

Proiect

la disciplina

Baze de Date

Sistem Informatizat de Gestiune a unui Lanț
de Policlinici

Autor: Maxim Staver

Grupa: 30223

Data prezentării: 14 ianuarie 2025

Rezumat:

Acest proiect prezintă dezvoltarea unei aplicații informatice de gestiune a unui lanț de policlinici, având scopul de a facilita gestionarea angajaților, pacienților, programărilor, rapoartelor medicale, salariilor și a altor resurse administrative. Aplicația este destinată să ajute utilizatorii (administratori, super-administratori, angajați) să acceseze și să gestioneze informațiile într-o manieră eficientă și intuitivă. Interfața grafică este concepută pentru a fi ușor de utilizat și accesibilă pe mai multe nivele de competență. Proiectul include funcționalități avansate pentru fiecare tip de utilizator, inclusiv posibilitatea de a vizualiza, modifica și adăuga informații relevante pentru activitatea zilnică a policlinicilor.

Cuprins

1. Introducere

1.1 Descrierea Generală a Proiectului

1.2 Scopul și Obiectivele

2. Tema Proiectului

2.1 Prezentarea Aplicației

2.2 Funcționalități și Utilizatori

3. Descrierea Bazei de Date

3.1 Structura generala

3.2 Detalii specifice

3.3 Nivelul de Normalizare

4. Elemente de programare a functionalitatilor

4.1 Proceduri și Triggere.Vederi.Lista completa si exemple

4.2. Alte exemple de Cod SQL

5. Interfața Grafică a Utilizatorului (GUI)

5.1 Descrierea functionalitatilor per tip de utilizator

5.2 Detalii de implementare funcționalități specifice

6. Manual de Utilizare

6.1 Înregistrare și Autentificare

7. Concluzii și dezvoltari ulterioare

7.1 Analiza funcționalităților curente- cele mai interesante solutii adoptate

7.2 Dezvoltari ulterioare

1. Introducere

1.1. Descrierea Generală a Proiectului

Acest proiect presupune dezvoltarea unei aplicații informatice destinate gestionării activităților dintr-un lanț de policlinici. Aplicația va oferi o interfață grafică prin care utilizatorii vor putea interacționa cu baza de date MySQL, gestionând informațiile relevante pentru activitățile zilnice ale policlinicilor. Sistemul va permite gestionarea angajaților, a serviciilor financiare-contabile, dar și administrarea operațiunilor curente din cadrul instituției. Aplicația va integra un sistem de control al accesului bazat pe roluri (RBAC), fiecare utilizator având drepturi de acces diferite în funcție de rolul său.

1.2 Scopul și Obiectivele

Scopul principal al acestui proiect este de a crea o aplicație complet funcțională care să ajute la administrarea unui lanț de policlinici, prin gestionarea eficientă a resurselor umane, a activităților financiare și a operațiunilor curente. Aplicația va facilita interacțiunea cu baza de date, permițând utilizatorilor să efectueze operațiuni de actualizare, căutare și gestionare a datelor relevante.

Obiectivele proiectului sunt:

- **Proiectarea unei baze de date normalizate:** Baza de date va include tabele interconectate, care vor gestiona informațiile despre angajați, servicii financiare și operațiuni curente.
- **Dezvoltarea unui sistem de control al accesului bazat pe roluri (RBAC):** Fiecare utilizator va avea drepturi specifice, în funcție de rolul său (administrator, contabil, medic, etc.), pentru a asigura securitatea și integritatea datelor.
- **Crearea unei interfețe grafice ușor de utilizat:** Aplicația va include o interfață prietenoasă pentru utilizator, care să faciliteze accesul la funcționalitățile de gestionare a policlinicii, precum adăugarea, modificarea și ștergerea datelor din baza de date.

Prin realizarea acestui proiect, studentul va dobândi experiență în proiectarea bazelor de date, programarea în Java, manipularea bazelor de date MySQL și dezvoltarea aplicațiilor cu interfețe grafice.

2. Tema Proiectului

2.1 Prezentarea Aplicației

Aplicația dezvoltată este un sistem informatic de gestionare a unui lanț de policlinici, având ca principal scop automatizarea și eficientizarea activităților administrative, financiare și medicale. Prin intermediul unei interfețe grafice intuitive, utilizatorii pot interacționa cu baza de date MySQL, gestionând informațiile în funcție de drepturile atribuite fiecărui rol.

Sistemul permite administrarea angajaților, programărilor, rapoartelor medicale, salariilor și resurselor financiare ale policlinicilor. De asemenea, aplicația asigură un control riguros al accesului utilizatorilor printr-un sistem bazat pe roluri (RBAC), oferind fiecărui utilizator funcționalități specifice.

