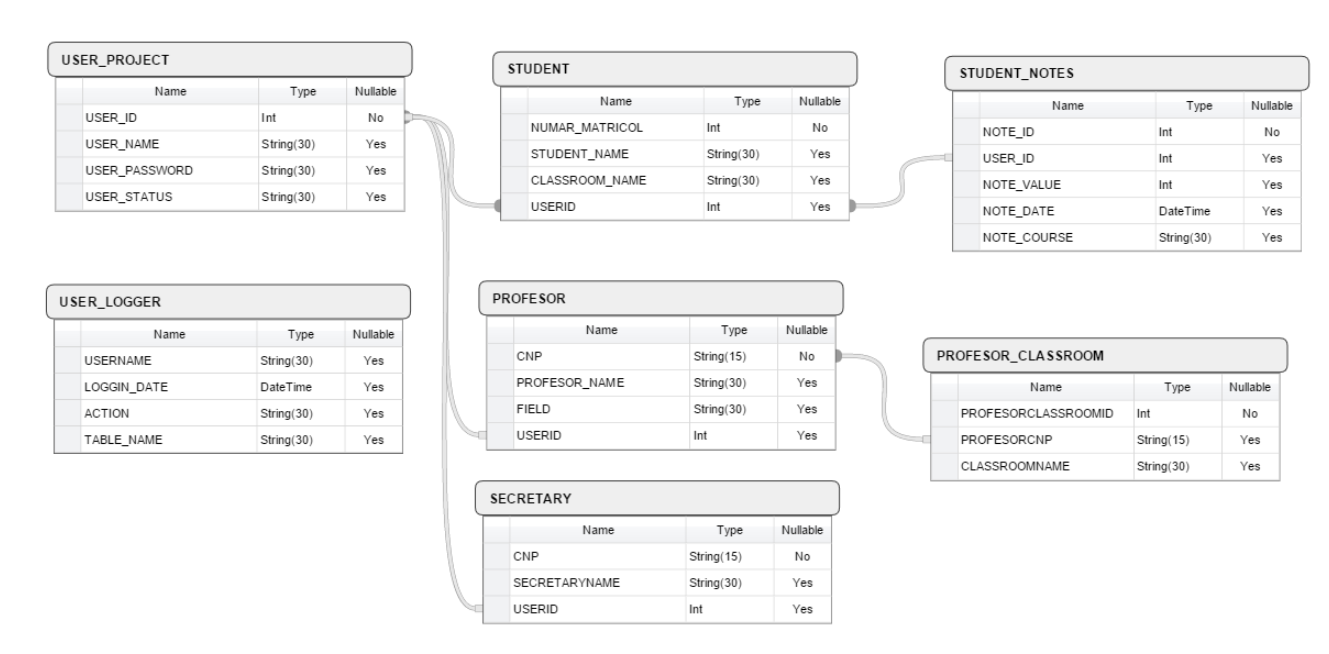
Catalog electronic pentru liceu

Student: Silaghi Alexandru-Mihail

Grupa: 342C5

Proiectul isi propune centralizarea intr-o baza de date a situatiei scolare (notele) ale unor elevi de liceu. Se pleaca de la premisa ca fiecare profesor preda o singura materie si un elev poate apartine unei singure clase. Entitatile participante sunt : entitatea elev, profesor si secretar. Elevul nu poate modifica nici-o nota, are atribut de read only , isi poate doar vedea situatia scolara. Profesorul introduce note dar doar la elevilor sai si de asemenea poate vedea situatia lor scolara a propriilor elevi doar la materia predata de ei. Secretara introduce useri in baza de date (elevi , profesori sau alte secretare) cat si leaga un profesor de clasa la care preda. Clasele nu sunt entitati in sine sunt reprezentate doar prin nume acesta fiind de forma numar legat de o litera reprezentativa ( 9A, 12B, 11C).

