

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE

DEPARTAMENTUL CALCULATOARE

PROIECT

La Disciplina

INTRODUCERE ÎN BAZE DE DATE

TITLUL LUCRĂRII:

Platformă de studiu

PROF. COORDONATOR: COSMINA IVAN & TAT TEODORA

AUTORI:

Păucean Ioana

Cordea Maria

Apetrei Diana-Andreea

An academic 2024-2025

grupa 30221

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Cuprins

1. Introducere	3
1.1. Repartizarea Responsabilităților	3
1.1. Descrierea Generală a Proiectului	4
1.2. Scopul și Obiectivele	4
2. Tema Proiectului	5
2.1. Prezentarea Aplicației	5
2.2. Funcționalități și Utilizatori	5
3. Descrierea Bazei de Date	7
3.1. Structura generala	7
3.2. Detalii specifice	9
3.3. Nivelul de Normalizare	9
4. Elemente de programare a functionalitatilor	10
4.1. Procedur.Vederi.Lista completa si exemple	10
4.2. Alte exemple de Cod SQL	13
5. Interfața Grafică a Utilizatorului (GUI)	13
5.1. Detalii de implementare funcționalități specifice	14
6. Manual de Utilizare	17
6.1. Înregistrare și Autentificare	17
6.2. Gestionarea Informațiilor per utilizatori	17
7. Concluzii și dezvoltari ulterioare	23
7.1. Analiza funcționalităților curente- cele mai interesante solutii adoptate	23
7.2. Dezvoltari ulterioare	24

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

1. Introducere

1.1. Repartizarea Responsabilităților

În cadrul proiectului, am îndeplinit toate cerințele specificate în tabelul din enunț, precum și pe cele suplimentare din secțiunea de bonus.

Sarcinile au fost repartizate astfel încât fiecare membru al echipei să contribuie în mod echilibrat la realizarea obiectivelor. Pentru organizarea logică și definirea relațiilor dintre tabelele bazei de date, toate cele trei membre ale echipei au colaborat îndeaproape, asigurând o structură bine fundamentată și eficientă, conform cerințelor proiectului.

Paucean Ioana a fost responsabilă de dezvoltarea interfeței utilizator și de implementarea mecanismelor necesare pentru integrarea acestora cu baza de date, asigurând o comunicare eficientă între front-end și back-end.

Pentru realizarea bazei de date, Cordea Maria și Apetrei Diana și-au împărțit astfel sarcinile:

Cordea Maria:

- creare_grup_studiu
- inscriere_student_grup_studiu
- AdaugaActivitate
- inscriere_activitate_studiu
- vizualizare_membri_grupa
- parasire_student_grup_studiu
- adauga_mesaj_grup_studiu
- vizualizare_mesaje_student_grup_studiu
- programare_examen_colocviu
- programare_activitate
- vizualizare_grupe_student
- vizualizare_activitati_profesor
- vizualizare_activitati_student
- sugestii_de_participanti
- adauga_profesor_la_activitate
- sugereaza_activitate_grup
- afiseaza_activitati_studiu
- afisare_studenti_curs
- cautare_dupa_nume_si_rol
- Eveniment: VerificaActivitatiPeriodice

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Apetrei Diana:

- inserare_administrator
- inserare_profesor
- inserare_student
- inserare_disciplina
- inserare_activitate_discipline
- modifica_ponderi_disciplina
- afisare_informatii_disciplina
- afisare_informatii_utilizator_id
- actualizeaza_nota_student
- afiseaza_catalog_student
- afiseaza_studenti_la_disciplina
- inscriere_student_la_disciplina
- vizualizare_studenti
- actualizare_utilizator
- sterge_utilizator_cu_cnp
- Trigger: insert_grupa_after_disciplina_insert

Procedurile dezvoltate de Cordea Maria și Apetrei Diana au fost integrate în cadrul proiectului pentru a susține funcționalitățile interfeței dezvoltate de Paucean Ioana, asigurând astfel coerența și funcționarea optimă a aplicației.

1.1.Descrierea Generală a Proiectului

Proiectul propus vizează dezvoltarea unei aplicații dedicate gestiunii unei platforme de studiu. Această aplicație va integra o interfață grafică intuitivă, care va permite utilizatorilor să acceseze rapid și eficient informațiile necesare pentru desfășurarea activităților academice. Structura aplicației va include module pentru gestionarea datelor referitoare la studenți, profesori, cursuri și activități didactice, asigurând astfel o organizare sistematică a informațiilor.

1.2.Scopul și Obiectivele

Scopul principal al acestui proiect este de a simplifica și eficientiza gestiunea unei platforme de studiu, care să răspundă nevoilor variate ale utilizatorilor facilitând interacțiunii dintre studenți, profesori și administrația universitară. Obiectivele specifice includ dezvoltarea unei baze de date relaționale, care să permită stocarea și gestionarea eficientă a informațiilor despre studenți, profesori, cursuri și alte informații relevante. Aplicația va oferi funcționalități de autentificare, gestionare a activităților didactice și interacțiune între studenți și profesori, contribuind astfel la o experiență educațională mai bine organizată.

2. Tema Proiectului

2.1. Prezentarea Aplicației

Aplicația propusă va utiliza un sistem de gestionare a bazelor de date MySQL, care va permite stocarea și manipularea eficientă a datelor. Interfața grafică va fi concepută pentru a oferi o experiență plăcută utilizatorilor, facilitând accesul rapid la informațiile necesare. Utilizatorii vor putea accesa aplicația printr-un proces de autentificare, care va asigura securitatea datelor. Acest proces de autentificare se va realiza prin intermediul e-mail-ului și al parolei.

2.2. Funcționalități și Utilizatori

Se va implementa un sistem de control al accesului bazat pe roluri, care va asigura că fiecare utilizator are acces doar la informațiile și funcționalitățile corespunzătoare rolului său.

Un utilizator poate avea următoarele roluri în sistem: Student, Profesor, Administrator, și Super-Administrator.

