



FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DEPARTAMENTUL CALCULATOARE

Disciplina Baze de Date

An academic : 2021 – 2022

Proiect V1 – Lanț de Policlinici

Data: 4.01.2022

Realizat de:

- *Dig Bogdan Alexandru*
- *David Dan Ioan*
- *Holuță Maria Antonia*

Cuprins

- 1.Introducere
- 2.Analiza cerințelor utilizatorilor (Specificațiile de proiect)
- 3.Modelul de date și descrierea acestuia
- 4.Detalii de implementare
- 5.Concluzii
- 6.Bibliografie



1.Introducere

Proiectul presupune crearea unei aplicații folosite de utilizatorii în cadrul unui lanț de policlinici. Prin intermediul unei interfațe grafice ușor de utilizat, se va facilita accesul la baza de date a fiecărei policlinici, funcția pe care angajatul o deține fiind decisivă (de medic, asistent, inspector de resurse sau expert financiar). Orice utilizator va putea vedea datele referitoare la propria persoană, dar vor avea limitări la modulele din Homepage.

2. Analiza cerințelor utilizatorilor (Specificațiile de proiect)

2.1 Ipoteze specifice domeniului ales pentru proiect

Modelul bazei de date trebuie să ia în considerare orice situație ce poate apărea în domeniul lanțului de policlinici. Luând în considerare funcția angajatului, un inspector de resurse poate să caute orice angajat, să actualizeze orarul lor, să angajeze, să vizualizeze și să adauge concedii. Un expert financiar poate consulta profitul oricărei policlinici sau al unui medic, un receptioner face programări și emite bon, un medic scrie raporturi medicale iar un asistent se ocupă de analizele efectuate de un pacient.

2.2 Organizare structurată a cerințelor utilizator

Se vor stoca în baza de date următoarele:

- Policlinicile, orarul fiecăreia
- Angajații, salariile, orarul și concediile
- Serviciile medicale, programările, rapoartele medicale, analizele, investigațiile
- Pacienții

Aplicația trebuie să afișeze unele date din diferite tabele, să caute în funcție de anumite elemente din baza de date, să modifice, adăugând valori la unele atribute.

3.Modelul de date și descrierea acestuia

3.1 Entități și atributele lor



a. Policlinică

Lanțul de policlinici este format din mai multe unități medicale, deci vom avea nevoie de această entitate pentru a stoca detaliile fiecăreia.

Are ca attribute: ID-ul policlinicii, denumirea, adresa, numărul de cabinet și descrierea serviciilor oferite.

b. Orar_Policlinică

Acest orar oferă programul de funcționare a fiecărei policlinici în parte, pentru fiecare zi a săptămânii.

Are ca attribute: ID-ul orarului, ID-ul policlinicii, ziua, ora de sfârșit și ora de început.

c. Angajat

Angajat reprezintă o entitate foarte importantă a aplicației care reține și unele informații sensibile ce trebuie să fie protejate.

Are ca attribute:

- CNP (cheia primară)
- utilizator (numele de utilizator care a fost înregistrat aplicației)
- parola (parola care a fost înregistrată aplicației)
- adresa
- email
- nume
- prenume
- numar_ore (reține numărul de ore din contractul de muncă care trebuie realizate până la final de lună)
- telefon
- IBAN
- data_angajare (data în care a fost angajat)
- functie (funcția pe care o deține, aceasta fiind de medic, asistent medical, recepționar, expert financiar sau inspector resurse, fiind importantă aplicației pentru a oferi diferite drepturi în funcție de acestea)



- salariu_negociat (salariul din contract pe lună)

c. Orar_Generic

Acest orar reprezintă un orar general în care angajatul lucrează, care poate fi modificat mai apoi de orarul specific.

Are ca attribute: ID-ul policlinicii, CNP-ul angajatului, ziua, ora de sfârșit și ora de început.

d. Orar_Specific

Acest orar reprezintă un orar mai detaliat în care angajatul poate să lucreze, având ore în plus sau în minus față de orarul generic.