Aplicația include mai multe tipuri de utilizatori, fiecare având acces la date și operațiuni relevante pentru activitatea sa:

- **Administrator** – gestionează utilizatorii sistemului și poate adăuga, modifica sau șterge conturi.
- **Super-administrator** – are drepturi extinse, inclusiv asupra administratorilor.
- **Inspector resurse umane** – gestionează și actualizează datele angajaților, inclusiv programul de lucru și concediile.
- **Expert financiar-contabil** – monitorizează veniturile și cheltuielile policlinicilor, având acces la datele financiare.
- **Medic** – gestionează programările pacienților, completează rapoarte medicale și poate personaliza lista de servicii medicale.
- **Asistent medical** – sprijină medicii în completarea rapoartelor medicale și monitorizează pacienții.
- **Recepționar** – realizează programări pentru pacienți și emite bonuri fiscale pentru consultații.

Fiecare utilizator accesează doar informațiile și funcționalitățile relevante pentru rolul său, asigurând astfel securitatea și integritatea datelor.

2.2 Funcționalități și Utilizatori

Aplicația oferă un set de funcționalități esențiale pentru gestionarea eficientă a policlinicilor:

Funcționalități generale:

- Autentificare securizată bazată pe roluri.
- Gestionarea utilizatorilor și a drepturilor de acces.

- Administrarea angajaților și evidența concediilor.
- Programarea și gestionarea consultațiilor pacienților.
- Emiterea de rapoarte medicale personalizate.
- Monitorizarea salariilor și a veniturilor policlinicilor.

Funcționalități specifice utilizatorilor:

Administrator:

- Aduagă, modifică și șterge conturi de utilizatori.

Super-administrator:

- Poate opera inclusiv asupra conturilor de administratori.

Inspector resurse umane:

- Caută și editează date despre angajați după nume, prenume și funcție.
- Accesează programul de lucru și evidența concediilor pentru toți angajații.

Expert financiar-contabil:

- Consultă situația financiară a policlinicilor.
- Vizualizează veniturile și cheltuielile, calculând profitul.

Recepționar:

- Creează programări pentru pacienți.
- Emite bonuri fiscale la finalizarea consultațiilor.

Asistent medical:

- Completează informații în rapoartele medicale ale pacienților.

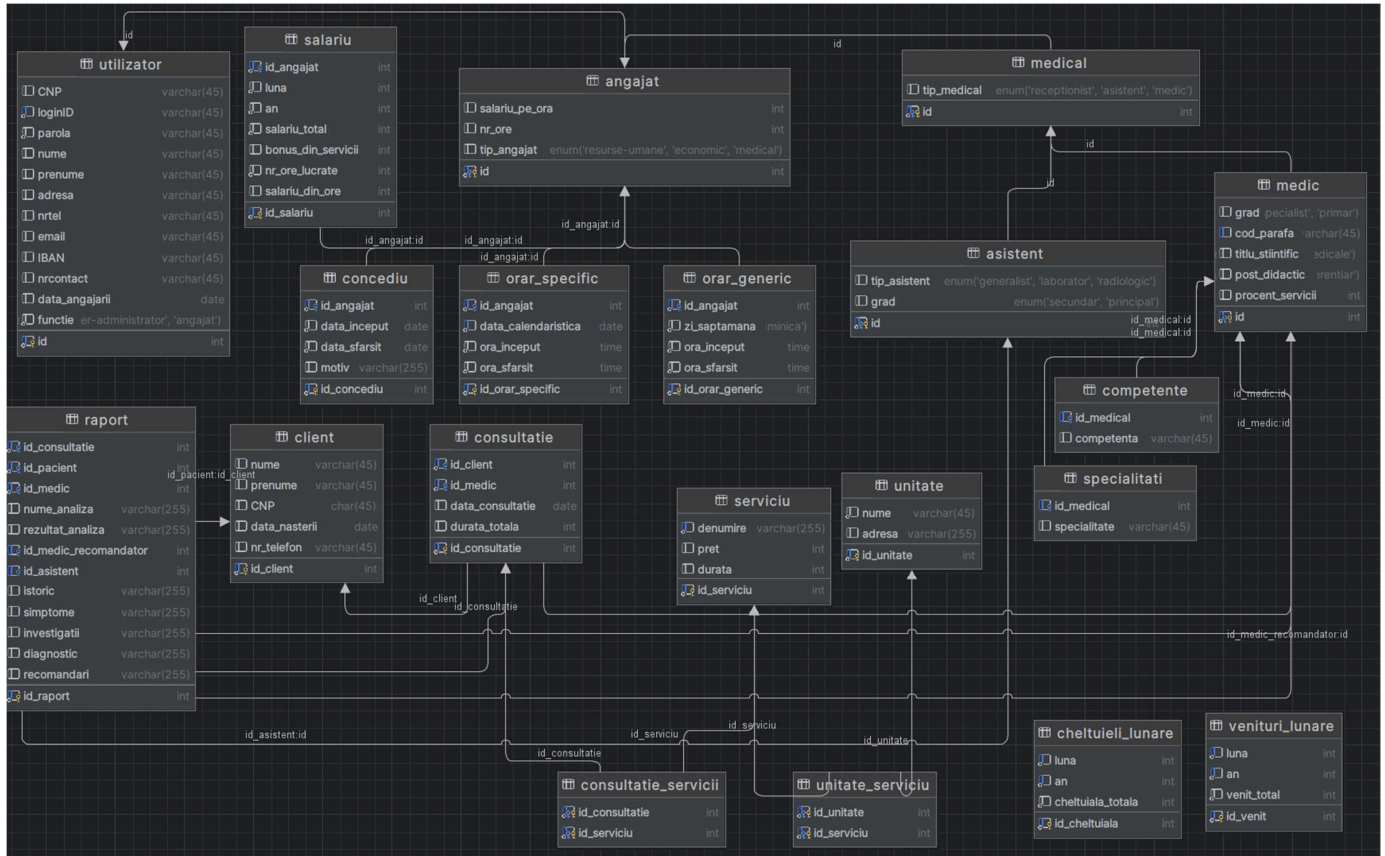
Medic:

- Vizualizează lista pacienților programați pentru ziua curentă.
- Completează rapoarte medicale pentru fiecare pacient.
- Gestionează lista de servicii medicale oferite.
- Poate opta pentru calculul salariului pe baza consultațiilor efectuate.

Această structură modulară asigură o gestionare eficientă a resurselor și a fluxului de lucru din cadrul policlinicilor, facilitând atât activitatea administrativă, cât și cea medicală.

3. Descrierea Bazei de Date

3.1 Structura generala



3.2 Detalii specifice

1. cheltuieli_lunare

Stochează cheltuielile totale ale clinicii pentru fiecare lună și an.

2. client

Reține informațiile despre clienții (pacienții) clinicii, inclusiv nume, prenume, CNP, data nașterii și date de contact.

3. serviciu

Conține lista serviciilor medicale oferite de clinică, cu denumirea, prețul și durata fiecărui serviciu.

4. unitate

Stochează informații despre unitățile (sediile) clinicii, inclusiv numele și adresa fiecărei unități.

5. unitate_serviciu

Asociază unitățile clinicii cu serviciile pe care le oferă, printr-o relație many-to-many între unitate și serviciu.

6. utilizator

Reprezintă utilizatorii aplicației (angajați și administratori), incluzând date de autentificare și informații personale.

7. angajat

Extinde tabelul utilizator și conține informații specifice angajaților, precum salariul pe oră, numărul de ore lucrate și tipul angajatului (resurse umane, economic, medical).

8. concediu

Înregistrează perioadele de concediu ale angajaților, incluzând datele de început și sfârșit, precum și motivul concediului.

9. medical

Extinde tabelul angajat pentru personalul medical, specificând tipul de angajat medical (receptionist, asistent, medic).

10. asistent

Extinde tabelul medical și definește tipul de asistent (generalist, laborator, radiologic) și gradul acestuia (secundar, principal).

11. medic

Extinde tabelul medical, incluzând informații despre grad, codul de parafă, titlul științific și procentul din venituri primit din servicii.

12. competente

Stochează competențele suplimentare ale unui medic.

13. consultatie

Înregistrează consultațiile efectuate, legând un client de un medic, cu data consultației și durata totală.

14. consultatie_servicii

Leagă consultațiile de serviciile medicale efectuate în cadrul acestora.

15. orar_generic

Definește programul de lucru standard al angajaților, cu zilele săptămânii și intervalele orare.

16. orar_specific

Reprezintă un program personalizat pentru un angajat într-o anumită zi calendaristică.

17. raport

Înregistrează rapoartele medicale asociate consultațiilor, conținând detalii despre analize, investigații, diagnostic și recomandări.

18. salariu

Stochează informații despre salariile lunare ale angajaților, inclusiv bonusurile și numărul de ore lucrate.

19. specialitati

Reține specialitățile medicilor.

20. venituri_lunare

Înregistrează veniturile totale ale clinicii pentru fiecare lună și an.

21. Funcții SQL (esteAdministrator, esteAngajat, esteSuperAdministrator)

Sunt funcții care verifică dacă un utilizator are un anumit rol în sistem.

Această bază de date este bine organizată pentru gestionarea unui lanț de clinici, acoperind aspecte precum utilizatori, angajați, servicii, programări și finanțe.

3.3 Nivelul de Normalizare

Baza de date utilizată pentru gestionarea clinicilor este normalizată cel puțin până la **a treia formă normală (3NF)**, ceea ce asigură eliminarea anomaliilor de inserare, actualizare și ștergere, menținând în același timp integritatea și eficiența datelor.

1. Prima Formă Normală (1NF)

Aceasta este respectată deoarece fiecare tabel are un identificator unic (cheie primară), iar toate atributele conțin valori atomice (nedivizibile). De exemplu, tabelul client are câmpuri precum nume, prenume și CNP, care nu conțin liste sau seturi de valori multiple.

2. A Doua Formă Normală (2NF)

Baza de date respectă 2NF deoarece toate tabelele care au chei compuse își asociază atributele dependente în tabele separate. De exemplu, tabelul consultatie_servicii gestionează relația many-to-many dintre consultatie și serviciu, evitând astfel redundanța datelor despre servicii în cadrul fiecărei consultații.