Rolurile și funcționalitățile utilizatorilor:

1. Student

- Căutarea cursurilor și înscrierea la acestea.
- Vizualizarea notelor obținute.
- Vizualizarea grupurilor de studiu și membrilor acestora.
- Trimiterea și citirea mesajelor pe grupuri.
- Vizualizarea și descărcarea activităților curente sau viitoare.
- Posibilitatea de a se înscrie sau de a părăsi un grup de studiu.
- Crearea activităților în grupurile de studiu, stabilind un număr minim de participanți și perioadele disponibile.
- Vizualizarea propriului profil fără a avea dreptul de a-l modifica.

2. Profesor

- Adăugarea și programarea activităților didactice, cum ar fi cursuri, seminare, laboratoare, colocvii și examene.
- Stabilirea ponderilor notelor pentru diferite tipuri de activități și notarea studenților.
- Vizualizarea listelor de studenți.
- Descărcarea cataloagelor.
- Vizualizarea și descărcarea activităților programate curente sau viitoare.

3. Administrator

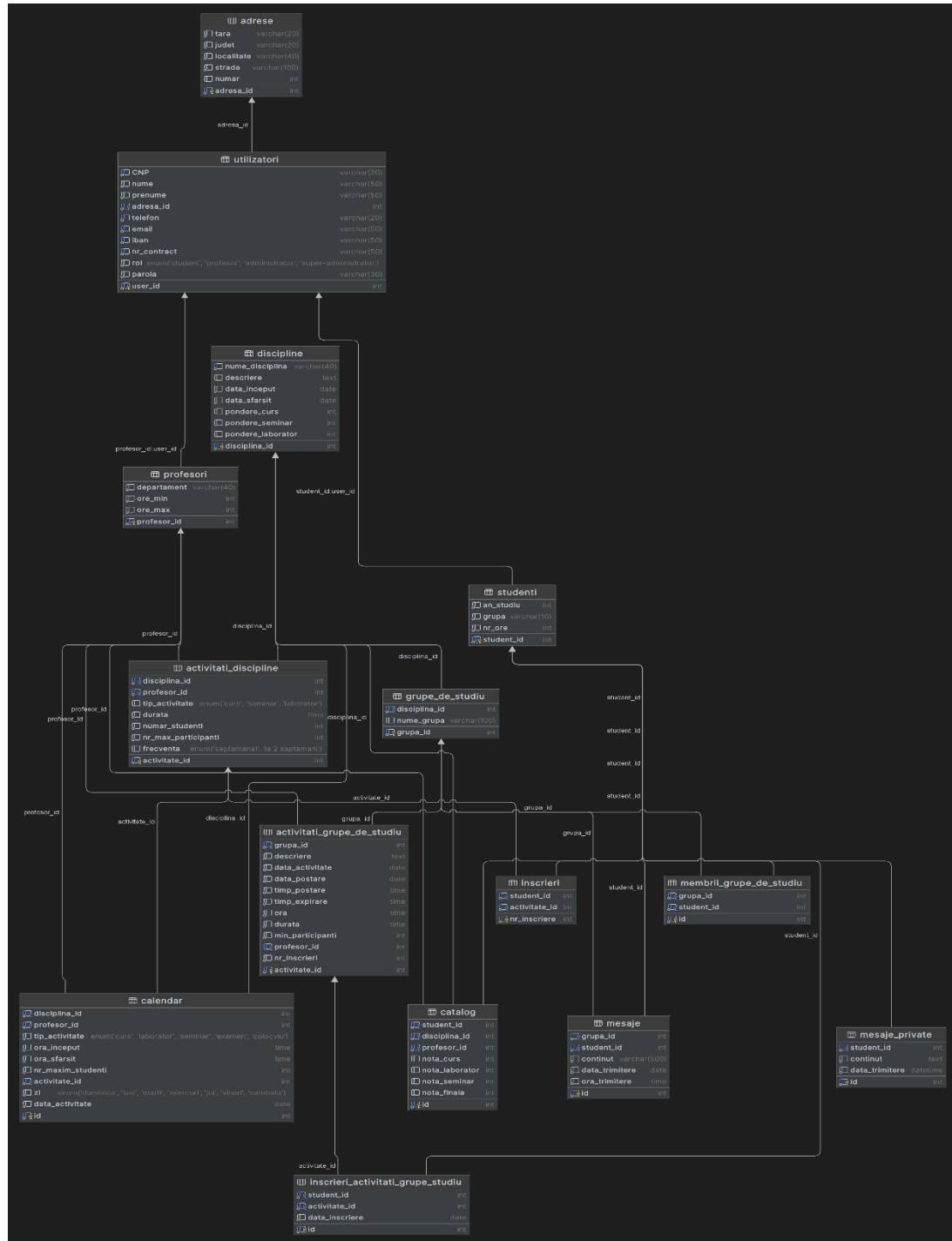
UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

- Adăugarea, modificarea și ștergerea informațiilor din baza de date legate de utilizatori.
 - Gestionarea utilizatorilor: căutare după nume și filtrare după tip.
 - Asignarea profesorilor la cursuri și gestionarea informațiilor despre acestea.
 - Vizualizarea tuturor studenților înscriși la un anumit curs.
4. Super-Administrator
- Gestionarea administratorilor: adăugare, modificare și ștergere a conturilor acestora.
 - Gestionarea utilizatorilor cu toate drepturile de administrator, dar cu un nivel superior de control asupra sistemului.