Are ca attribute: ID-ul policlinicii, CNP-ul angajatului, data, ora de sfârșit și ora de început.

e. Medic

Un angajat poate să fie un medic, oferindu-se drepturi particulare (poate să consulte datele referitoare la propria persoană și are toate drepturile necesare modulului pentru gestiunea activităților operaționale)

Are ca attribute: CNP-ul angajatului, postul didactic, codul de parafă și comisionul stabilit.

f. Grad

Gradul este specific fiecărui medic în parte.

Are ca attribute: ID-ul medicului, gradul (specialist sau primar) și denumirea specialității.

g. Specialitate

Specialitatea este specifică fiecărui medic în parte. Se va reține denumirea fiecărei specialități.

h. Competență

Competența este specifică fiecărui medic în parte. Se va reține CNP-ul medicului și competența acestuia.

i. Cabinet (BONUS)

Un medic poate să aibă unul sau mai multe cabinete. Se va reține numărul fiecărui cabinet (nr_cabinet), ID-ul policlinicii și CNP-ul medicului.

j. Asistent medical



Un angajat poate să fie un asistent medical.

Are ca atribute: CNP, grad (secundar sau principal) și tip (generalist, laborator, sau radiologie).

k. Salariu

Fiecare angajat are un anumit salariu.

Are ca atribute: CNP-ul angajatului, salariul (exprimat numeric) și data încasării.

l. Concediu

În această tabelă se va specifica, pentru orice angajat, perioada în care acesta va avea zile libere.

Are ca atribute: CNP-ul angajatului, data plecării (data_plecure) și data revenirii (data_revenire).

m. Echipament_medical

În această tabelă se va specifica, pentru fiecare cabinet ce echipament deține.

Are ca atribute: ID_cabinet, ID_echipament și echipament (enumerarea/descrierea echipamentului specific fiecărui cabinet)

n. Lista_Servicii_Cabinet

Acest tabel este necesar pentru a rezolva relația de many-to-many dintre cabinet și servicii_medicale.

Are ca atribute: ID_cabinet, ID_serviciu.

o. Servicii_Medicale

În această tabelă se va specifica orice serviciu pus la dispoziția pacienților, fiind necesară o programare pentru aceștia de către un recepționar.

Are ca atribute: ID_serviciu, durata, denumire_serviciu și pret.

p. Lista_Servicii_Client

Acest tabel este necesar pentru a rezolva relația de many-to-many dintre servicii_medicale și programare.

Are ca atribute: ID_programare, ID_serviciu.



q. Programare

În această tabelă se vor face toate programările de către orice recepționar, care va corespunde unui pacient.

Are ca atribute:

- ID_programare
- ID_policlinica
- CNP_pacient
- CNP_medic
- data_consultație
- ora

r. Pacient

Tabela Pacient va reține informații principale despre fiecare pacient, pentru a putea fi programat.

Are ca atribute: nume, prenume și CNP (CNP-ul pacientului)

s. Raport_Medical

După fiecare programare, un medic poate să scrie un raport medical pentru un pacient.

Are ca atribute:

- ID_programare
- CNP_pacient
- CNP_medic
- diagnostic
- recomandări
- simptome
- nume_medic_recomandat
- prenume_medic_recomandat
- nume_medic_consultant



- nume_medic_consultant
- nume_asistent
- prenume_asistent

t. Analize_Medicale

După fiecare programare, un asistent poate să completeze detalii despre fiecare analiză efectuată.

Are ca atribute: ID_raport, detalii_analize, validare (când o analiză este validată, nu se vor mai putea face modificări), analiza_efectuata.

u. Investigatii

Se vor stoca aici detaliile investigației de la orice raport medical.

