

Proiect

**Policlinică**

La Disciplina “Introducere în Baze de date”

Realizat de Bărăgan Andrei Iustin, Bledea Dragoș Ioan, Gavrilă Tiberiu Valentin și Țiplea Ionuț

Facultatea de Automatică și Calculatoare

An II de studiu

Grupa 30222

Prof. Coordonator,

Ivan Cosmina

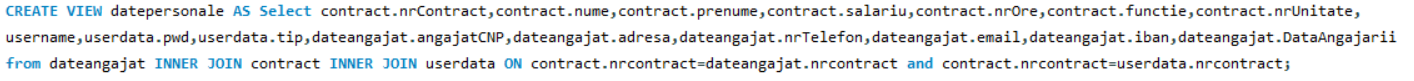
**Introducere**

Proiectul nostru dezvoltă o aplicație cu o interfață în limbajul de programare Java care permite operarea a unei multitudini de date a mai multor policlinici. Interfața va comunica în permanență cu serverul MySQL din care își va extrage, va salva, va modifica și va șterge date despre unul sau mai mulți angajați.

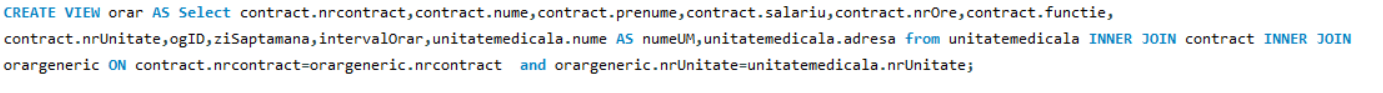
În continuare vor fi prezentate mai multe view-uri din limbajul MySQL, aplicația Java și 5 exemple folosind algebra relațională.

**MySQL**

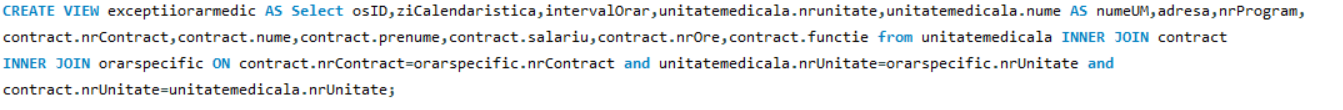
În codul din SQL nu avem proceduri sau triggere, majoritatea codului este scris în Java, iar acuma vom explica rolul fiecărui view făcut.



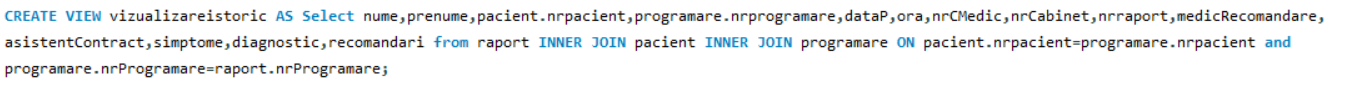
View-ul Datepersonale a fost creat pentru a ne fi mai ușor să vedem toate datele despre un angajat.



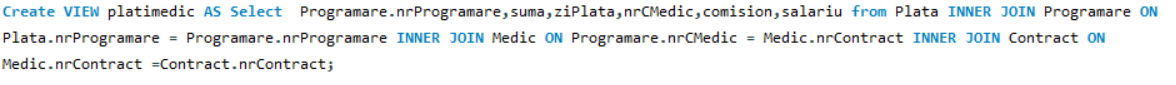
View-ul Orar a fost creat pentru a vedea orarul generic al unui angajat care nu este medic deoarece acesta are un orar mai special.



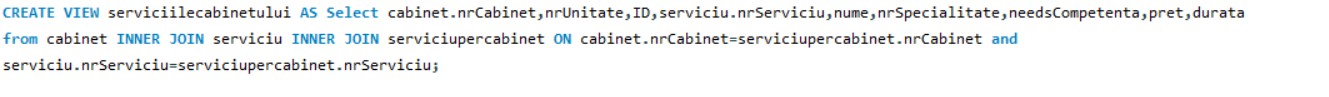
View-ul ExceptiiOrarMedic a fost creat pentru a vedea orarul specific al unui medic împreună cu unitatea medicală din care face parte.