3. A Treia Formă Normală (3NF)

Se respectă și 3NF deoarece toate atributele ne-cheie depind exclusiv de cheia primară a tabelului și nu există dependențe tranzitive. De exemplu, în tabelul medic, informațiile despre grad, codul de parafă și procentul din venituri depind direct de cheia primară (id_medic) și nu de alte atribute.

4. Elemente de programare a functionalitatilor

4.1 Proceduri și Triggere. [Vederi. Lista completa si exemple](#)

Funcții pentru Autentificare și Verificarea Rolurilor

- verificaAutentificare(String loginID, String parola)

- Verifică dacă există un utilizator cu **loginID** și **parolă** în baza de date.
- Returnează **funcția (rolul)** utilizatorului (Administrator, Super-Administrator, Angajat) dacă autentificarea este validă, altfel returnează null.

Cod:

```
String sql = "SELECT functie FROM utilizator WHERE loginID = ? AND parola = ?"
```

Funcții pentru Gestionarea Angajaților

- obtineTipAngajat(String loginID)

- Returnează tipul de angajat (resurse-umane, economic, medical) asociat unui utilizator.
- Se face un **JOIN** între utilizator și angajat pentru a găsi tipul de angajat corespunzător

- getInfoAngajat(String loginID)

- Returnează un Map<String, String> cu toate informațiile despre un angajat, inclusiv nume, prenume, salariu pe oră și număr de ore lucrate.
- Se face un **JOIN** între utilizator și angajat pentru a obține detalii complete despre angajat.

```
String query = "SELECT u.num, u.prenume, u.adresa, u.nr_tel, u.email, u.IBAN, " +  
                "u.nr_contact, u.data_angajarii, u.functie, a.salariu_pe_ora, a.nr_ore, " +  
                "a.tip_angajat " +  
                "FROM utilizator u " +  
                "JOIN angajat a ON u.id = a.id " +  
                "WHERE u.loginID = ?";
```

- actualizeazaUtilizator(int id, ...)

- Actualizează informațiile unui utilizator în baza de date.

- Există două variante: una pentru **Admin**, alta pentru **Super-Admin** (care poate modifica și rolul utilizatorului).

```
String sql = "UPDATE utilizator SET CNP = ?, loginID = ?, parola = ?, nume = ?,  
prenume = ?, " + "adresa = ?, nrtele = ?, email = ?, IBAN = ?, functie = ?, nrcontact =  
? WHERE id = ?";
```

Funcția cautaAngajati

caută angajați în baza de date pe baza **numelui, prenumelui și funcției**, excluzând administratorii. Folosește **JOIN** pentru a include detalii despre tipul de angajat (tip_angajat) și specializarea medicală (tip_medical). Permite căutare flexibilă cu **LIKE**, acceptând și valori parțiale. Returnează un ResultSet cu rezultatele găsite, care poate fi utilizat pentru afișarea în interfața grafică.

Cod SQL utilizat în funcție:

```
String sql = "SELECT u.*, a.tip_angajat, m.tip_medical " +  
            "FROM utilizator u " +  
            "LEFT JOIN angajat a ON u.id = a.id " +  
            "LEFT JOIN medical m ON a.id = m.id " +  
            "WHERE u.functie NOT IN ('Administrator', 'Super-Administrator') " +  
// Excludem administratorii  
            "AND (u.nume LIKE ? OR ? IS NULL OR ? = '') " +  
            "AND (u.prenume LIKE ? OR ? IS NULL OR ? = '') " +  
            "AND (u.functie LIKE ? OR a.tip_angajat LIKE ? OR m.tip_medical LIKE ?  
" +  
            "      OR ? IS NULL OR ? = '')";
```

Funcția getBonuri returnează lista consultațiilor efectuate, incluzând **ID-ul consultației, numele și prenumele clientului, precum și data consultației**. Aceasta face un **JOIN** între tabelele consultatie și client pentru a asocia fiecare consultație cu datele pacientului. Rezultatele sunt returnate sub formă de ResultSet, care poate fi utilizat pentru afișarea bonurilor în interfața grafică.

Cod SQL utilizat în funcție:

```
SELECT c.id_consultatie, cl.nume, cl.prenume, c.data_consultatie  
FROM consultatie c  
JOIN client cl ON c.id_client = cl.id_client;
```

Funcția getDetailsForConsultatie returnează detalii complete despre o consultație, incluzând **numele pacientului, numele medicului, data consultației, durata, serviciile oferite, prețul acestora și unitatea medicală unde a avut loc consultația**. Aceasta folosește mai multe **JOIN-uri** între tabelele consultatie, client, utilizator, serviciu și unitate, pentru a furniza toate informațiile necesare. Rezultatul este stocat într-o listă de mapări (List<Map<String, String>>), care poate fi utilizată pentru afișare în interfața grafică.