3. Descrierea Bazei de Date

3.1. Structura generala

Structura bazei de date este alcătuită din 15 tabele



UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

În structura bazei de date proiectate, tabelele sunt interconectate prin următoarele relații:

1. **Tabelul adrese** are o relație One-to-One cu tabela utilizatori prin coloana adresa_id.
2. **Tabelul utilizatori are relații:**
 - One-to-One cu tabela studenti prin coloana user_id.
 - One-to-One cu tabela profesori prin coloana user_id.
3. **Tabelul studenti are relații:**
 - One-to-Many cu tabela catalog prin coloana student_id.
 - Many-to-Many cu tabela activitati_discipline, relația fiind gestionată prin tabela inscrieri.
 - Many-to-Many cu tabela grupe_de_studiu, relația fiind gestionată prin tabela membri_grupe_de_studiu.
4. **Tabelul profesori are relații:**
 - Many-to-Many cu tabela discipline relația fiind gestionată prin tabela activitati_discipline
 - One-to-Many cu tabela calendar prin coloana profesor_id.
 - One-to-Many cu tabela activitati_discipline prin coloana profesor_id.
 - One-to-Many cu tabela catalog asociate prin coloana id_profesor
5. **Tabelul discipline are relații:**
 - One-to-Many cu tabela catalog prin coloana disciplina_id.
 - Many-to-Many cu tabela profesori relația fiind gestionată prin tabela
 - One-to-Many cu tabela activitati_discipline prin coloana disciplina_id.
 - One-to-Many cu tabela grupe_de_studiu prin coloana disciplina_id.
6. **Tabelul catalog are relații:**
 - One-to-One cu tabela studenti prin coloana student_id.
 - Many-to-One cu tabela discipline prin coloana disciplina_id.
 - Many-to-One cu tabela profesori asociate prin coloana id_profesor
7. **Tabelul grupe_de_studiu are relații:**
 - Many-to-Many cu tabela studenti, relația fiind gestionată prin tabela membri_grupe_de_studiu.
 - One-to-Many cu tabela activitati_grupe_de_studiu prin coloana grupa_id.
8. **Tabelul membri_grupe_de_studiu are relații:**
 - Many-to-One cu tabela grupe_de_studiu prin coloana grupa_id.
 - Many-to-One cu tabela studenti prin coloana student_id.
9. **Tabelul activitati_grupe_de_studiu are relații:**
 - Many-to-One cu tabela grupe_de_studiu prin coloana grupa_id.
 - Many-to-Many cu tabela studenti, relația fiind gestionată prin tabela inscrieri_activitati_grupe_studiu.

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

10. Tabelul inscrieri_activitati_grupe_studiu are relații:

Many-to-One cu tabela activitati_grupe_de_studiu prin coloana activitate_id.

Many-to-One cu tabela studenti prin coloana student_id.

11. Tabelul mesaje are relații:

Many-to-One cu tabela grupe_de_studiu prin coloana grupa_id.

Many-to-One cu tabela studenti prin coloana student_id.

12. Tabelul activitati_discipline are relații:

Many-to-One cu tabela discipline prin coloana disciplina_id.

Many-to-One cu tabela profesori prin coloana profesor_id.

Many-to-Many cu tabela studenti, relația fiind gestionată prin tabela inscrieri.

One-to-One cu tabela calendar prin coloana calendar_id

13. Tabelul calendar are relații:

Many-to-One cu tabela discipline prin coloana disciplina_id.

Many-to-One cu tabela profesori prin coloana profesor_id.

One-to-One cu tabela calendar prin coloana calendar_id

14. Tabelul inscrieri are relații:

Many-to-One cu tabela studenti prin coloana student_id.

Many-to-One cu tabela discipline prin coloana disciplina_id.

15. Tabelul mesaje_private are relații:

Many-to-One cu tabela utilizatori prin coloana id_destinatar.

Many-to-One cu tabela utilizatori prin coloana id_expeditor.

3.2.Detalii specifice

3.3.Nivelul de Normalizare

Analiza nivelului de normalizare pentru această schemă implică evaluarea structurii tabelelor și a relațiilor dintre ele. În toate tabelele, atributele iau valori atomice și scalare, fără liste sau seturi de date. De exemplu, în tabela utilizatori, atributele precum user_id, nume și email conțin câte o singură valoare pentru fiecare înregistrare. Astfel, toate relațiile respectă cerințele primei forme normale (FN1).

În cazul tabelelor cu chei primare simple, cum ar fi utilizatori, discipline sau grupe_de_studiu, nu există dependențe parțiale între atributele non-prime și cheia primară, ceea ce îndeplinește cerințele formei normale a doua (FN2). De exemplu, în tabela activitati_discipline, cheia primară id determină direct toate celelalte atribute, iar relațiile sunt gestionate prin chei externe, cum ar fi disciplina_id și profesor_id.

Nu există dependențe tranzitive între atributele non-prime și cheia primară în tabelele cu chei simple. De exemplu, în tabela studenti, cheia primară id determină direct toate atributele, cum ar fi nume, prenume sau grupa_id, fără dependențe intermediare. Astfel, toate tabelele din schema bazei de date respectă cerințele formei normale a treia (FN3).

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

În ceea ce privește forma normală Boyce-Codd (BCNF), schema bazei de date este conformă deoarece toate dependențele funcționale sunt determinate de cheile candidate. De exemplu, în tabela utilizatori, dacă CNP este unic și utilizat pentru a identifica utilizatorii, acesta poate fi considerat o cheie candidat, iar toate dependențele funcționale sunt gestionate corect.