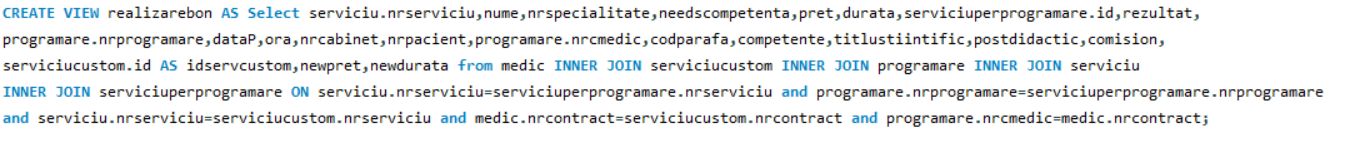
View-ul VizualizareIstoric a fost creat pentru a ne fi mai ușor să vedem istoricul programărilor și rapoartelor unui pacient.



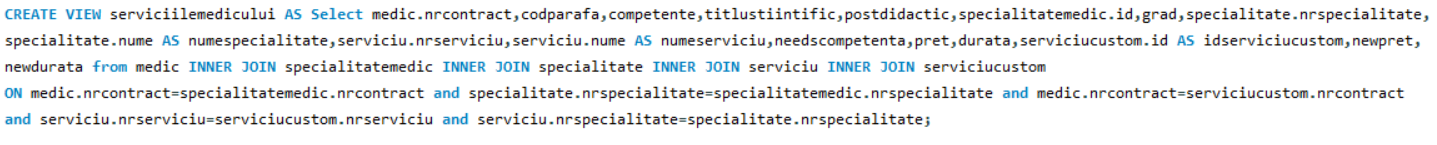
View-ul PlatiMedic a fost creat pentru a ne fi mai ușor să vedem plățile programărilor la anumiți medici efectuate de pacienți.



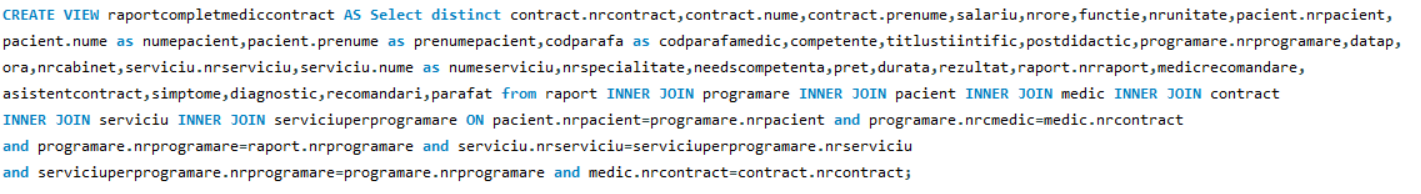
View-ul ServiciileCabinetului conține toate serviciile care pot fi prestate într-un cabinet.



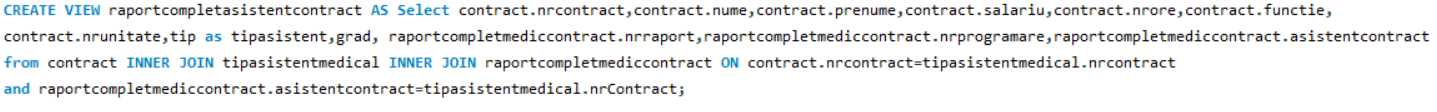
View-ul RealizareBon a fost creat pentru a ne spune dacă medicul are un serviciu căruia i-a modificat prețul sau durata atunci când e momentul să se realizeze un bon pentru acel serviciu



View-ul ServiciileMedicului a fost creat pentru vederea serviicilor unui medic în momentul creări unei programări și dacă acestea au un preț sau o durată modificată de medic.

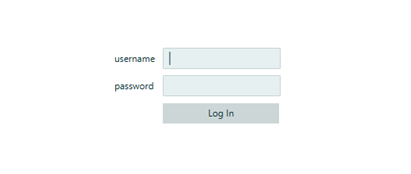


View-ul RaportCompletMedicContract a fost creat pentru a vedea un raport mult mai detaliat, care va conține și numele serviciului care a fost prestat, numele medicului care a făcut raportul și numele asistentului.

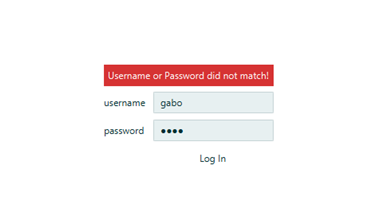


Viewul RaportCompletAsistentContract a fost creat pentru a vedea numele, prenumele și tipul asistentului în rapoartele în care sunt trecuți asistenți.