**Diagrama bazei de date**.



a

**Structura tabelelor.**

Structura tabelei **USER\_PROJECT**

Name Null Type

------------- -------- ------------

USER\_ID NOT NULL NUMBER

USER\_NAME VARCHAR2(30)

USER\_PASSWORD VARCHAR2(30)

USER\_STATUS VARCHAR2(30)

Structura tabelei **STUDENT**

Name Null Type

-------------- -------- ------------

NUMAR\_MATRICOL NOT NULL NUMBER

STUDENT\_NAME VARCHAR2(30)

CLASSROOM\_NAME VARCHAR2(30)

USERID NUMBER

Structura tabelei **PROFESOR**

Name Null Type

------------- -------- ------------

CNP NOT NULL VARCHAR2(15)

PROFESOR\_NAME VARCHAR2(30)

FIELD VARCHAR2(30)

USERID NUMBER

Structura tabelei **SECRETARY**

Name Null Type

------------- -------- ------------

CNP NOT NULL VARCHAR2(30)

SECRETARYNAME VARCHAR2(30)

USERID NUMBER

Structura tabelei **PROFESOR\_CLASSROOM**

Name Null Type

------------------- ---- ------------

PROFESORCLASSROOMID NUMBER

PROFESORCNP VARCHAR2(15)

CLASSROOMNAME VARCHAR2(10)

Structura tabelei **STUDENT\_NOTES**

Name Null Type

----------- -------- ------------

NOTE\_ID NOT NULL NUMBER

USER\_ID NUMBER

NOTE\_VALUE NUMBER

NOTE\_DATE DATE

NOTE\_COURSE VARCHAR2(30)

**Constrangeri de integritate**

Structura tabelara este sub forma de arbore. Tabela principala este tabela USER\_PROJECT care determina validitatea logarii si statul userului. Cheia primara a acestei tabele este USER\_ID, ea fiind foreign key pentru tabelele STUDENT, PROFESOR si SECRETARY. Legatura dintre tabela STUDENT si tabela STUDENT\_NOTES unde se tin notele fiecarui elev se face prin USER\_ID. Legatura dintre tabela PROFESOR si PROFESOR\_CLASSROOM se face prin coloana PROFESORCNP. Pentru generarea id-urilor se foloseste cate o secventa iar la inserarea in tabelele USER\_PROJECT , PROFESOR, SECRETARY, STUDENT\_NOTES si STUDENT se insereaza in tabela USER\_LOGGER actiunea facuta, data si tabela pe care s-a facut actiunea

**Descrierea functiilor si procedurilor**

Procedura getStudentDetails : Pe baza user\_id-ului va extrage din tabela STUDENT numele studentului, numele clasei in care acesta invata si numarul matricol.

Procedura getStudentNotes : Pe baza numarului matricol aceasta procedura scoate sub forma unui SYS\_REFCURSOR notele studentului.

Procedura getAuthValue : Pe baza username-ului si parolei aceasta procedura valideaza autentificarea din interfata si intoarce statutul userului (student, profesor sau secretara).

Procedura getProfesorDetails : Pe baza user\_id-ului aceasta procedura aduce din baza numele profesorului, materia la care acesta preda si CNP-ul.

Functia getProfesorStudents : Pe baza CNP-ului functia intoarce un SYS\_REFCURSOR cu toti elevii la care profesorul preda.

Procedura addNote : Adauga nota in tabela STUDENT\_NOTES.

Procedura deleteNote : Sterge nota din tabela STUDENT\_NOTES.

Procedura getSecretaryData : Intoarce pe baza user\_id-ului numele secretarei si CNP-ul

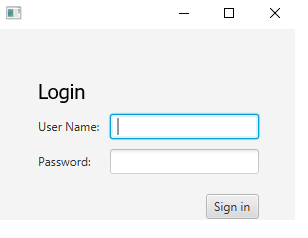
Functiile addStudent, deleteStudent, addProfesor,delProfesor,addSecretary, delSecretary, addProfClass, delProfClass sunt functii folosite de interfata secretarei prin care se insereaza sau sterg elevii, profesorii sau secretarele . Aceste functii au ca parametrii de intrare la inserare numele entitatii , cheia primara (nr matricol pentru elevi si cnp pentru profesori si secretare) , username-ul si parola entitatii. Pentru stergere se cere doar cheia primara. Functiile intorc in caz de succes 0 iar in caz de esec -1. S-a preferat notatia numerica deoarece oracle jdbc nu sustine valori booleene nule.