4. Elemente de programare a functionalitatilor

4.1.Procedur.Vederi.Lista completa si exemple

Proceduri:

- **actualizare_utilizator(p_ID_utilizator, p_CNP, p_prenume, p_nume, p_adresa_id, p_telefon, p_email, p_iban, p_nr_contract, p_rol, p_parola)**
Permite actualizarea informațiilor unui utilizator (student, profesor, administrator, super-administrator) pe baza ID-ului utilizatorului, incluzând date personale, adresă, telefon, email, rol și parola.
- **adauga_mesaj_grup_studiu(student_id, nume_disciplina, mesaj)**
Permite unui student să adauge mesaje în chat-ul unui grup de studiu.
- **adauga_profesor_la_activitate(denumire, nume, prenume)**
Permite adăugarea unui profesor la o activitate de studiu, specificând denumirea activității și numele profesorului.
- **AdaugaActivitate(student_id, nume_disciplina, descriere, data_activitate, ora, durata, min_participanti, timp_expirare)**
Permite unui student să adauge o activitate în grupul de studiu, specificând detalii precum data, ora, durata, numărul minim de participanți și timpul de expirare.
- **afiseaza_activitati_studiu(student_id)**
Permite afișarea tuturor activităților din ziua curentă sau din viitor, din grupurile de studiu la care studentul este înscris și la care se poate alătura.
- **afiseaza_catalog_student(student_id_in)**
Returnează catalogul pentru un student specific, afișând disciplina, numele profesorului, notele pe activități (curs, seminar, laborator) și nota finală.
- **afisare_informatii_disciplina(nume_disciplina_in)**
Afișează informații despre o disciplină specifică, inclusiv profesorii care predau cursuri, seminare și laboratoare, precum și toate detaliile generale din tabelul discipline. Returnează o eroare dacă disciplina nu există.
- **afisare_informatii_utilizator_id(utilizator_id)**
Returnează detalii despre un utilizator specific, în funcție de rolul acestuia: pentru un profesor, afișează disciplinele și activitățile la care participă, pentru un student, afișează disciplinele la care este înscris și profesorii responsabili pentru fiecare tip de activitate. Generază o eroare dacă utilizatorul nu există.
- **afisare_studenti_curs(nume_disciplina)**
Permite unui administrator să afișeze toți studenții înscriși la un anumit curs.

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

- **actualizare_utilizator(p_ID_utilizator, p_CNP, p_prenume, p_nume, p_adresa_id, p_telefon, p_email, p_iban, p_nr_contract, p_rol, p_parola)**
Actualizează informațiile unui utilizator (student, profesor, administrator) pe baza ID-ului utilizatorului, incluzând date personale, adresă, telefon, email, rol și parola.
- **actualizeaza_nota_student(profesor_id_in, student_id_in, nume_disciplina_in, nota_curs_in, nota_seminar_in, nota_laborator_in)**
Actualizează notele unui student pentru o anumită disciplină (curs, seminar, laborator). Dacă studentul nu este înscris la o activitate a profesorului sau disciplina nu există, generează o eroare. Calculează nota finală pe baza ponderilor activităților.
- **cautare_dupa_nume_si_rol(nume, prenume, rol)**
Permite unui administrator să caute utilizatori pe baza numelui și să filtreze rezultatele după rol. Dacă rolul selectat este „fara rol,” sistemul returnează toți utilizatorii cu acel nume, indiferent de rol.
- **creare_grup_studiu(nume_disciplina, nume_grupa)**
Creează un grup de studiu pentru o anumită disciplină, asociat cu un nume specific de grup.
- **inscriere_activitate_studiu(student_id, descriere_activitate)**
Permite unui student să se înscrie la o activitate dintr-un grup de studiu, cu condiția să facă parte din grupul asociat disciplinei respective.
- **inscriere_student_grup_studiu(student_id, nume_disciplina)**
Permite înscrierea unui student într-un grup de studiu, verificând dacă acesta este deja înscris la disciplina respectivă.
- **inscriere_student_la_disciplina(student_id_in, nume_disciplina_in)**
Inscrie un student la cursul, seminarul și laboratorul unei discipline, în funcție de disponibilitate și fără suprapuneri de orar. Verifică existența disciplinei și actualizează automat tabelul catalog cu studentul și profesorul alocat.
- **inserare_administrator(p_CNP, p_prenume, p_nume, p_adresa_id, p_telefon, p_email, p_iban, p_nr_contract, p_parola)**
Adaugă un utilizator cu rolul de "administrator" în tabela utilizatori.
- **inserare_activitate_discipline(nume_disciplina_in, profesor_id, activitate_predata, durata, nr_max_participanti, frecventa)**
Adaugă o activitate specifică (curs, seminar, laborator) în cadrul unei discipline, asociată unui profesor. Verifică dacă disciplina există înainte de inserare. Dacă nu, generează o eroare.
- **inserare_disciplina(nume_disciplina, descriere, data_inceput, data_sfarsit)**
Adaugă o disciplină nouă în tabela discipline.
- **inserare_profesor(p_CNP, p_prenume, p_nume, p_adresa_id, p_telefon, p_email, p_iban, p_nr_contract, p_parola, p_departament, p_ore_min,**

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

p_ore_max)

Adaugă un utilizator cu rolul de "profesor" și îl asociază cu informații suplimentare din tabela profesori.

- **inserare_student(p_CNP, p_prenume, p_nume, p_adresa_id, p_telefon, p_email, p_iban, p_nr_contract, p_parola, p_an_studiu, p_grupa, p_nr_ore)**

Adaugă un utilizator cu rolul de "student" și îi asociază informații specifice din tabela studenti.

- **modifica_ponderi_disciplina(profesor_id_in, nume_disciplina_in, pondere_curs_in, pondere_seminar_in, pondere_laborator_in)**

Modifică ponderile procentuale ale activităților (curs, seminar, laborator) pentru o disciplină. Permite doar profesorului care predă cursul să modifice aceste valori și verifică dacă suma ponderilor este egală cu 100%.

- **parasire_student_grup_studiu(student_id, nume_disciplina)**

Permite unui student să părăsească un grup de studiu asociat unei discipline.

- **programare_activitate(profesor_id, nume_disciplina, tip_activitate, zi, ora_start)**

Permite unui profesor să programeze o activitate didactică (curs, seminar, laborator) în calendar, indicând ziua, ora de început și tipul activității.

- **programare_examen_colocviu(profesor_id, nume_disciplina, tip_activitate, data_activitate, ora_start, durata)**

Permite unui profesor să programeze un examen sau colocviu în calendar, specificând detalii precum data, ora de început și durata activității.

- **sugereaza_activitate_grup(student_id)**

Sugerează unei activități dintr-un grup de studiu la care studentul este înscris, luând în considerare orarul acestuia și activitățile disponibile.