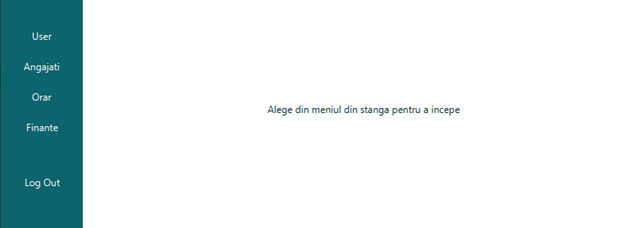
**Funcționalitatea programului**



Ecranul de autentificare nu conține decât doua câmpuri care trebuie completate cu numele de utilizator si parola contului angajatului/utilizatorului, și un buton de autentificare. În cazul în care utilizatorul completează greșit este el este întâmpinat de un mesaj de eroare.



Mai jos s-a făcut autentificarea cu un cont administrator. Am tratat un administrator ca fiind o extensie a unui angajat de la resurse umane, așadar poate realiza toate funcțiile acestuia.

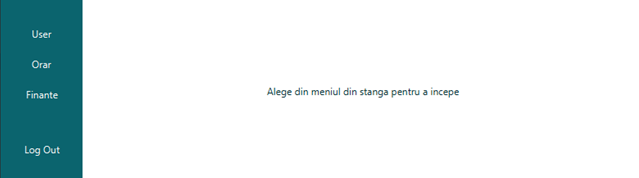


În meniul din stânga sunt 4 secțiuni: user, angajați, orar, finanțe. Funcționalități din user (vederea propriilor date personale), orar (consultarea orarului propriu), și finanțe (consultarea salariului propriu pe diferite luni) trebuie să fie disponibile pentru fiecare angajat, așadar vor apărea mereu.

Specific resurselor umane este secțiunea angajat, care permite vizualizarea și modificarea datelor personale oricărui angajat.

În continuare bor fi prezentate secțiunile disponibile pentru fiecare tip de angajat. (Exceptând resurse umane)

Agent finanțe:



Medic:



Recepționer:



Asistent Medical:

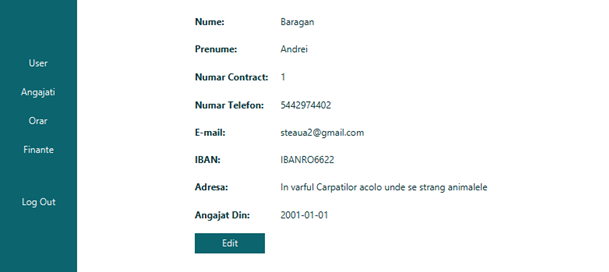


În continuare se va arata fiecare secțiune:

1. User



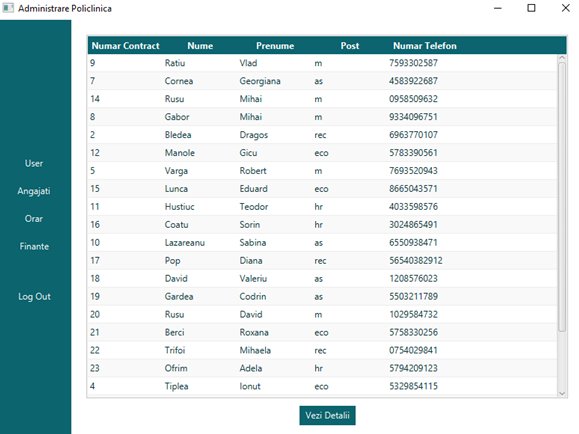
* 1. User (din punct de vedere al resurselor umane)



* 1. User (modul editare)



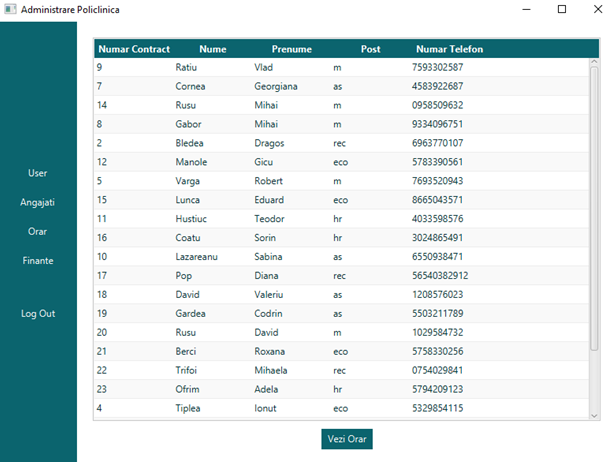
1. Angajați



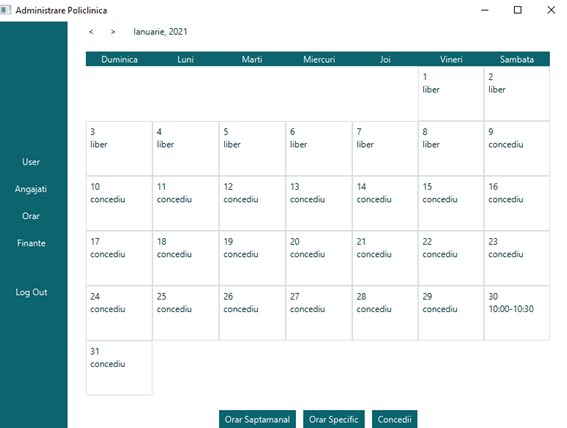
După ce s-a selectat un angajat, se pot vizualiza detaliile acestuia cu același format din 1.1

