**PROIECT**

la disciplina

Introducere in Baze de Date

**Gestiunea unei biblioteci**

Bendre Roxana, Moldovan Teodora

An academic :2017 – 2018

**Cuprins**

**1.Introducere**

* **Tema proiectului**
* **Specificație proiect**

**2.Modelul de date și descrierea acestuia**

* **Tabele**
* **Normalizarea datelor**
* **Diagrama EER**

**3.Detalii de implementare**

* **Interogări**
* **Obiecte de tip vedere**
* **Triggere**
* **Proceduri stocate**
* **Users**

**4.Interfata Java**

**5.Concluzii di dezvoltari ulterioare**

**1.Introducere**

**Tema proiectului**

Proiectul constă în realizarea unei baze de date cu ajutorul căreia să se gestioneze o bibliotecă, cu elementele necesare: membri, angajați, cărți, săli de studiu și evenimente, comenzi într-o manieră organizată și automatizată. Pentru a facilita accesul la baza de date, aceasta este insotita de o interfata grafica.

**Specificație proiect**

Proiectul constă într-o bază de date implementată în MySQL Worbench și o interfață grafica realizata în Eclipse Java Oxygen care oferă membriilor bibliotecii posibilitatea de a vizualiza cartile, operele, informatii despre alti membri precum si sa imprumute, sa faca rezervari pentru anumite carti, sa returneze carti si sa anuleze rezervari, sa vada informatii despre saliile de studiu si locurile disponibile in acestea si sa vada cand au loc evenimente in cadrul bibliotecii. Pe de alta parte, angajatii bibliotecii pot vizualiza toate aceste informatii, dar nu pot imprumuta carti sau sa faca rezervari. In schimb, acestia pot adauga membrii in saliile de studiu, deoarece fiecare membru este inregistrat atunci cand intra intr-o sala si pot adauga opere noi.

Baza de date este creata astfel incat sa redea cat mai fidel modul de lucru real al unei biblioteci.

* Fiecare carte din biblioteca este unic identificata prin ISBN si poate sa fie sub forma tiparita sau de manuscris. Pentru cele tiparite este important sa se cunoasca editura, iar pentru manuscrise anul in care au fost redactate. Despre carti de cunosc limba in care este scrisa, numarul total de copii, numarul de copii disponibile, categoria din care face parte, statusul(daca este disponibila,rezervata sau imprumutata), locatia acesteia in biblioteca si tariful de penallizare pe o zi, inn caz de intarziere la returnare.
* Biblioteca tine o evidenta a operelor in baza de date. Acestea nu exista fizica neaparat, dar o carte nu poate exista daca nu exista si opera asociata acesteia. Despre fiecare opera se cunosc titlul si autorul.
* Membrii bibliotecii sunt identificati printr-un ID unic. Despre acestia se cunoaste numele,prenumele, adresa, ocupatia, data nasterii, sexul si adresa de email. Adresa de email este utilizata pentru ca un membru sa poata intra in aplicatie. Membrii pot sa imprumute cari,sa le rezerve, sa le restituie si sa anuleze rezervarile facute. Se retin de asemenea informatii despre imprumuturile curente, istoricul de imprumuturi si rezervari. Pentru fiecare dintre acestea se cunosc dataile importante(inceput,final,data returnarii) .
* Biblioteca dispune de sali de studiu in care membrii pot sa vina sa invete. Accesul membriilor in sala de studiu este inregistrat de catre un angajat si se pune in evidenta data accesului in sala. Pentru fiecare sala de studiu este inregistrat numarul de locuri disponibile, numar care variaza in functie de membrii care sunt in sala.
* Fiecare angajat al bibliotecii are un ID unic, nume , prenume si se cunoaste data angajarii acestuia.
* Biblioteca organizeaza evenimente din diverse domenii la care membrii pot participa.

**2.Modelul de date și descrierea acestuia**

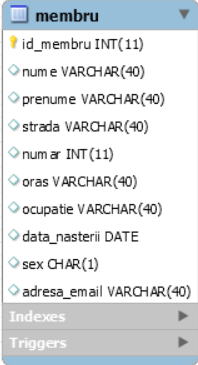
Descrierea bazei de date relaționale

Tabele

*Tabela membru:*

*-*conține informațiile despre fiecare membru înscris la bibliotecă.