Cod SQL utilizat în funcție:

```
SELECT c.id_consultatie, cl.nume AS nume_client, cl.prenume AS prenume_client,  
       u.nume AS nume_medic, u.prenume AS prenume_medic, c.data_consultatie,  
       c.durata_totala,  
       s.denumire AS serviciu, s.pret AS pret_serviciu, un.nume AS unitate  
FROM consultatie c  
JOIN client cl ON c.id_client = cl.id_client  
JOIN utilizator u ON c.id_medic = u.id
```

```

JOIN consultatie_servicii cs ON c.id_consultatie = cs.id_consultatie
JOIN serviciu s ON cs.id_serviciu = s.id_serviciu
JOIN unitate_serviciu us ON s.id_serviciu = us.id_serviciu
JOIN unitate un ON us.id_unitate = un.id_unitate
WHERE c.id_consultatie = ?
GROUP BY c.id_consultatie, cl.numa, cl.prenume, u.numa, u.prenume,
         c.data_consultatie, c.durata_totala, s.denumire, s.pret, un.numa;

```

getRaportData() realizează o interogare SQL pentru a obține informațiile complete dintr-o serie de tabele interconectate: raport, consultatie, client, utilizator, consultatie_servicii, și serviciu. Interogarea selectează date detaliate despre rapoartele de consultație, incluzând informații despre pacient, medic, servicii oferite și rezultatele analizelor. Rezultatul este un set de date care include informații precum numele pacientului, prenumele pacientului, numele medicului, simptomele, diagnosticul, recomandările, precum și serviciile oferite, toate fiind grupate în funcție de ID-ul raportului. Funcția returnează un ResultSet care conține aceste informații.

Codul SQL utilizat pentru interogare este:

```

SELECT r.id_consultatie, r.numa_analiza, r.rezultat_analiza, r.id_medic_recomandator,
       r.id_asistent, r.istoric, r.simptome, r.investigatii, r.diagnostic,
       r.recomandari,
       c.numa AS numa_pacient, c.prenume AS prenume_pacient, m.numa AS numa_medic,
       m.prenume AS prenume_medic,
       GROUP_CONCAT(s.denumire SEPARATOR ', ') AS servicii_oferite
FROM raport r
JOIN consultatie cons ON r.id_consultatie = cons.id_consultatie
JOIN client c ON cons.id_client = c.id_client
JOIN utilizator m ON cons.id_medic = m.id
LEFT JOIN consultatie_servicii cs ON cons.id_consultatie = cs.id_consultatie
LEFT JOIN serviciu s ON cs.id_serviciu = s.id_serviciu
GROUP BY r.id_raport;

```

4.2. Alte exemple de Cod SQL

Implementările de cod SQL sunt în clasa `UtilizatorDAO`, care se ocupă de gestionarea tuturor interacțiunilor cu baza de date, cum ar fi citirea, scrierea, actualizarea și ștergerea datelor.

5. Interfața Grafică a Utilizatorului (GUI)

5.1 Descrierea functionalitatilor per tip de utilizator

Aplicația include mai multe tipuri de utilizatori, fiecare având acces la date și operațiuni relevante pentru activitatea sa:

- **Administrator** – gestionează utilizatorii sistemului și poate adăuga, modifica sau șterge conturi.
- **Super-administrator** – are drepturi extinse, inclusiv asupra administratorilor.

- **Inspector resurse umane** – gestionează și actualizează datele angajaților, inclusiv programul de lucru și concediile.
- **Expert financiar-contabil** – monitorizează veniturile și cheltuielile policlinicilor, având acces la datele financiare.
- **Medic** – gestionează programările pacienților, completează rapoarte medicale și poate personaliza lista de servicii medicale.
- **Asistent medical** – sprijină medicii în completarea rapoartelor medicale și monitorizează pacienții.
- **Recepționar** – realizează programări pentru pacienți și emite bonuri fiscale pentru consultații.

5.2 Detalii de implementare funcționalități specifice

Ferestrele Administrator si Super Administrator

Se pot adauga, modifica si sterge utilizatori, afisand detalii despre acestia. Daca se logeaza un utilizator Administrator, acesta poate actiona doar asupra angajatilor, spre deosebire de Super Administrator care are permisiuni totale.

Fereastra Administrator

CNP:

Login ID:

Parola:

Nume:

Prenume:

Adresa:

Nr. Tel:

Email:

IBAN:

Functie:

Nr. Contact:

ID	CNP	Login ID	Nume	Prenume	Functie	Data Angajarii
1	1234567890123	admin01	Popescu	Ion	Administrator	2022-01-15
2	2345678901234	superadmin01	Ionescu	Maria	Super-Administrator	2023-02-20
3	3456789012345	hr	Georgescu	Andrei	Angajat	2021-03-10
4	4567890123456	economic01	Marin	Elena	Angajat	2020-06-05
5	5678901234567	admin03	Dumitrescu	Ionela	Administrator	2024-04-25
6	6789012345678	medic01	Vasile	Adrian	Angajat	2021-09-17
7	7890123456789	superadmin04	Popa	Gabriela	Super-Administrator	2023-07-30
8	8901234567890	receptionist01	Nistor	Mihai	Angajat	2022-08-19
9	9012345678901	asistent01	Constantin	Florin	Angajat	2020-11-22
10	1122334455667	admin05	Iordache	Stefan	Administrator	2024-01-10

