

## **PROIECT la DISCIPLINA**

### **INTRODUCERE IN BAZE DE DATE –v3**

**An academic 2021-2022**

**Punctaj – 50/100 puncte**

**Termen predare: saptamana 14/semI**

În cadrul acestui proiect, se va realiza o aplicație cu interfață grafică care va oferi utilizatorilor funcționalități, corespunzător cu drepturile pe care aceștia le dețin în sistem, descrise prin intermediul unor roluri.

După implementarea proiectului, studentul va fi capabil să:

- proiecteze o bază de date relațională cu mai multe tabele normalizate, între care există relații de dependență, gestionând informațiile din cadrul acestora prin intermediul unei aplicații;
- dezvolte un sistem de control al drepturilor de acces al utilizatorilor în cadrul sistemului informatic pe baza rolurilor (RBAC – role-based access control);
- realizeze o interfață grafică pentru a facilita interacțiunea utilizatorului cu aplicația, procesând evenimentele pe care acesta le generează.

Cunoștințele necesare sunt:

- programarea în limbaj OOP;
- manipularea bazelor de date folosind MySQL;
- folosirea JDBC - API pentru SQL Connection;

Termene

Săptămâna10 – stabilirea echipei de 3 studenți, team leader și responsabilități, modelul UML și implementarea bazei de date, eventuale întrebări referitor la cerințele proiectului.  
Săptămâna11 – schiță de meniu/interfață, funcționalități pentru un utilizator (administrator/profesor/student) Săptămâna12 – logare în aplicație, implementare roluri

Săptămâna 14 – predare

Echipa de proiect

Echipa va fi formată din 3 studenți. Este acceptabil ca din echipă să facă parte studenți din grupe diferite, proiectul va fi prezentat de către team-lead, în cadrul grupei din care acesta face parte, dar este de dorit să se limiteze aceste situații.

## Enunț

Se dorește implementarea unui sistem informatic destinat gestiunii unei **structuri bancare**.

Aplicația va folosi un sistem de gestiune pentru baze de date MySQL, iar interacțiunea cu aceasta va fi realizată doar prin interfața grafică. Funcționalitățile pe care le va oferi programul vizează operații ce țin de gestiunea clienților, angajaților și administrarea operațiilor curente din cadrul unei bănci.

Aplicația va putea fi accesată, pe baza unui proces de autentificare, de către mai multe tipuri de utilizatori: *clienți*, *angajați*, *administratori*. Pentru fiecare tip de utilizator se vor reține informații precum CNP, nume, prenume, adresă, număr de telefon, email, cont IBAN, numărul de contract. Fiecare utilizator își va putea vizualiza datele personale imediat după ce va accesa sistemul informatic, fără a avea însă posibilitatea de a le modifica. Totodată, programul trebuie să ofere și o funcționalitate pentru deautentificare, prin care se revine la fereastra care solicită datele de acces, astfel încât și un alt utilizator să îl poată folosi ulterior, fără a fi necesară repornirea sa.

Utilizatorul de tip **administrator** poate adăuga, modifica și șterge informații în baza de date, informații legate de utilizatori.

Administratorii pot să caute utilizatorii după nume și îi pot filtra după tip, pot asigna angajații la sucursale și pot face căutare după tipul operațiunilor bancare. La căutarea unei operațiuni, se vor afișa toate datele despre operațiune și client și numele angajatului responsabil.

Pentru un utilizator de tip **angajat** se vor reține norma, salariul, sucursala la care lucrează, departamentul de care aparține. Fiecare departament are un set de activități pe care un angajat le poate realiza.

	HR	IT	Functionar
Activitati bancare	nu	nu	da
Modificare baza de date	nu	da	nu
Activitati legate de utilizatori	da	nu	Doar personale si clienti

Pentru un utilizator de tip **client** se va reține și adresa, data nașterii, sursa principală de venit și dacă poate sau nu realiza tranzacții online.

Aplicația va permite gestiunea cu ușurință a activităților bancare. Conturile pot fi curente sau de economii și pot fi deținute de unul sau mai mulți clienți. Un client nu poate să aibă mai mult de 5 conturi.

Toate comisiunile, dobânzile și posibilele taxe sau praguri sunt setate din interfața de către admin. De exemplu, pentru conturile de economii se scade o comision de 2% când este lichidat și se adaugă o dobândă de 5% când este creat. De asemenea există depozite de 1, 3 și 6 luni pentru care se adaugă o dobândă de 5%, 10% respective 15% când se îndeplinește perioada pentru care a fost făcut depozitul. Clientul își poate lichida depozitul, caz în care acestuia este sters și banii sunt adăugați în contul curent. Dacă suma introdusă în depozit este mai mare de 500000 lei, un angajat trebuie să aprobe deschiderea depozitului, iar un admin lichidarea lui.

Clientul poate solicita eliberarea unui card, care va fi conectat la contul curent, iar cererea sa trebuie să fie aprobată de un admin și de un angajat.