- **sugestii_de_participanti(student_id)**

Afișează aleatoriu 4 studenți care sunt înscrși în grupuri de studiu în care studentul specificat (identificat prin student_id) nu este deja membru.

- **sterge_utilizator_cu_cnp(cnp_in)**

Șterge un utilizator pe baza CNP-ului, împreună cu toate informațiile și activitățile acestuia (în funcție de rolul utilizatorului, fie student, fie profesor), inclusiv înregistrările asociate în tabelele de activități și studenți. Se face actualizarea corespunzătoare a numărului de studenți din activitățile respective.

- **vizualizare_activitati_profesor(profesor_id, data_aleasa)**

Permite unui profesor să vizualizeze toate activitățile programate pentru ziua curentă sau pentru o dată viitoare specificată.

- **vizualizare_activitati_student(student_id, data_aleasa)**

Permite unui student să vizualizeze toate activitățile la care este înscris pentru ziua curentă sau pentru o dată viitoare specificată.

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

- **vizualizare_grupe_student(student_id)**
Permite unui student să vadă toate grupurile de studiu la care este înscris.
- **vizualizare_membri_grupa(student_id, nume_disciplina)**
Permite unui student să vadă lista completă a membrilor care fac parte dintr-un grup de studiu din care face și el parte.
- **vizualizare_mesaje_student_grup_studiu(student_id, nume_disciplina)**
Permite unui student să vizualizeze toate mesajele postate în chat-ul unui grup de studiu.
- **vizualizare_studenti(profesor_id_in, nume_disciplina_in, activitate_disciplina_in)**
Afișează lista studenților înscriși la o anumită activitate (curs, seminar, laborator) a unui profesor, împreună cu notele acestora (curs, seminar, laborator și nota finală).

4.2. Alte exemple de Cod SQL

Eveniment: VerificaActivitatiPeriodice

Acest eveniment verifică toate activitățile de studiu pentru a identifica cele al căror timp de expirare a trecut. Dacă, după expirarea timpului, activitatea nu a atins numărul minim de participanți necesari, aceasta este ștearsă automat. Înainte de ștergere, fiecare student înscris primește un mesaj privat care îl informează că activitatea a fost anulată.

Trigger: insert_grupa_after_disciplina_insert

Acest trigger este activat după ce o nouă disciplină este inserată în tabela discipline. Trigger-ul creează automat o nouă intrare în tabela grupe_de_studiu, asociind disciplina nou adăugată cu o grupă de studiu, care are un câmp nume_grupa setat la NULL.

5. Interfața Grafică a Utilizatorului (GUI)

Orice aplicație necesită o interfață grafică pentru a permite utilizatorilor să acceseze ușor informațiile și funcțiile disponibile. În acest proiect, s-a creat o interfață grafică în Java, folosind clase de obiecte care sunt alese în funcție de cerințele fiecărui domeniu de activitate, pentru a asigura o utilizare eficientă și intuitivă.

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

5.1. Detalii de implementare funcționalități specifice

Dintre clasele folosite le amintim cele mai importante:

BDconnection – stabilește conexiunea cu baza de date pentru pagina de log in și gestionează unde intri gen dacă ești student te loghează pe pagina de student, dacă ești admin te loghează pe pagina de admin, dacă ești profesor te loghează pe pagina de profesor și dacă ești superAdmin te loghează pe pagina de superadmini

Clasa BDconnection gestionează conexiunile la baza de date și logica pentru schimbarea scenelor în aplicația JavaFX. Este utilizată pentru autentificarea utilizatorilor și navigarea în funcție de rolul acestora. Iată o descriere succintă:

Funcționalități principale:

1. Schimbarea scenelor (changeScene*):

- Metodele changeSceneStudent, changeSceneProfesor, changeSceneAdmin, changeSceneSuperAdmin și changeScene sunt utilizate pentru a naviga între scenele aplicației.
- Acestea folosesc FXMLLoader pentru a încărca fișiere FXML corespunzătoare și pentru a configura informațiile utilizatorilor în controlerele respective.

2. Autentificarea utilizatorilor (loginUser):

- Conectează utilizatorii la platformă prin verificarea acreditivelor (email și parolă) din baza de date.
- Navighează către scenele corespunzătoare (student, profesor, administrator sau super-administrator) în funcție de rolul utilizatorului.

3. Verificarea conexiunii (verificareConexiune):

- Testează conexiunea la baza de date și verifică dacă un utilizator specificat există, afișând informațiile acestuia în consolă.

4. Acces la baza de date:

- Utilizează JDBC pentru a efectua operațiuni pe baza de date platforma_studii.
- Include gestionarea resurselor JDBC, cum ar fi Connection, PreparedStatement și ResultSet.

5. Gestionarea erorilor:

- Captură și afișare de excepții pentru erori legate de baza de date sau încărcarea scenelor.

Observații:

- Clasa este puternic centrată pe interacțiunea cu baza de date și gestionarea interfeței utilizatorului.
- Utilizează variabile precum width și height pentru a seta dimensiunile ferestrelor.
- Gestionarea resurselor JDBC este corectă, cu utilizarea blocurilor finally pentru închidere.

Această clasă este un element cheie al aplicației, asigurând conectivitatea cu baza de date și o experiență fluidă pentru utilizatori în funcție de rolurile lor

ProfilController – clasa care se ocupă de gestionarea tuturor funcționalităților pentru pagina de student. Aici sunt create funcții ce se ocupă cu tot ce poate studentul să facă (inscriere curs, cautare curs, descarcare orar, vizualizare note etc)

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Codul din fișier definește clasa ProfilController, care gestionează interfața și funcționalitățile profilului unui utilizator (student) într-o aplicație JavaFX conectată la o bază de date. Mai jos este o descriere succintă a funcționalităților implementate:

Funcționalități principale:

1. Gestionarea informațiilor utilizatorului:

- Metoda setUserInfo afișează informațiile personale ale studentului (nume, prenume, an de studiu, grupă, email etc.) pe interfața grafică.
- Metoda loadBonus afișează sugestii personalizate pentru activități bonus.