Adauga Utilizator

Modifica Utilizator

Sterge Utilizator

Incarca Utilizatori

Fereastra Informatii Angajat – au acces toti utilizatorii de tip angajat

Fereastra Angajat

Informații Angajat

Salariu pe Oră: 45

Tip Angajat: medical

Nr. Tel: 0711122334

Email: mihai.nistor@email.com

IBAN: RO49HHHH1B3100759712211388

Data Angajării: 2022-08-19

Adresă: Str. Tineretului, Nr. 25, Brașov

Nume: Nistor

Nr. Ore: 40

Prenume: Mihai

Funcție: Angajat

Nr. Contact: 0810112233

Fereastra receptionist – accesibila numai utilizatorilor de tip receptionist, are 3 tab-uri: Creare programare, Inregistrare pacient si Emitere Bon.

Fereastra Receptionist

Creare Programare Inregistrare Pacient Emitere Bon

ID Client:

ID Medic:

Data Consultatie:

Selectează	ID Serviciu	Denumire	Preț	Durată (minute)
<input type="checkbox"/>	1	Consultatie generala	150	30
<input checked="" type="checkbox"/>	2	Consultatie dermatologica	200	40
<input checked="" type="checkbox"/>	3	Radiografie toracica	100	20
<input type="checkbox"/>	4	Analize de sange	80	15
<input checked="" type="checkbox"/>	5	Electrocardiograma	120	20
<input type="checkbox"/>	6	Tratament stomatologic	300	60

Crează Programare

Fereastra Receptionist

Creare Programare

Înregistrare Pacient

Emitere Bon

ID Client	Nume	Prenume	CNP	Data Nașterii	Nr. Telefon
1	Popescu	Ion	1960512301234	1990-05-12	0721123456
2	Ionescu	Maria	2970512345678	1985-11-25	0745123456
3	Vasile	George	1910207896543	1992-02-18	0735123456
4	Georgescu	Ana	2060503456789	1988-09-30	0741123456
5	Dumitru	Elena	2950723345678	1995-06-14	0727123456

Nume:

Prenume:

CNP:

Data Nașterii (YYYY-MM-DD):

Nr. Telefon:

Adaugă Pacient

Daca se apase pe butonul **Emitere Bon** (tab-ul 3) se va deschide o fereastră cu informatiile despre acesta unde s-a calculat pretul in functiile de serviciile care au fost selectate cand a fost create programarea (tab-ul 1).

Bon Consultație

Nume Client:

Popescu

Prenume Client:

Ion

Preț:

230,00 RON

Nume Medic:

Georgescu

Prenume Medic:

Alexandru

Data Consultației:

2025-01-14

Durata (minute):

45

Unitate:

Clinica MedLife

Servicii:

Analize de sange - 80,00 RON

Consultatie generala - 150,00 RON

Fereastra Receptionist

Creare Programare

Înregistrare Pacient

Emitere Bon

ID Client:

ID Medic:

Data Consultatie:

Selectează	ID Serviciu	Denumire	Preț	Durăta (minute)
<input type="checkbox"/>	1	Consultatie generala	150	30
<input checked="" type="checkbox"/>	2	Consultatie dermatologica	200	40
<input checked="" type="checkbox"/>	3	Radiografie toracica	100	20
<input type="checkbox"/>	4	Analize de sange	80	15
<input checked="" type="checkbox"/>	5	Electrocardiograma	120	20
<input type="checkbox"/>	6	Tratament stomatologic	300	60

Crează Programare

Fereastra Medic/Asistent permite vederea programarilor facute de utilizatorii receptionist in tab-ul Programari si vederea istoricului de rapoarte in tab-ul Istoric. Doar medicii au posibilitatea de a edita un raport si de a-l parafa.

Fereastra Medic Asistent

Programări

Istoric

ID Consultatie	ID Pacient	Nume Pacient	Prenume Pacient	Nume Medic	Prenume Medic	Data Consultatiei	Servicii
1	1	Popescu	Ion	Vasile	Adrian	2025-01-11	Consultatie generala, Analize de sange
2	2	Ionescu	Maria	Popescu	Ion	2025-01-12	Consultatie dermatologica, Radiografie toracica
3	3	Vasile	George	Ionescu	Maria	2025-01-13	Analize de sange, Electrocardiograma
4	4	Georgescu	Ana	Georgescu	Alexandru	2025-01-13	Tratament stomatologic
5	5	Dumitru	Elena	Vasile	Adrian	2025-01-14	Consultatie generala, Analize de sange
6	1	Popescu	Ion	Georgescu	Alexandru	2025-01-14	Consultatie generala, Analize de sange

