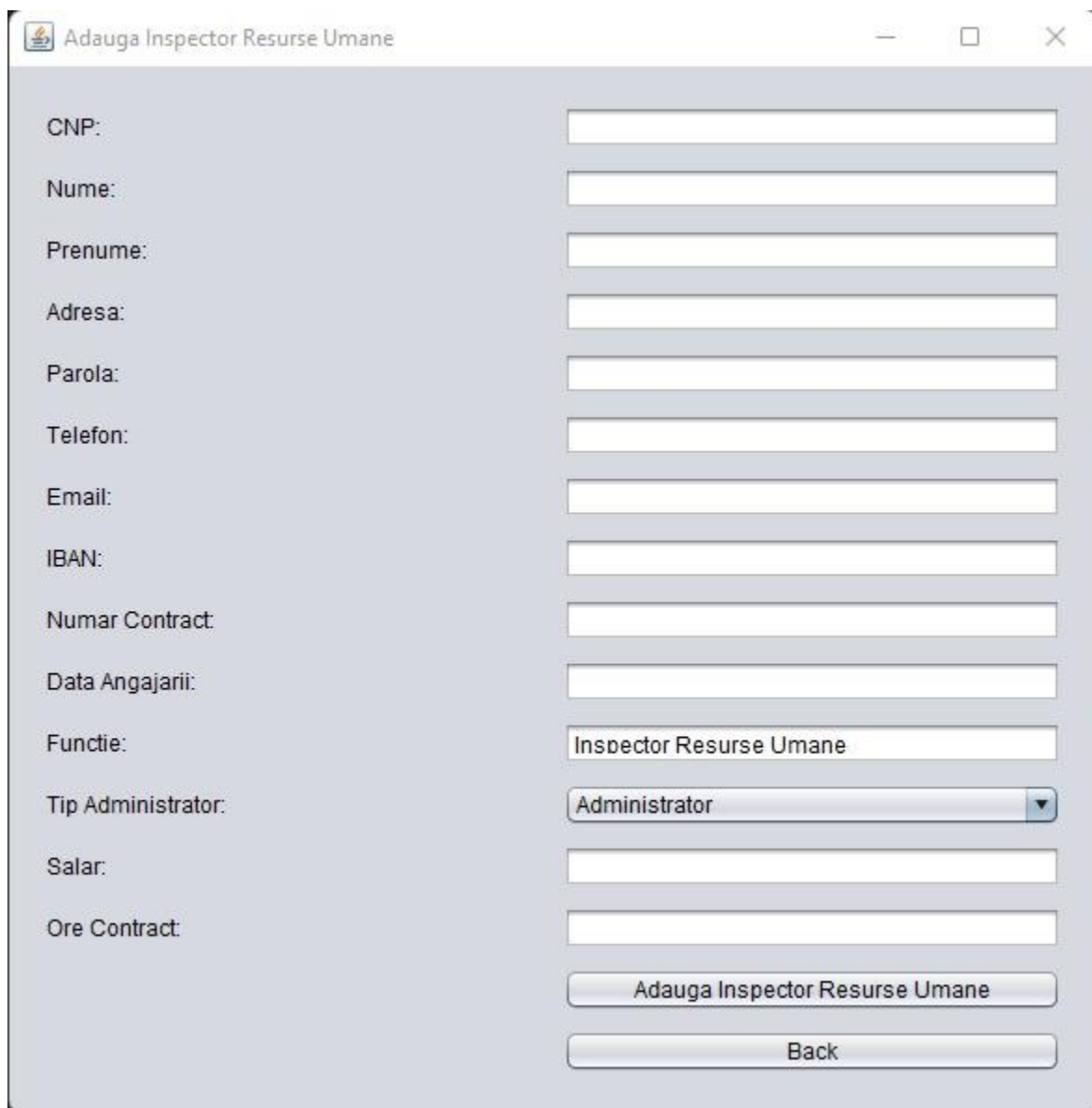
Angajatul este căutat după ID-ul său. Totodată, din această pagină se poate reveni la cea anterioară.

- Adăugarea Angajatului



Inspectorul de resurse umane are permisiunea de a adăuga un nou angajat, care poate fi inspector de resurse umane, expert financiar contabil, recepționar, asistent medical sau medic. Totodată, poate reveni la pagina anterioară.