2. Gestionarea activităților și programului:

- loadActivitati încarcă activitățile programate pentru un student într-o zi selectată.
- afiseazaOre este legată de un buton pentru a afișa activitățile selectate în interfață.
- descarcaActivitati permite exportarea activităților într-un fișier CSV.

3. Gestionarea cursurilor:

- cautaCurs și cautaCursButton permit căutarea detaliilor despre un curs specific.
- inscriereCurs și inscriereCursFuncie gestionează înscrierea unui student la un curs.

4. Gestionarea grupelor de studiu:

- loadGrupeStudiu încarcă lista grupelor la care studentul este înscris.
- inscriereGrupaStudiu, veziMembriiGrupaStudiu și iesiGrup permit înscrierea, vizualizarea membrilor și ieșirea dintr-o grupă.

5. Gestionarea activităților de grup:

- loadTabelActivitatiStudiu încarcă activitățile grupului în care studentul este înscris.
- confirmareInscriereActivitate permite înscrierea studentului la o activitate.

6. Gestionarea sugestiilor:

- loadSugestii afișează sugestii de grupuri și participanți pentru student.

7. Export și afișare date:

- Exportul datelor în fișiere CSV.
- Configurarea și afișarea tabelelor pentru note, activități, grupuri și sugestii.

Observații:

- Clasa folosește multiple componente JavaFX, inclusiv Label, TextField, TableView, și Button.
- Operațiile cu baza de date sunt realizate prin JDBC, cu apeluri la proceduri stocate pentru manipularea datelor.
- Include tratamentul erorilor pentru interacțiunea cu baza de date și afișarea alertelor utilizatorului.

Această clasă este esențială pentru interfața utilizatorului și pentru conectarea la baza de date, oferind funcționalități extinse pentru managementul activităților și informațiilor unui student

Main.java

Aceasta este o clasă principală a unei aplicații JavaFX care inițializează și lansează interfața grafică a utilizatorului (GUI). Iată o descriere succintă a clasei:

1. Moștenirea clasei Application:

- Clasa extinde Application, ceea ce o face punctul de pornire pentru o aplicație JavaFX.

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

2. Metoda start(Stage stage):

- Este metoda principală care configurează și afișează scena principală a aplicației.
- Încarcă o interfață grafică definită într-un fișier FXML (log-in.fxml), folosind FXMLLoader.
- Setează titlul ferestrei principale ca fiind "Logheaza-te!!".
- Creează o scenă cu dimensiunile 685x400 și o atașează la fereastra principală.

3. Metoda main(String[] args):

- Este punctul de intrare al aplicației.
- Apelează metoda launch(), care inițializează ciclul de viață al aplicației JavaFX.

4. Comentariu în cod:

- Există un comentariu ce face referire la o conexiune la baza de date prin clasa BDconnection, dar această funcționalitate este momentan dezactivată. Această clasă este responsabilă cu inițializarea și afișarea ferestrei principale a aplicației, fiind concepută pentru a porni de la un ecran de autentificare (log-in.fxml).

AdminController – Gestionează funcționalitățile pentru administrator, inclusiv administrarea utilizatorilor, cursurilor și grupelor de studiu. Interacționează cu baza de date pentru manipularea datelor administrative.

SuperAdminController – Similar cu AdminController, dar cu privilegii extinse pentru super-administrator, inclusiv gestionarea conturilor de administrator și setările globale ale aplicației.

LogInController – Se ocupă cu autentificarea utilizatorilor, validând acreditările și redirectionându-i către pagina corespunzătoare în funcție de rol (student, profesor, admin, super-admin).

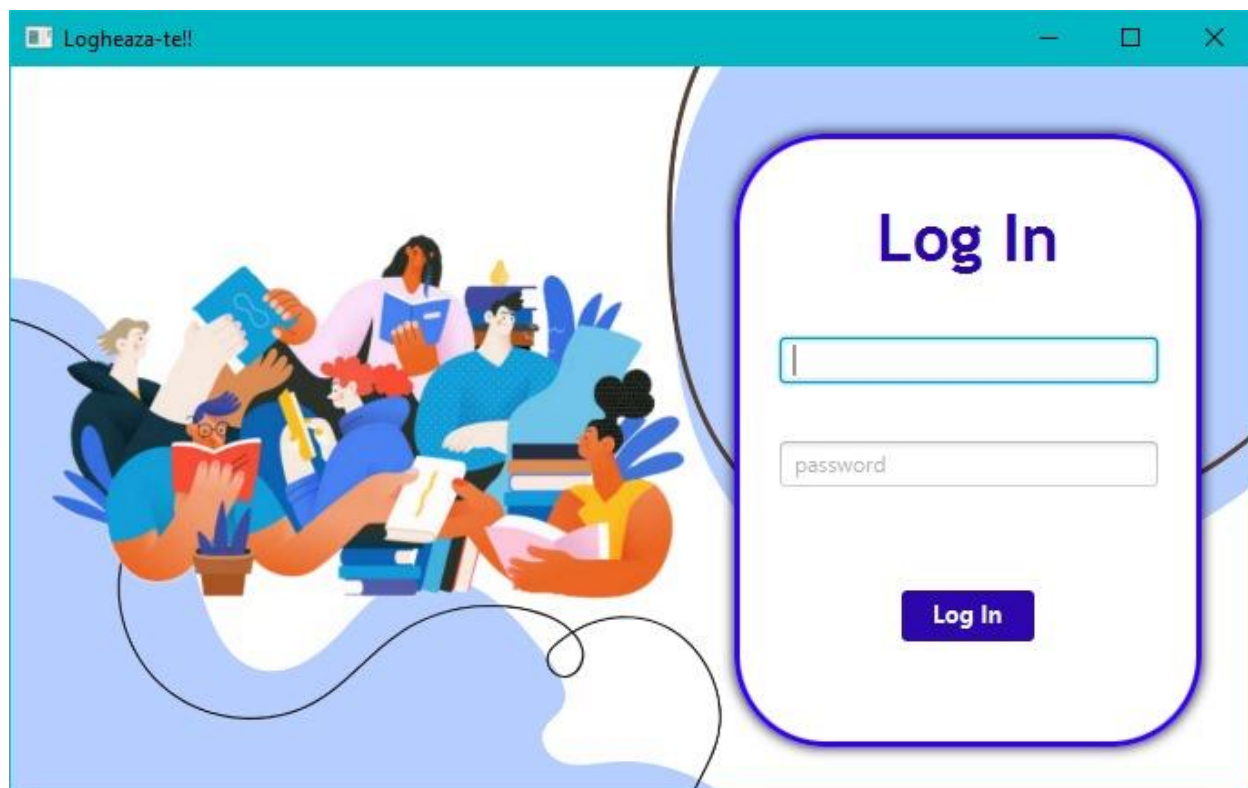
MesajeController – Gestionează funcționalitățile de chat, permițând trimiterea și primirea mesajelor între utilizatori, cu stocarea și vizualizarea acestora din baza de date, într-o interfață intuitivă.