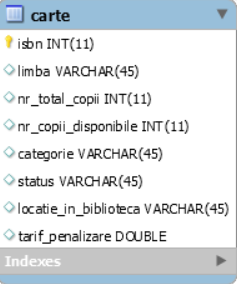
1. Id\_membru – cheie primară
2. Nume – numele membrului
3. Prenume – prenumele membrului
4. Strada – adresa la care locuiește membrul
5. Numar – adresa la care locuiește membrul
6. Oras - orașul în care s-a născut membrul
7. Ocupatie – cu ce se ocupă fiecare membru
8. Data\_nasterii – data în care s-a născut membrul, determină vârsta unui membru
9. Sex – sexul unui membru, feminin sau masculin
10. Adresa\_email – email-ul fiecarui membru



*Tabela carte:*

-conține informațiile despre fiecare carte.

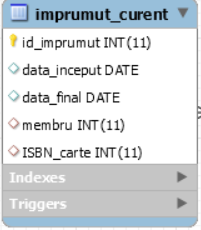
1. ISBN – cheie primară, reprezintă codul de identificare al cărților
2. Limba – limba în care este scrisă cartea
3. Nr\_total\_copii - numărul de copii pentru fiecare carte
4. Nr\_copii\_disponibile - numărul de copii disponibile în bibliotecă
5. Categorie – genul de literatură
6. Status - dacă este disponibilă sau nu în bibliotecă
7. Locatie\_in\_biblioteca - în ce zonă se află cartea
8. Tarif\_penalizare – este tariful aplicat în cazul în care nu se returnează cartea la timp



*Tabela imprumut\_curent:*

-conține informațiile necesare pentru împrumutarea curentă a unei cărți.

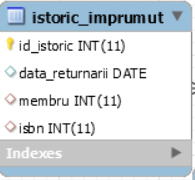
1. Id\_imprumut – cheie primară
2. Data\_inceput – data la care se împrumută cartea
3. Data\_final – data până la care este împrumutată cartea
4. Membru – cheie străină din tabela Membru
5. Isbn\_carte – cheie străină din tabela Carte



*Tabela istoric\_imprumut:*

-conține datele cărților împrumutate.

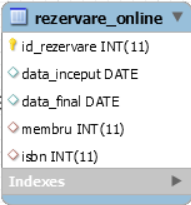
1. Id\_istoric – cheie primară
2. Data\_returnarii – data la care s-a returnat cartea
3. Membru – cheie străină din tabela Membru
4. Isbn – cheie străină din tabela Carte



*Tabela rezervare\_online:*

-conține informațiile cu privire la rezervarea online a unei cărți de către un membru al bibliotecii.

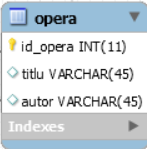
1. Id\_rezervare – cheie primară
2. Data\_inceput – data din care se rezervă cartea
3. Data\_final – data până la care trebuie returnată
4. Membru – cheie străină din tabela Membru
5. Isbn – cheie străină din tabela Carte



*Tabela opera:*

-conține informațiile unei opere.

1. Id\_opera – cheie primară
2. Titlu – titlul operei
3. Autor – autorul operei

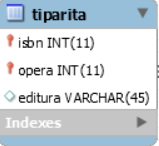


*Tabela tiparita:*

-este de fapt rezultatul unei relații M:N între tabelele ”carte” și ”opera”.

-conține informațiile despre cărțile tipărite.

1. Isbn – cheie străină din tabela Carte
2. Opera – cheie străină din tabela Opera
3. Editura – numele editurii cărții

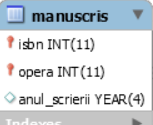


*Tabela manuscris:*

-este de fapt rezultatul unei relații M:N între tabelele ”carte” și ”opera”.

-conține informațiile despre manuscrise.

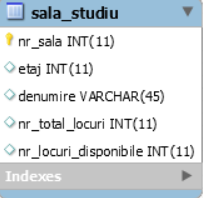
1. Isbn – cheie străină din tabela Carte
2. Opera – cheie străină din tabela Opera
3. Anul\_scrierii – anul în care este scris manuscrisul



*Tabela sala\_studiu:*

-conține informațiile necesare despre fiecare sală de studiu din cadrul bibliotecii.