Are ca atribute: ID_raport, ID_serviciu și detalii_investigație.

v. Portofoliu_Servicii

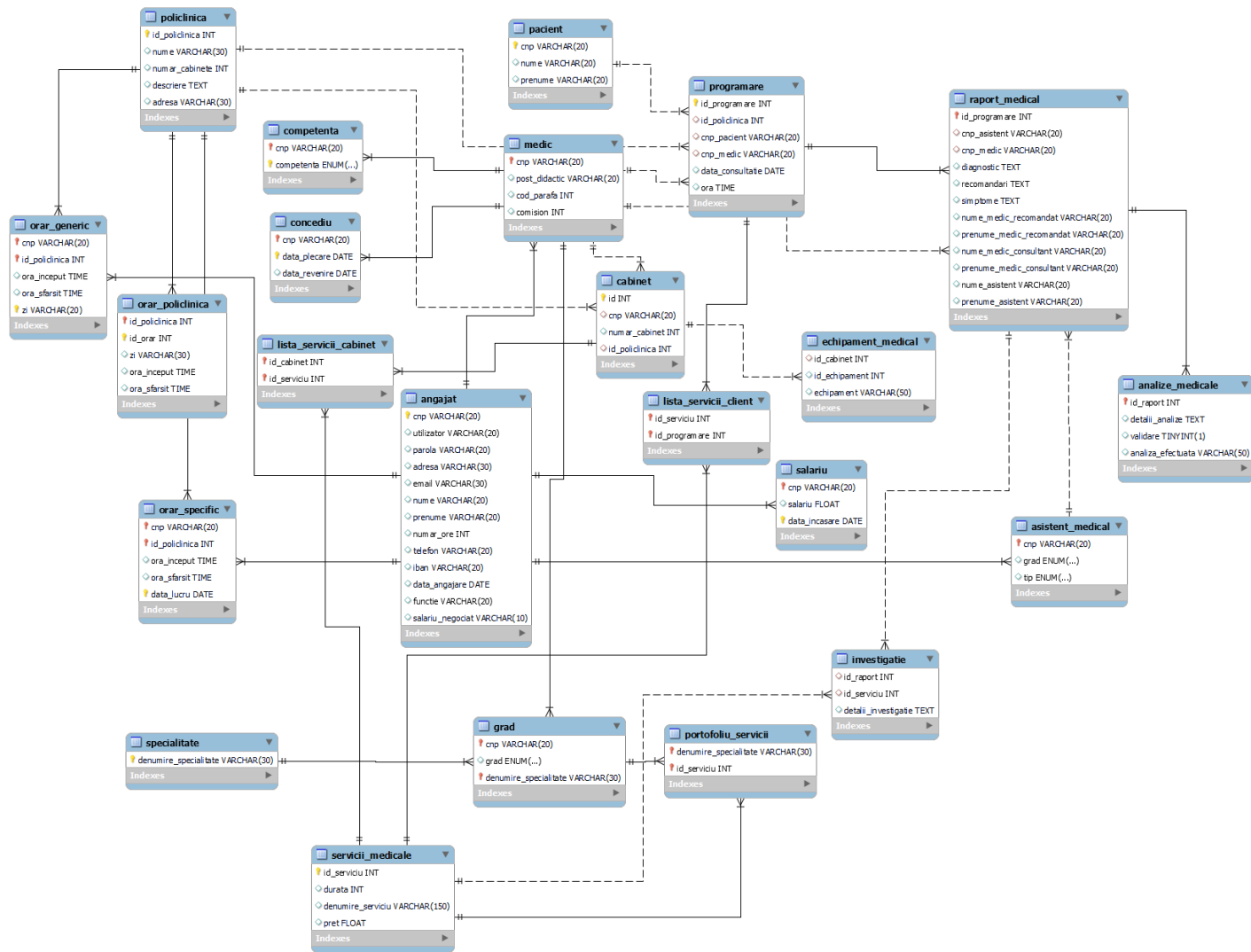
Această entitate va rezolva relația de many-to-many dintre grad (gradul medicului) și servicii_medicale, având ca atribute: denumire_specialitate și id_serviciu.

w. Portofoliu echipament (BONUS)

Această entitate va rezolva relația de many-to-many dintre Servicii_Medicale și Echipament_Medical, având ca atribute: id_serviciu și id_echipament.