Clientul poate face transferuri bancare între către alți clienți, caz în care trebuie să menționeze numele titularului contului și IBAN-ul. Transferul între conturi are status de “CREATED” când este inițiat de client și se realizează doar când este aprobat de un angajat, atunci banii sunt transferați și statusul devine “SUCCESSUL”. Dacă suma

este prea mare, statusul devine “ERROR” si transferul este anulat. Daca IBAN-ul este către o alta banca, se percepe un comision de 1%. Pentru a afla daca banca este diferita, verificați substring din IBAN (ex: ROBTRL25235346 si ROING5363464764).

Clientul isi poate salva o lista de contacte favorite catre care sa faca transferuri mai usor, fara sa mai specifice IBAN-ul.

Clienții pot sa plătească facturi folosind un cont special, prin alegerea unui furnizor si introducerea codului facturii. Se va genera o factura cu detaliile tranzacției.

Angajații pot sa vada toate tranzacțiile si sa extraga rapoarte despre statusul lor (filtrate după data, status, angajat responsabil sau client). De asemenea, pot sa vadă toate conturile unui client.

Adminul poate efectua plata unor salarii specificând suma, un cont sursa si o lista de IBAN-uri destinatie.

La logare, clienții isi vad contul curent si un meniu principal din care pot naviga spre toate activitățile bancare, iar angajații vad o lista cu operațiunile care trebuie autorizate de ei.

Accesul la date se va face dupa urmatoarele reguli:

	<b>Client</b>	<b>Angajat</b>	<b>Admin</b>
<b>Date personale</b>	CRUD	CRUD	CRUD
<b>Date despre conturi si depozite personale</b>	CRUD	CRUD	CRUD
<b>Date despre alti utilizatori</b>	Nu	Doar vizualizare	CRUD
<b>Date statistice</b>	Nu	Da	Doar vizualizare
<b>Date de administrare (comisioane, curs valutar, adaugare tipuri de operatiuni)</b>	Doar vizualizare	Doar vizualizare	CRUD

Se va implementa un sistem de notificari care la logarea in aplicatie sa semnaleze ultimele actualizari dupa cum urmeaza:

	<b>Client</b>	<b>Angajat</b>	<b>Admin</b>
<b>Date personale modificate</b>	da	nu	nu
<b>Paraguri / comisioane modificate</b>	nu	da	nu
<b>Sold modificat</b>	da	nu	nu
<b>Status tranzactie modificat</b>	da	da	nu
<b>Card emis</b>	da	nu	nu
<b>Structura BD modificata</b>	nu	nu	da

Notificarile vor fi salvate in baza de date si afisate pe paginile utilizatorilor corespunzatori si vor si marcate ca si vazute dupa ce au fost afisate o data.

### **Precizări suplimentare**

Popularea bazei de date cu informații trebuie să fie corespunzătoare pentru a putea ilustra funcționalitățile implementate.

Orice specificație care nu este menționată mai sus reprezintă decizie de implementare. Puteți considera orice simplificare în condițiile în care enunțul nu precizează altfel.

Folosiți proceduri pentru operațiile CRUD și trigger pentru verificarea constrângerilor.

### Barem de corectare și notare

Punctaj (p)	Criterii de acordare
10 p	<p>proiectarea tabelor în baza de date</p> <ul style="list-style-type: none"> <li>conformitatea structurii conceptuale a bazei de date cu o formă normală</li> <li>definirea de chei primare, chei străine (legături între toate tabelele bazei de date), constrângeri de integritate pentru formatul atributelor:</li> </ul> <p>populare corespunzătoare</p>
5p	<p>gestiunea utilizatorilor</p> <ul style="list-style-type: none"> <li>autentificare, deautentificare</li> <li>operații utilizator de tip administrator</li> <li>implementare sistem de control al drepturilor de acces al utilizatorilor în cadrul sistemului informatic pe baza rolurilor</li> <li>deschidere cont</li> <li>cautare utilizatori</li> </ul>
5p	<p>funcționalități pentru utilizatorul de tip client</p> <ul style="list-style-type: none"> <li>deschidere cont / depozit</li> <li>lichidare cont / depozit</li> <li>solicitare card</li> <li>vizualizare conturi și depozite</li> <li>plata facturi</li> <li>transfer între conturi</li> </ul>
5p	<p>funcționalități pentru utilizatorul de tip angajat</p> <ul style="list-style-type: none"> <li>aprobare depozite</li> <li>vizualizare tranzacții</li> <li>vizualizare clienți și conturi</li> </ul>
5p	<p>funcționalități pentru gestiunea utilizatorilor și a conturilor</p> <ul style="list-style-type: none"> <li>operații CRUD pe utilizatori</li> <li>operații CRUD pe conturi / depozite</li> </ul>
5p	modularizare :structura aplicației, lizibilitatea codului, comentarii, README
5p	validarea datelor :asigurarea consistenței informațiilor reținute în tabele: mesaje de eroare
10p	Realizarea interfeței grafice

## **BONUS**

(10p) Predarea cu o saptamana mai devreme

(10p) Dezvoltarea unui mecanism de versionare care să faciliteze tranziția de la o versiune a bazei de date la alta.

Salvati operatiile care au adus modificari asupra bazei de date si efectuati inversul operatiei pentru a readuce baza de date la starea anterioara.

De asemenea, scrieți o altă procedură stocată ce primește ca parametru un număr de versiune și aduce baza de date la versiunea respectivă.