Intocmire Raport

Fereastra Medic Asistent

Programări

Istoric

ID Consult...	Nume Analiza	Rezultat Analiza	ID Medic Reco...	ID Asistent	Istoric	Simptome	Investigatii	Diagnostic	Recomandari	Nume Pacient	Nume Medic	Servicii Oferite
1	Analiza sange	Normal	12	9	Istoric medical g...	Durere in piept	Electrocardiogr...	Cardiopatie isc...	Tratament cu bet...	Popescu Ion	Vasile Adrian	Consultatie generala, Anali...
2	Ecografie abdom...	Anomalii vizibile	0	0	Fara antecedente	Durere abdominala	Ecografie abdo...	Apendicita	Interventie chirur...	Ionescu Maria	Popescu Ion	Consultatie dermatologica, ...
3	Tomografie com...	Nodul pulmonar ...	6	0	Fumat cronic	Tuse persistenta	Tomografie	Cancer pulmonar	Tratament oncolo...	Vasile George	Ionescu Maria	Analize de sange, Electroca...
4	Rmn cerebral	Normal	11	0	Fara antecedent...	Durere de cap	RMN cerebral	Migrena	Tratament analge...	Georgescu Ana	Georgescu Alexandru	Tratament stomatologic
5	Analiza sange	Anemie	0	0	Istoric medical slab	Oboseala cronica	Analize de sange	Anemie feripriva	Suplimente de fie...	Dumitru Elena	Vasile Adrian	Consultatie generala, Anali...

Modificare Raport

Parafare Raport

Fereastra Intocmire Raport se deschide cand un utilizator asistent sau medica apasa pe butonul Intocmire raport din fereastra Medic/Asistent (tab-ul programari)

Fereastra Medic Asistent (Programări)

ID Consultatie	ID Pacient	Nume Pacient
1	1	Popescu
2	2	Ionescu
3	3	Vasile
4	4	Georgescu
5	5	Dumitru
6	1	Popescu

Intocmire Raport Consultatie

ID Consultatie:

ID Pacient:

ID Medic:

Nume Analiza:

Rezultatul Analizei:

ID Medic Recomandator:

ID Asistent:

Istoric:

Simptome:

Investigatii:

Diagnostic:

Recomandari:

Trimite Raport

Servicii

la Analize de sange
logica, Radiografie toracica
Electrocardiograma
logic
la Analize de sange
la Analize de sange

Fereastra Gestiune Resurse Umane – angajatii HR pot vedea, adauga, modifica, sterge si analiza orarul altor angajati. Totodata pot cauta dupa nume, prenume sau functie in baza de date.

Gestionare Resurse Umane

Nume: Prenume:

Funcție: **Caută**

ID	Nume	Prenume	Funcție	Tip Angajat	Tip Medical
3	Georgescu	Andrei	Angajat	resurse-umane	
4	Marin	Elena	Angajat	economic	
6	Vasile	Adrian	Angajat	medical	medic
8	Nistor	Mihai	Angajat	medical	receptionist
9	Constantin	Florin	Angajat	medical	asistent
11	Popescu	Ion	Angajat	medical	medic
12	Ionescu	Maria	Angajat	medical	medic
13	Georgescu	Alexandru	Angajat	medical	medic