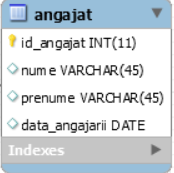
1. Nr\_sala – cheie primară
2. Etaj – etajul unde se află sala
3. Denumire – denumirea sălii
4. Nr\_total\_locuri - numărul de locuri din sală
5. Nr\_locuri\_disponibile - numărul de locuri disponibile din sală



*Tabela angajat:*

-conține informațiile despre angajații bibliotecii.

1. Id\_angajat – cheie primară
2. Nume – numele angajatului
3. Prenume – prenumele angajatului
4. Data\_angajarii – data la care s-a angajat

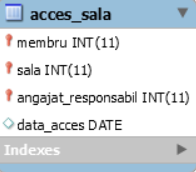


*Tabela acces\_sala:*

-este de fapt rezultatul unei relații M:N între tabelele ”membru” și ”sala\_studiu” și rezultatul unei relații M:N între tabelele ”membru” și ”angajat”.

-conține informațiile despre accesul în sală.

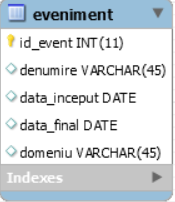
1. Membru – cheie străină din tabela Membru
2. Sala – cheie străină din tabela Sala\_studiu
3. Angajat\_responsabil – cheie străină din tabela Angajat
4. Data\_acces – data în care ai acces la sală



*Tabela eveniment:*

-conține informațiile cu privire la evenimentele din cadrul bibliotecii.

1. Id\_even – cheie primară
2. Denumire – denumirea evenimentului
3. Data\_inceput – data la care începe un eveniment
4. Data\_final – data la care se sfârșește un eveniment
5. Domeniu – domeniul evenimentului

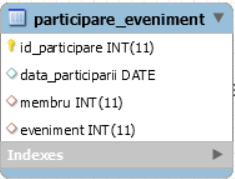


*Tabela participare\_eveniment:*

-este de fapt rezultatul unei relații M:N între tabelele ”membru” și ”eveniment”.

-conține informațiile despre participarea la evenimente.

1. Id\_participare – cheie primară
2. Data\_participarii – data la care se participă la eveniment
3. Membru – cheie străină din tabela Membru
4. Eveniment – cheie străină din tabela Eveniment



Normalizarea datelor

Baza de date se aﬂa in nivelul de normalizare Boyce-Codd. Conform cercetatorului Chris Date, acest nivel de normalizare poate ﬁ rezumat astfel: ”(1)Fiecare atribut trebuie (2)sa reprezinte o informatie (3)despre cheie, (4)intreaga cheie, (5)si nimic in afara cheii”. In cazul bazei noastre de date:

1. Fiecare atribut este atomic si nu depinde de alte atribute

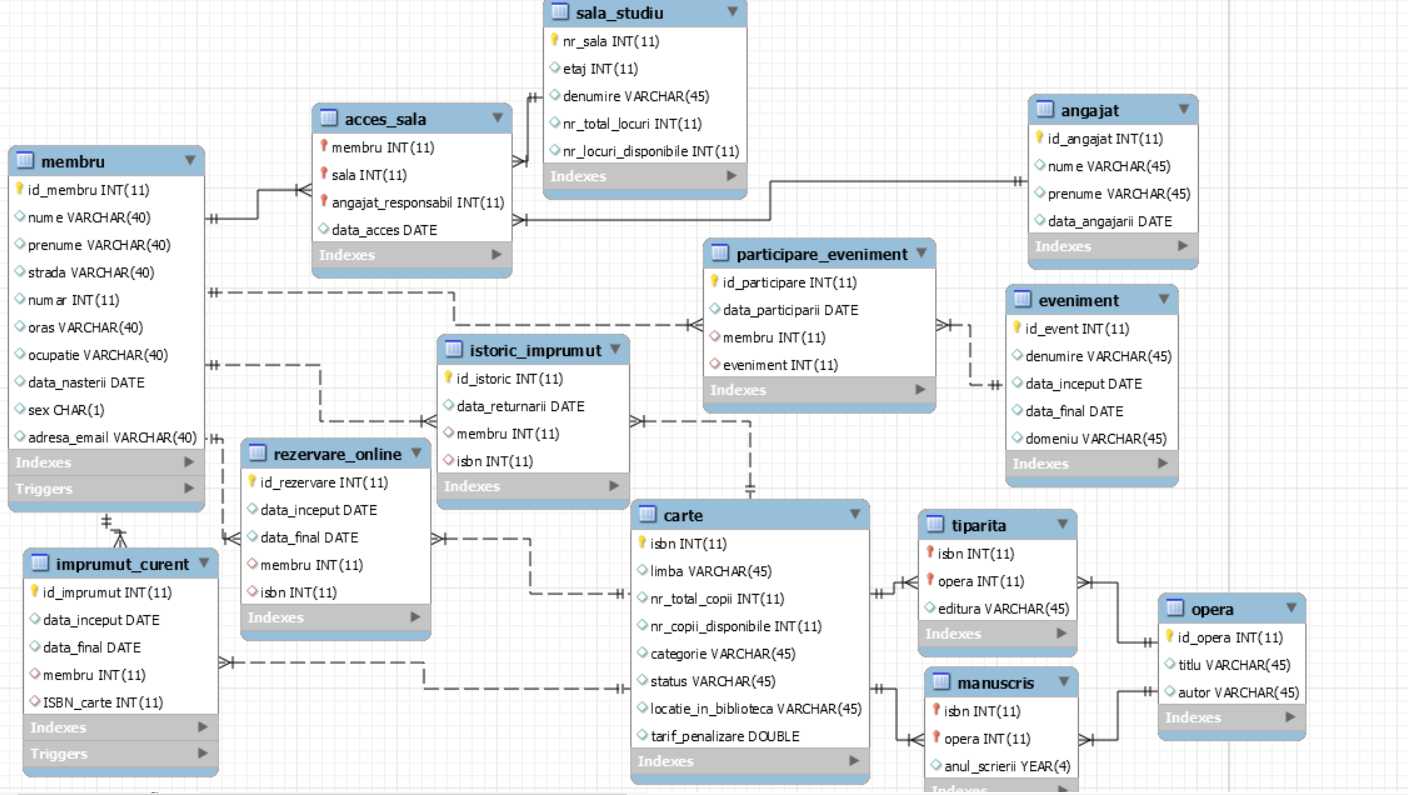
2. Fiecare atribut reprezinta o singura informatie

3. Exista atribute chei, reprezentate de id-ul unic al tabelelor

4. Cheia primara este suﬁcienta ca sa determine celelalte atribute ale tabelelor

5. Cheia primara este singurul atribut care determina celelalte atribute, nu exista un alt atribut care poate determina in mod unic o tupla.

Diagrama EER



**3.Detalii de implementare**

**MySQL**

*Interogări*

În continuare se prezintă o serie de interogări ce demonstrează cum se foloseste bază de date:

1. /\* numărul fetelor înscrise la bibliotecă \*/

select count(\*) from membru where sex='F';

1. /\* anul celui mai vechi manuscris \*/

select min(manuscris.anul\_scrierii) as 'Anul manuscris vechi'

from manuscris,carte,opera

where opera.id\_opera=manuscris.opera and carte.isbn=manuscris.isbn;

1. /\* titlul cărților din Zona de literatură \*/

select distinct opera.titlu

from opera,tiparita,manuscris,carte

where (opera.id\_opera=tiparita.opera and carte.isbn=tiparita.isbn and locatie\_in\_biblioteca='Zona

literatura') or (carte.isbn=manuscris.isbn and opera.id\_opera=manuscris.opera and

locatie\_in\_biblioteca='Zona literatura');

1. /\* numărul de cărți tipărite are sunt disponibile \*/

select count(carte.isbn)

from carte,tiparita

where carte.isbn=tiparita.isbn and carte.status='Disponibil';

1. /\* numărul membrilor care au rezervare de la data de 09.12.2017 până la data 13.12.2017 \*/

select count(membru.nume)

from membru,rezervare\_online

where membru.id\_membru=rezervare\_online.membru and data\_inceput>='2017-12-09' and

data\_final<='2017-12-13';