1. Orar (remarcați ca s-a modificat descrierea butonului de jos)

Prin apăsarea butonului (cu un angajat selectat) se intră în orarul acestuia



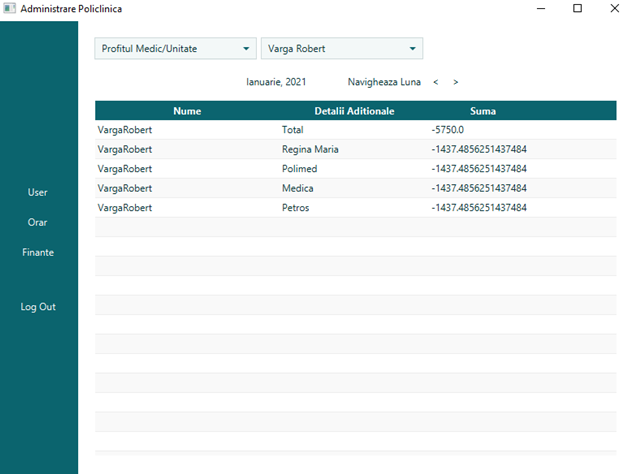
* 1. Orar efectiv



Pentru angajații care nu sunt medici, opțiunea de a modifica orarul specific nu apare.

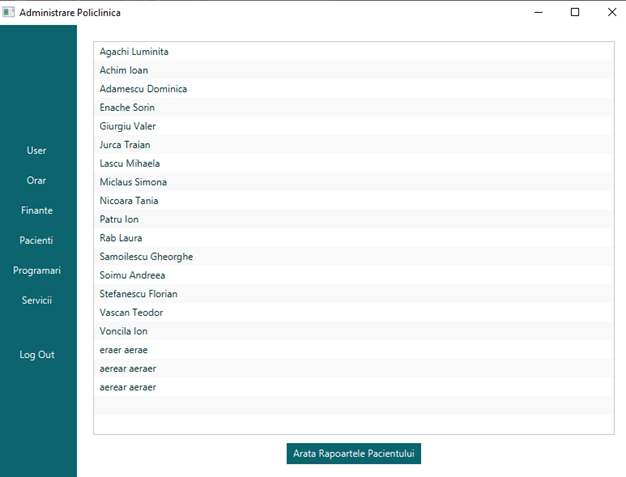
Și bineînțeles, pentru angajații care nu fac parte din resurse umane, nu apar opțiunile de modificare orar.

1. Finanțe



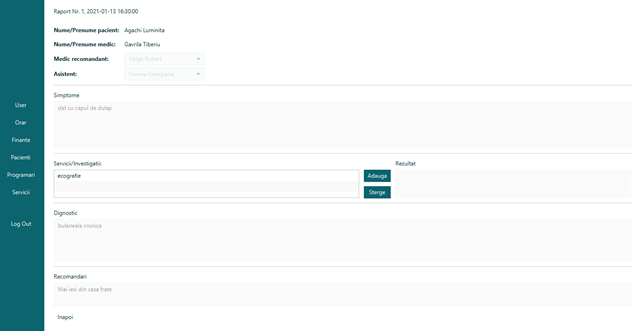
În partea de sus sunt doua meniuri, prima este folosita pentru a selecta ce se poate vedea (in funcție de angajat), a doua este specifică agentului de finanțe, și nu apare la alți angajați. Ea se folosește pentru a selecta medicul pentru a vedea ce profit a realizat pentru fiecare unitate, sau fiecare specialitate.

1. Pacienti

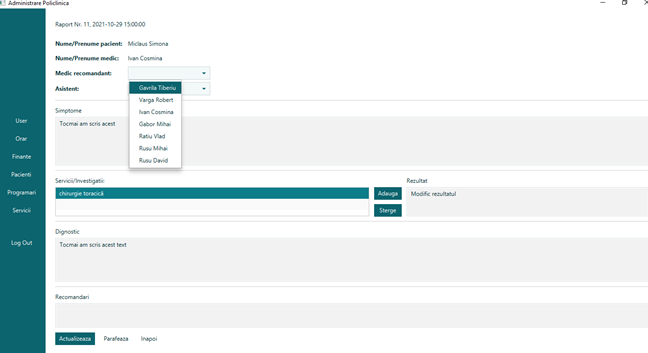


După selectarea unui pacient, se va arăta o listă de rapoarte ce aparțin pacientului în cauză (sau altfel zis, istoricul medical). După selectarea raportului, acesta se va afișa.