- Adăugarea Inspectorului de Resurse Umane



The screenshot shows a web application window titled "Adauga Inspector Resurse Umane". The window contains a form with the following fields and controls:

- CNP: Text input field
- Nume: Text input field
- Prenume: Text input field
- Adresa: Text input field
- Parola: Text input field
- Telefon: Text input field
- Email: Text input field
- IBAN: Text input field
- Numar Contract: Text input field
- Data Angajarii: Text input field
- Functie: Text input field with the value "Inspector Resurse Umane"
- Tip Administrator: Drop-down menu with the value "Administrator"
- Salar: Text input field
- Ore Contract: Text input field

At the bottom of the form, there are two buttons: "Adauga Inspector Resurse Umane" and "Back".

Din această interfață, se adaugă inspector resurse umane, introducând datele specificate, fără a fi nevoie de introducerea funcției, fiind completată automat. De asemenea, tipul administratorului va fi selectat dintr-o listă de tip drop-down, iar la finalul paginii există un buton de back, ce permite accesarea paginii de dinainte.

Analog acestei pagini, se vor adăuga și angajații având celelalte funcții.

- Ștergerea angajatului



Stergere Angajat

Enter Angajat ID:

Delete Angajat

Back

Angajatul este șters, introducându-se ID-ul său. Pagina dispune de un buton, care navighează înapoi.

- Concediul



Concediu Page

Angajat ID:

Start Date (YYYY-MM-DD):

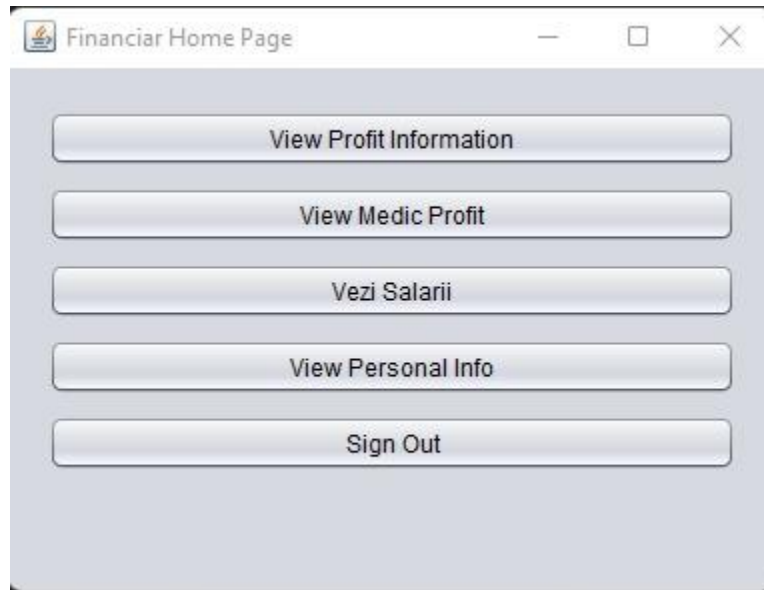
End Date (YYYY-MM-DD):

Submit

Back

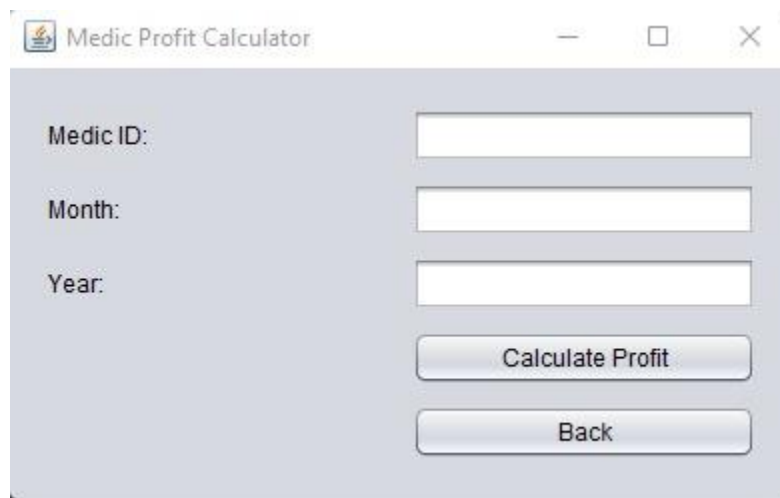
Această pagină permite stabilirea concediului a unui angajat după ID-ul său, între 2 date calendaristice.