**Conectarea la baza de date**

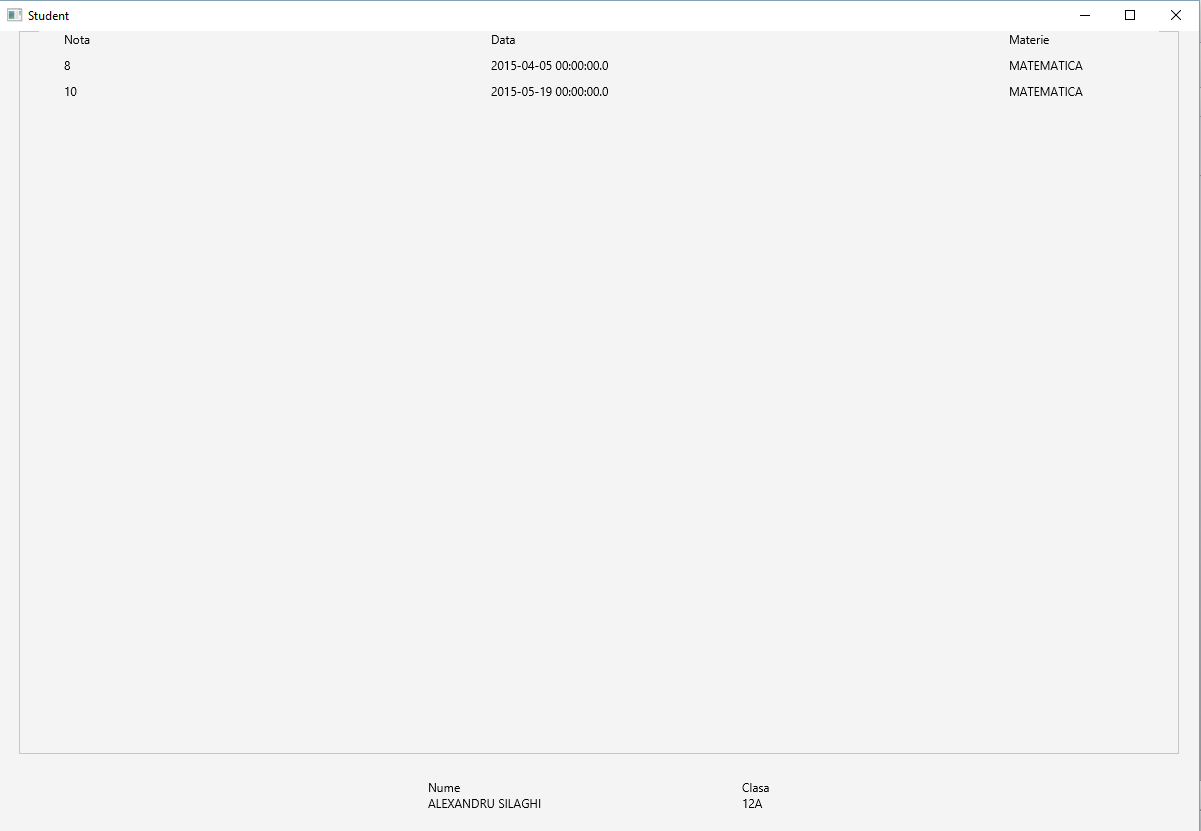
Conexiunea la baza de date se face prin JDBC , cu ajutorul librariei ojdbc7. Pentru conectare am folosit un user special creat ,i-am dat grant de creare sesiune si creare tabele. Pe baza unui url : jdbc:oracle:thin:c##proiectubd/123456@localhost:1521/orcl in care se regasesc : username-ul/parola@nume\_host:port/nume\_serviciu. Pe baza acestui url , OracleDataSource creeaza o conexiune ce poate fi preluata. In clasa JavaDBConnection se afla conexiunea si metodele ce apeleaza baza de date. Pentru preluarea unor date din baza sau efecturarea de operatiuni asupra bazei se instantieaza clasa de mai sus si se apeleaza metodele ei.