1. Raport



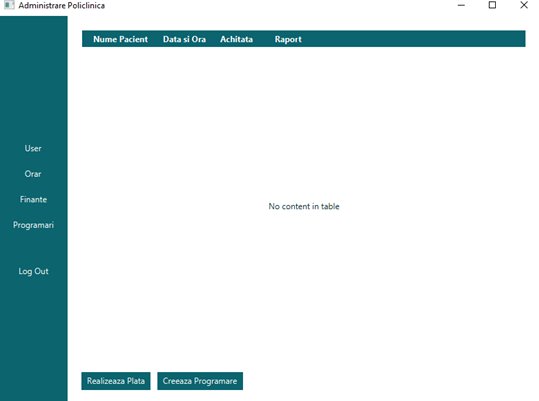
Acest specific raport este parafat si nu poate fi modificat, (nu se poate scrie in câmpurile text, si nu apar butoanele de modificare (sau nu fac nimic)).



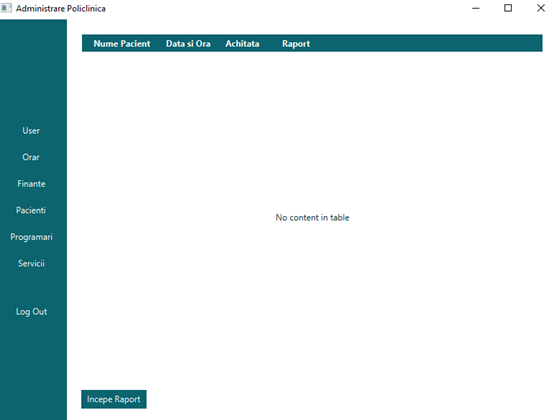
Raportul de mai sus nu este parafat si poate fi modificat. Raportul deja a fost creat anterior accesării lui, deci apare butonul de actualizare. Si angajatul care-l vizualizează este un medic, deci apare și butonul de parafare.

1. Programări (Din perspectiva unui recepționer)

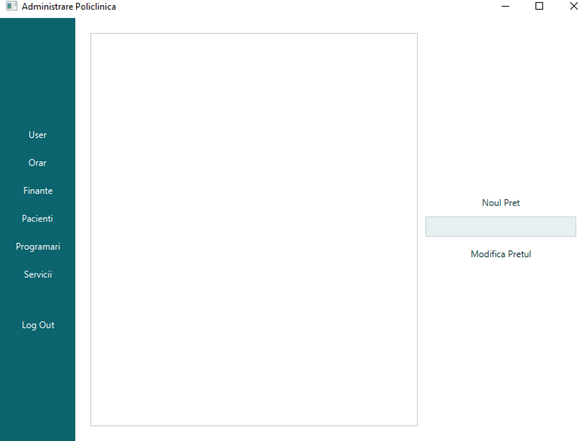
În tabela principale sunt vizualizate programările care fie nu sunt plătite, sau nu au un raport asociat.



7.1 Programări (din perspectiva unui Medic)



1. Servicii Personalizate (ale medicului)



**Algebra relațională**

Select Count(\*) from raport where nrRaport>4;

# ΠCOUNT(\*) (σ(NRRAPORT>4))

Select Count(\*) from raport where parafat=false and diagnostic is not NULL;

# ΠCOUNT(\*) (σ(PARAFAT=FALSE ^ DIAGNOSTIC IS NOT NULL(RAPORT) ) )

Select contract.nume,contract.prenume from contract, medic where medic.nrcontract=contract.nrcontract;

# ΠCONTRACTNUME,CONTRACTPRENUME (σ NRCONTRACT=NRCONTRACT(CONTRACT))

Select Concat(pacient.nume,pacient.prenume) into numepacient from pacient,programare where pacient.nrpacient=programare.nrpacient;

# ΠCONCAT(PACIENT.NUME,PACIENT.PRENUME) (σ (NRPACIENT=NRPACIENT(PACIENT) ) )

Select Count(\*) from raport,programare where nrRaport<20 and programare.nrprogramare>4;

# ΠCOUNT(\*) (σ( NRRAPORT<20 ^ NRPROGRAMARE(PROGRAMARE)>4) )