3.2 Diagrama EEL/UML pentru modelul de date complet





3.3 Algebră relațională

Login

SELECT * FROM Angajat where parola = 'inspector' AND utilizator = 'cassandra';

Π {cnp, utilizator, parola, adresa, email, nume, prenume, numar_ore, telefon, iban, data_angajare, functie, salariu_negociat} (σ (parola = 'inspector' AND utilizator = 'cassandra')(Angajat))

Concedii

SELECT c.cnp, c.data_plecure, c.data_revenire, a.nume, a.prenume FROM concediu c join angajat a on c.cnp = a.cnp ;

Π {c.cnp, c.data_plecure, c.data_revenire, a.nume, a.prenume} (σ { c.cnp = a.cnp} (concediu \bowtie angajat))

ConcediiRestricted

SELECT * FROM concediu WHERE cnp = '1234';

Π { cnp, data_plecure, data_revenire} (σ cnp = '1234'(Concediu))

OrarGeneric

SELECT o.ora_inceput, o.ora_sfarsit, o.zi, o.id_policlinica, o.cnp, p.adresa, p.nume FROM orar_generic o join policlinica p on o.id_policlinica = p.id_policlinica WHERE o.cnp = '1234';

Π { o.ora_inceput, o.ora_sfarsit, o.zi, o.id_policlinica, o.cnp, p.adresa, p.nume } (σ { o.id_policlinica = p.id_policlinica AND o.cnp = '1234' } (orar_generic o \bowtie policlinica p))

GestiuneResurseUmane

SELECT * FROM orar_policlinica;

Π {id_policlinica, id_orar, zi, ora_inceput, ora_sfarsit} (orar_policlinica)



4. Detalii de implementare

- MySQL Workbench, localhost: 3306
- Conexiunea la baza de date: mysql-connector-java-8.0.27.jar
- Interfață: Java Swing, mediu de design al interfeței: GUI Designer

Proiectul are mai multe clase, pentru orice entitate în parte, clasa Angajat fiind una principală aplicației, pentru a reține toate datele referitoare la persoana care folosește contul și pentru a se acorda diferite priorități. Pe lângă acestea, mai sunt prezente unele clase pentru o livizibilitate mai bună a proiectului și pentru unele metode de calcul. Restul claselor sunt folosite împreună cu form-ul lor, cu ajutorul lui GUI Designer, pentru a face posibilă o navigare și accesare ușoară a utilizatorilor în aplicație.

Drepturile acordate fiecărui utilizator și elementele de utilizare:

În baza de date:

Adminii vor putea actualiza, șterge sau insera în tabelele legate de angajați, dar nu vor putea opera asupra unor utilizatori de tip admin. Pentru aceasta, avem rolul de super_admin.

În aplicație:

Inspectorii de resurse au full access la modulul de gestiune a resurselor umane experții financiari au la dispoziție o secțiune în care pot consulta orarul săptămânal și informații despre concediile efectuate, pe când restul pot să vadă doar date referitoare la propria persoană.

Experții financiari au full access la modulul de operații financiar contabile, iar ceilalți pot să vadă doar date referitoare la propria persoană.

Recepționerii vor face programări și vor emite bonuri fiscale în modulul de gestiune al activităților operaționale, medicii vor scrie rapoarte medicale iar asistenții vor putea să modifice date referitoare la analize până acestea sunt validate. Inspectorul și expertul financiar nu primesc deloc acces la acest modul.



UNIVERSITATEA TEHNICĂ
DIN CLUJ-NAPOCA

LogIn:

LogIn

UserName

cassandra

Password

.....