Snapshots :

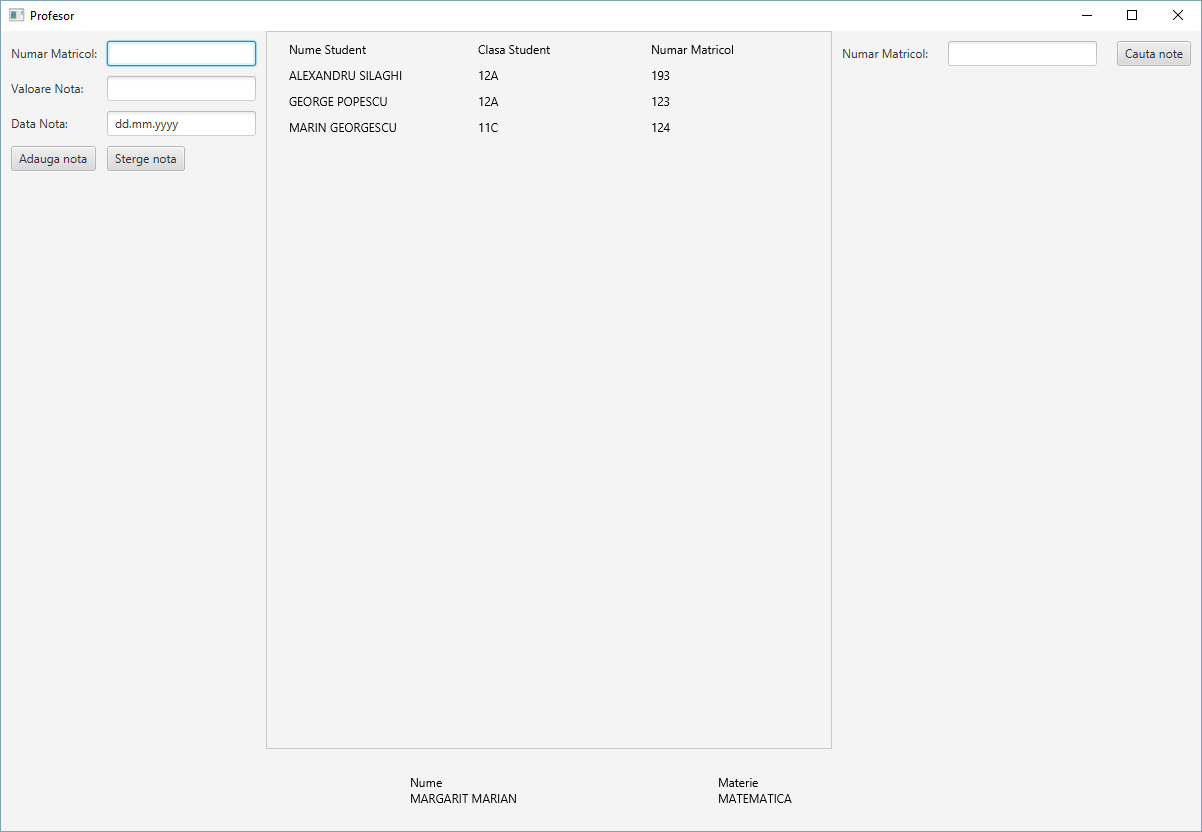
Interfata de logare :



Interfata student :



Interfata Profesor



Interfata Secretara