VizualizareStudentiController – Permite administratorului să vizualizeze toți studenții înscriși la un curs selectat, afișând datele într-un tabel, interacționând cu baza de date pentru a obține informațiile necesare.

6. Manual de Utilizare

6.1.Înregistrare și Autentificare

Pentru ca un utilizator să se logheze, va trebui să introducă în aceste casete emailul și parola.



6.2. Gestionarea Informațiilor per utilizatori

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

În funcție de rolul acestuia, îi va apărea o interfață sugestivă, specifică fiecărui rol: student, profesor, administrator, super-administrator.

Interfața pentru Student:



UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Platforma Studii

PROFIL

ORAR

CURSURI

NOTE

GRUPE DE STUDIU

ACTIVITATI

ADauga O ACTIVITATE

SAU INSCRIE-TE...

Grupa	Descriere	Data	Ora
Grupa_Fizica	Activitate de studiu pentru Fizica	2025-04-28	10:00:00

Activitate de studiu pentru Fizica

CONFIRMARE INSCRIERE

Orarul tau e liber pentru aceasta activitate, poate te-ar interesa

Sugestie activitate: Activitate de studiu pentru Fizica (Grupa ID: 6, Data: 2025-04-28, Ora: 10:00, Durata: 02:00)

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Interfața pentru Profesor:

Platforma Studii

PROFIL

ORAR

CATALOG

MENIU

Vizualizare studenti

DESCARCA CATALOG

Analiza matematica I

curs

Afiseaza

id_student	NUME	PRENUME	CURS	SEMINAR	LABORATOR	NOTA FINALA
36	Popescu	Ion	9	2	0	7
37	Popa	Andreea	0	0	0	0
39	Nistor	Gabriela	0	0	0	0
63	Barbu	Daniel	0	0	0	0
64	Stoica	Diana	0	0	0	0
66	Radulescu	Oana	0	0	0	0
67	Voicu	Andrei	0	0	0	0

Modificare nota

- selecteaza un student din tabelul de mai sus si modifica-i nota

0

0

0

Modifica

nota curs

seminar

laborator

Platforma Studii

PROFIL

ORAR

CATALOG

MENIU

Logout

Ionescu Maria

- profesor

Departament: Matematica

Ore minime: 8 Ore maxime: 20

email: maria.ionescu@example.com

telefon: 0755123457

Nr. contract: C12346

Adresa: Romania, jud. Cluj, loc. Cluj-Napoca

str.Strada Memorandumului, nr. 8

CNP: 1234567890124

IBAN: RO49AAAA1B3100750000001

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Platforma Studii

PROFIL

ORAR

CATALOG

MENIU

Modificare ponderi

introdu numele disciplinei

pondere cursCURS

pondere seminarSEMINAR

pondere laboratorLABORATOR

MODIFICA

Stabilire activitate (curs, laborator, seminar)

programare examen/colocviu

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Interfața pentru Administrator:

Platforma Studii

PROFILUTILIZATORIDISCIPLIECURSURI

Cauta utilizator

user_id	Nume	Prenume	CNP	Adresa	telefon	email	IBAN	nr_contract
29	Badea	Claudia	1234567350214	214	0755000014	claudia.badea...	RO49AAHA18...	P12314
94	Badea	Claudia	1234567890191	151	0755163524	claudiaa.bad...	RO49AAA18...	C12418