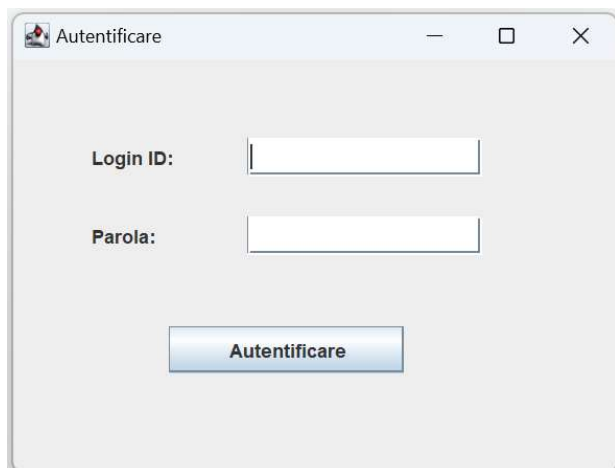
Modifică **Șterge** **Vezi Orar**

Modul Economic — □ ×

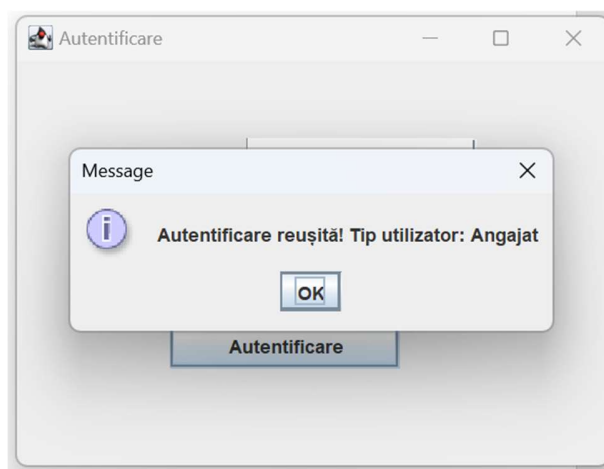
6. Manual de Utilizare

6.1 Înregistrare și Autentificare

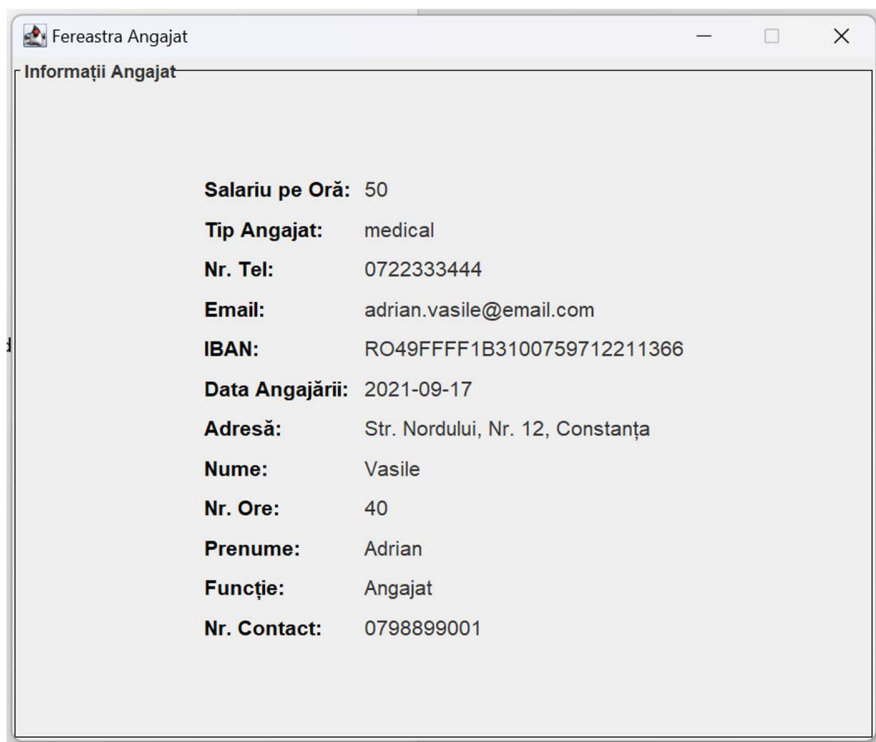
La logare se introduce numele de utilizator și parola, iar sistemul va deschide ferestrele în funcție de funcțiile din baza de date. Dacă se introduc date greșite sau lipsesc date la logare, va apărea un mesaj în legătură cu acest lucru.



Dupa introducerea corecta a datelor, o fereastra pop-up va anunta ce tip de utilizator s-a logat.



Daca utilizatorul nu e admin/sper-admin, automat se deschide fereastra cu informatii impreuna cu celelalte ferestre corespunzatoare tipului de utilizator.



Salariu pe Oră:	50
Tip Angajat:	medical
Nr. Tel:	0722333444
Email:	adrian.vasile@email.com
IBAN:	RO49FFFF1B3100759712211366
Data Angajării:	2021-09-17
Adresă:	Str. Nordului, Nr. 12, Constanța
Nume:	Vasile
Nr. Ore:	40
Prenume:	Adrian
Funcție:	Angajat
Nr. Contact:	0798899001

7. Concluzii și dezvoltari ulterioare

7.1 Analiza funcționalităților curente- cele mai interesante solutii adoptate

Aplicatia pentru gestionarea unui lanț de policlinici a fost concepută astfel încât să răspundă nevoilor variate ale diferitelor categorii de utilizatori, de la administratori și super-administratori până la angajați de tip recepționeri, asistenți medicali, medici, inspectori de resurse umane și experți financiari-contabili. Una dintre cele mai interesante soluții adoptate este integrarea unui sistem complex de gestionare a programărilor și a rapoartelor medicale. Astfel, fiecare pacient beneficiază de un raport detaliat după consultație, care poate fi folosit pentru a urmări progresul în cadrul serviciilor medicale oferite. În plus, angajații pot vizualiza informațiile relevante doar pentru ei, menținând astfel un nivel adecvat de confidențialitate și acces la datele specifice rolurilor lor. Interfața grafică ușor de utilizat, combinată cu funcționalitățile complexe de gestionare a resurselor și programărilor, face ca aplicația să fie o soluție eficientă și adaptată nevoilor organizaționale ale policlinicilor.

7.2 Dezvoltari ulterioare

Pentru a extinde și îmbunătăți funcționalitățile aplicației, sunt posibile mai multe direcții de dezvoltare ulterioare. Un aspect important ar fi integrarea unui sistem de management al stocurilor pentru medicamente și echipamente, astfel încât să se poată monitoriza în timp real disponibilitatea acestora în fiecare policlinică. De asemenea, dezvoltarea unui sistem de notificări și alerte pentru angajați, cum ar fi notificarea despre schimbări în programul lor de lucru sau actualizări privind pacienții, ar contribui la o gestionare mai eficientă a activităților zilnice. Pe termen lung, integrarea unui sistem de analiză a datelor care să ofere rapoarte detaliate despre performanțele financiare ale lanțului de policlinici ar putea ajuta la optimizarea strategiilor economice și îmbunătățirea serviciilor oferite. Aceste îmbunătățiri ar consolida aplicația, transformând-o într-un instrument complet de gestionare a clinicilor, cu un impact pozitiv asupra eficienței operaționale și a satisfacției pacienților.