- Home Page pentru Expert Financiar



Expertul financiar are posibilitatea să vadă informațiile legate de profit, profitul medicilor, salariile, informațiile proprii și să se delogheze.

- Calularea Profitului unui Medic



Pentru un anumit medic, se introduce ID-ul său, luna și anul, calculând-se astfel profitul său.

C. ModelsPackage

În acest pachet, avem enum-uri și clase-Model. Enum-urile sunt pentru zilele săptămânii. Am ales să folosim enum-uri în loc de string-uri pentru evitarea unor erori umane de scriere ce ar putea compromite funcționalitățile programului.

Clasele-model utilizate în acest proiect sunt:

- Angajat preia atributele din tabela angajati.
- AsistentMedical preia atributele din tabela asistenti_medicali.
- Autentificare preia atributele din tabela autentificare.
- Bonuri preia atributele din tabela bonuri.
- Concedii preia atributele din tabela concedii.
- Consultatie preia atributele din tabela consultatii.
- Medic preia atributele din tabela medici.
- OrarGeneric preia atributele din tabela orar_generic.
- OrarSpecific preia atributele din tabela orar_specific.
- Pacient preia atributele din tabela pacient.
- Programare preia atributele din tabela programari.
- ServiciuMedical preia atributele din tabela servicii_medicale.
- Utilizator preia atributele din tabela utilizatori.

7. Manual de utilizare

La pornirea aplicației, aceasta se deschide pe pagina de logare. Dacă suntem deja angajat, ne vom loga cu credențialele setate la crearea contului. Altfel, cineva de la departamentul de resurse umane va trebui să ne adauge ca angajat. Apoi, după logare, în funcție de tipul de angajat, ni se va deschide una din paginile de pornire personalizate.

1. Dacă suntem asistent medical sau medic, putem să vizualizăm lista de pacienți, să ne vedem

datele personale și să vizualizăm toate programările și consultațiile.

2. Dacă suntem recepționar, atunci avem abilitatea de a introduce pacienții în baza de date

Făcând inițial programarea. În momentul alegerii serviciului medical dorit, avem un algoritm care va afișa doar medicii care prestează acel serviciu. Programarea poate fi realizată numai dacă medicul lucrează în ziua respectivă, dacă nu este liber sau nu este în concediu. Apoi la locul de față în momentul consultației, aceasta este introdusă în baza de date tot de către recepționar. Acesta mai poate să vadă lista tuturor pacienților și datele proprii.

3. În cazul în care aparținem departamentului de resurse umane, putem căuta orice angajat și

Îi vedem datele în afară de parolă. Putem, de asemenea, să adăugăm angajați sau să îi ștergem. În momentul în care se adaugă un angajat, se alege tipul de angajat și se va deschide formular potrivit pentru a-l introduce. Mai putem adăuga concedii oricărui angajat și să ne vedem datele personale.

4. Într-un final, dacă suntem expert financiar contabil, putem să urmărim atât profitul lanțului

de policlinici, cât și profitul oricărui medic. Mai putem vedea salariile oricui pe ultimele 6 luni și datele noastre personale.

Fiecare dintre pagini conține un buton de back pentru a reveni la pagina anterioară

8. Concluzii și dezvoltări ulterioare

Aplicația ne oferă o multitudine de funcționalități pentru a eficientiza munca de stocare a informațiilor în cadrul unui lanț de policlinici și un design simplist pentru a ușura navigarea printre meniuri și ferestre.

Cea mai notabilă funcție pe care ne dorim să o implementăm în viitor este aceea de a repartiza pacienții la alți medici care prestează același serviciu ca medicul care a fost șters din baza de date. De asemenea, am dori să îmbunătățim modul în care sunt calculate și afișate salariile.

9. Bibliografie

- *Cursurile și Laboratoarele din cadrul materiilor: Baze de Date și Programare Orientată pe Obiecte.*
- <https://stackoverflow.com/>
- <https://www.w3schools.com/>
- <https://openjfx.io/>
- <https://www.geeksforgeeks.org/>
- <https://dev.mysql.com/doc/>