Clear OK

Homepage:

Welcome

Home Account Info

-----Selecteaza modulul-----

Gestiune Resurse Umane

Operatii Financiar Contabile

Gestiune Activitati Operationale

Log Out



La Homepage utilizatorii vor alege modulul (inspectorii de resurse si experții financiari nu au deloc acces la ultimul modul), se vor deloga sau vor consulta informațiile despre cont:

Welcome

Home Account Info

Datele Mele

CNP: 3547900495966397

Adresa: 5235 Portage Terrace

Email: cglencrosche0@japanpost.jp

Numar de ore: 86

Telefon: 417-541-8495

IBAN: GR62 3113 972B 5430

Data angajare: 2021-04-07

Nume: Glencrosche

Prenume: Casandra

Functie: inspector resurse

Salariu Negociat: 2000

Log Out

Gestiune Resurse Umane pentru utilizator de tip inspector resurse:

Welcome

Back

-----Selecteaza operatia-----

nume prenume functie

Cauta Angajat

Orar Policlinici

Concedii

Angajează

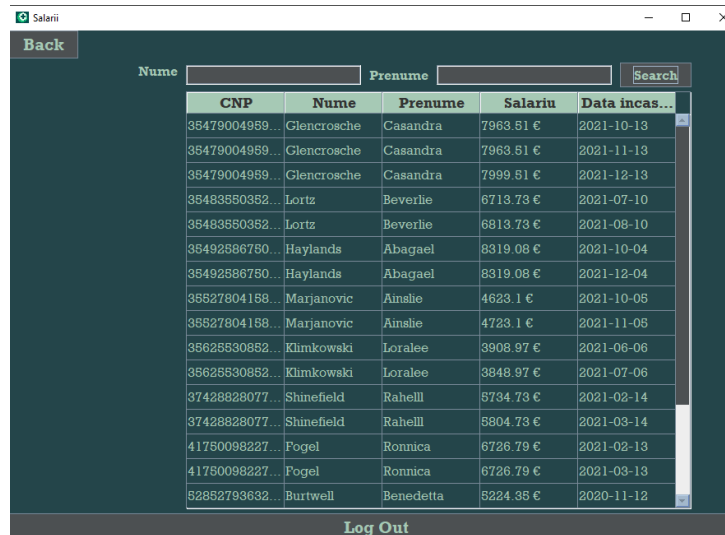
Log Out



Operații Financiar Contabile pentru utilizator de tip expert financiar:



Vizualizarea salariilor:



Elemente de securizare a aplicației:

Aplicația este securizată cu ajutorul prepare statement-urilor în cazul concatenării, pentru a nu permite orice atac SQL Injection.

5. Concluzii, limitări si dezvoltări ulterioare

Nevoia de a reține informații legate de domeniul medical devine din ce în ce mai mare, oferind un acces mai ușor și mai rapid atât pentru pacienți, cât și pentru alți utilizatori. Interfața



este simplu de utilizat, fiind structurată în pagină de Login și Homepage pentru a selecta modulul din care fac parte diferite operații.

O dezvoltare ulterioară o constituie crearea conturilor pentru pacienți, în care aceștia ar putea vizualiza programările sale actuale, respectiv ulterioare, analizele și rezultatele acestora. În același timp, pacienții ar putea să își facă propriile programări, selectând medicul, specializarea, data și ora. De asemenea, extinderea operațiilor accesibile la alte tipuri de angajați ar aduce o îmbunătățire a aplicației.

6. Bibliografie

<https://examples.javacodegeeks.com/desktop-java/ide/intellij-gui-designer-example/>

<https://stackoverflow.com/questions/18521892/execute-update-with-union>

<https://stackoverflow.com/questions/299495/how-to-add-an-image-to-a-jpanel>

<https://www.formdev.com/jformdesigner/doc/ui/designer/jtabbedpane/>

https://ftp.utcluj.ro/pub/users/civan/IBD/2_LABORATOR/12_Normalizare/IBD_Lab8.2_FN.pdf

<https://www.youtube.com/watch?v=aIdIXsi1qTU>

<https://www.youtube.com/watch?v=Kmg00avvEw&list=WL&index=2&t=5370s>