29

MODIFICASTERGEADAUGA PROFESORADAUGA STUDENTASIGNARE ACTIVITATE

Platforma Studii

PROFILUTILIZATORIDISCIPLIECURSURI

Cordea Maria

- admin

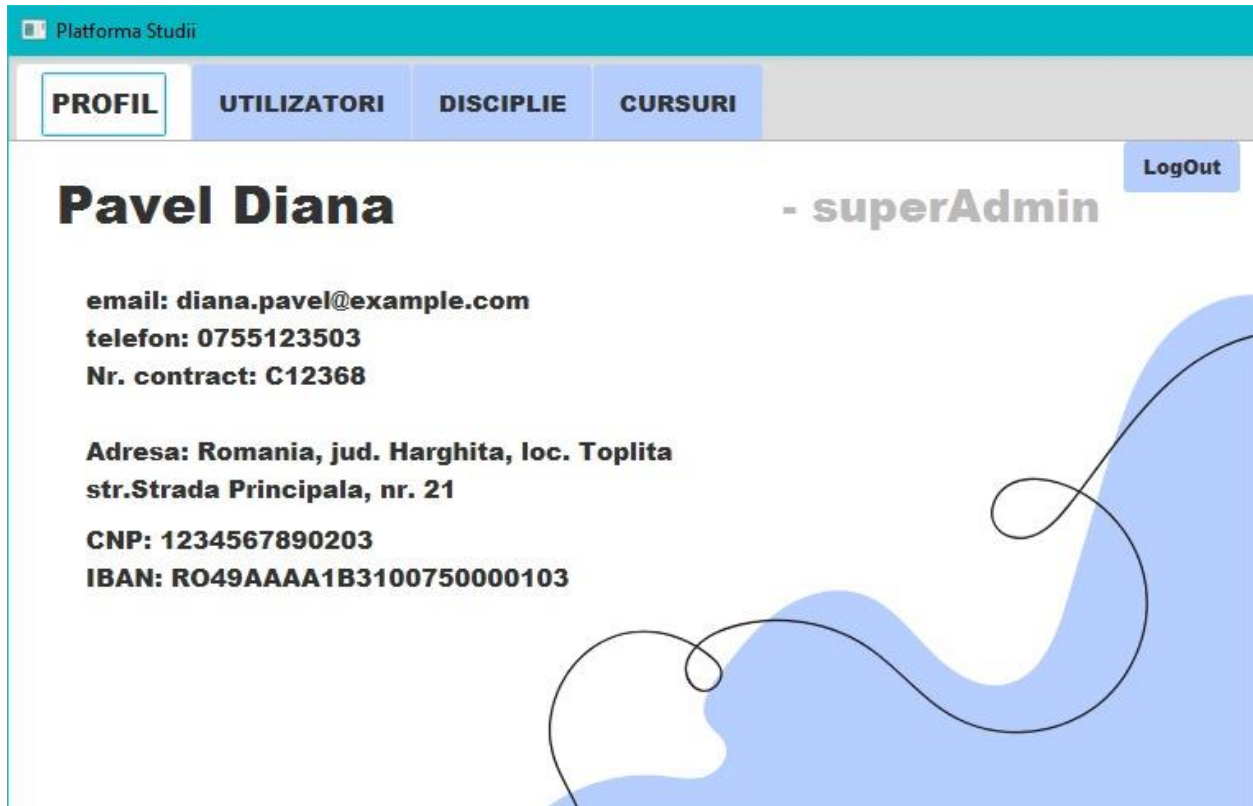
LogOut

email: maria.cordea@pisi.com
telefon: 0764773426
Nr. contract: BR7474

Adresa: Romania, jud. Cluj, loc. Cluj-Napoca
str.Strada Memorandumului, nr. 8
CNP: 7763527718933
IBAN: RO49BBBB1B3100750000014

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Interfața pentru Super Administrator



7. Concluzii și dezvoltari ulterioare

7.1. Analiza funcționalităților curente- cele mai interesante solutii adoptate

În cadrul bazei de date platforma_de_studiu, una dintre cele mai ingenioase metode adoptate a fost implementarea procedurilor stocate. Acestea sunt structuri de cod SQL care permit executarea unor operațiuni complexe direct în baza de date, fără a fi necesar ca logica să fie executată pe aplicația clientului. Procedurile au fost scrise astfel încât să poată fi apelate exclusiv din interfața aplicației, oferind mai multe avantaje:

- **Centralizarea logicii în baza de date**

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

Prin definirea procedurilor stocate, logica aplicației este centralizată în baza de date, asigurându-se astfel că regulile de business sunt respectate uniform. Aceste proceduri pot manipula datele într-un mod eficient și coerent, fără a depinde de logica implementată pe fiecare interfață a utilizatorului.

- **Reducerea complexității aplicației client**

Aplicația client nu trebuie să se preocupe de detaliile implementării logicii de afaceri. Procedurile stocate preiau această responsabilitate, ceea ce duce la o aplicație mai simplă și mai ușor de întreținut. Astfel, apelarea unui singur procedeu stocat poate înlocui mai multe interogări SQL complexe.

- **Optimizarea performanței**

Procedurile stocate sunt pre-compilate și optimizate de serverul de baze de date. Acest lucru reduce timpul de procesare al interogărilor și îmbunătățește performanța aplicației, în special când sunt necesare operațiuni multiple sau complexe asupra unui volum mare de date.

7.2.Dezvoltari ulterioare

O platformă de studii bazată pe o bază de date poate evolua continuu pentru a sprijini atât utilizatorii (studenți, profesori, administratori), cât și instituțiile educaționale. Dezvoltările ulterioare ar trebui să includă atât îmbunătățiri tehnologice cât și funcționalități care să îmbunătățească interacțiunea cu utilizatorii și să crească eficiența administrativă. Câteva idei pentru dezvoltarea ulterioară sunt:

- **Integrarea unui sistem de gestionare a conținutului (CMS)**

Pentru gestionarea materialelor educaționale:

- Un CMS integrat ar permite profesorilor și administratorilor să încarce și să administreze conținutul educațional direct din platformă (de exemplu, documente PDF, video, lecții interactive).
- Profesorii ar putea să își creeze propriile lecții și să le editeze în mod dinamic, iar materialele să fie corelate automat cu disciplinele și activitățile corespunzătoare.

- **Generarea automată de rapoarte și statistici:**

Implementarea unui sistem de generare automată a rapoartelor de performanță pentru profesori și studenți (de exemplu, rapoarte de progres, rezultate la examene, participare la activități).

- **Alocarea automată a resurselor educaționale:**

Să se implementeze un sistem care să aloce automat resursele educaționale necesare în funcție de activitățile planificate (de exemplu, să stabilească sălile de curs, să aloce echipamente pentru laboratoare, etc.).

- **Înscriere la examene și colocvii:**

O adăugire importantă ar fi posibilitatea ca studenții să se poată înscrie nu doar la cursuri, ci și la examene și colocvii. Acest proces ar include selectarea examenelor disponibile

UNIVERSITATEA TEHNICA DIN CLUJ-NAPOCA

pentru fiecare disciplină, cu posibilitatea de a vizualiza datele și locațiile acestora. Studenții ar putea, de asemenea, să se retragă de la examene până la o anumită dată limită.