1. /\* lista membrilor care participă la evenimentul ShoeBox în data de 04.12.2017 \*/

select nume,prenume

from membru,eveniment,participare\_eveniment

where membru.id\_membru=participare\_eveniment.membru and

participare\_eveniment.data\_participarii='2017-12-04' and eveniment.denumire='ShoeBox';

1. /\* lista id-urilor membrilor care au făcut rezervări online \*/

select membru.id\_membru

from membru,rezervare\_online

where membru.id\_membru=rezervare\_online.membru;

1. /\* ISBN-ul cărților restante până la data curentă, numărul de zile restante și tariful de penalizare \*/

select carte.isbn,(curdate()-imprumut\_curent.data\_final) as 'Zile restante' ,((curdate()

-imprumut\_curent.data\_final)\*carte.tarif\_penalizare) as 'Penalizare'

from carte,imprumut\_curent,istoric\_imprumut

where carte.isbn=imprumut\_curent.isbn\_carte and curdate()>=imprumut\_curent.data\_final and

carte.isbn not in (select istoric\_imprumut.isbn from istoric\_imprumut);

1. /\* numele si prenumele membrilor cărora li s-a permis accesul la sală de către angajatul Ifrim Raluca \*/

select membru.nume,membru.prenume

from membru,acces\_sala,angajat

where membru.id\_membru=acces\_sala.membru and

acces\_sala.angajat\_responsabil=angajat.id\_angajat and angajat.nume='Ifrim' and

angajat.prenume='Raluca';

1. /\* lista denumirilor sălilor de studiu ordonate alfabetic \*/

select sala\_studiu.denumire

from sala\_studiu

order by sala\_studiu.denumire asc;

1. /\* numele angajațiilor și numele membrilor care au avut acces la sala Victor Babes \*/

select angajat.nume as 'Nume angajat',angajat.prenume as 'Prenume angajat',membru.nume as '

Nume membru',membru.prenume as 'Prenume membru'

from angajat,acces\_sala,membru,sala\_studiu

where angajat.id\_angajat=acces\_sala.angajat\_responsabil and

membru.id\_membru=acces\_sala.membru and sala\_studiu.nr\_sala=acces\_sala.sala

and sala\_studiu.denumire='Victor Babes';

1. /\* membri ordonați dupa vârstă \*/

select nume,prenume,(year(curdate())-year(membru.data\_nasterii)) as 'Varsta'

from membru

order by (year(curdate())-year(membru.data\_nasterii)) asc;

1. /\* numele și numărul de cărți din fiecare zonă \*/

select carte.locatie\_in\_biblioteca,count(isbn) as 'Numar carti'

from carte

group by locatie\_in\_biblioteca;

1. /\* titlul cărților care se găsesc în cel puțin două exemplare \*/

select distinct opera.titlu

from opera,tiparita,manuscris,carte

where (opera.id\_opera=tiparita.opera and carte.isbn=tiparita.isbn and nr\_copii>=2)

or (carte.isbn=manuscris.isbn and opera.id\_opera=manuscris.opera and nr\_copii>=2);

1. /\* numele membrilor care au participat la mai mult de un eveniment \*/

select membru.nume,membru.prenume

from membru,participare\_eveniment

where membru.id\_membru=participare\_eveniment.membru group by

participare\_eveniment.membru having count(participare\_eveniment.membru)>1;

1. /\* numele membrilor care au împrumutat aceeași carte \*/

select membru.nume

from membru

where membru.id\_membru in (select istoric\_imprumut.membru from istoric\_imprumut where

istoric\_imprumut.isbn in(select carte.isbn from carte));

*Obiecte de tip vedere*

1. /\* Aflați în ce sală de studiu sunt disponibile peste 20 de locuri \*/

drop view if exists locuri\_in\_sala;

create view locuri\_in\_sala as

select denumire as 'Denumire sala', nr\_locuri\_disponibile as 'Locuri disponibile'

from sala\_studiu

where nr\_locuri\_disponibile>20;

1. /\* Numărul de evenimente care încep în luna decembrie \*/

drop view if exists evenimente\_decembrie;

create view evenimente\_decembrie as

select distinct count(id\_event) as 'Numar evenimente'

from eveniment

where month(data\_inceput)=12;

1. /\* Cărțile tipărite în limba română \*/

drop view if exists tiparite\_rom;

create view tiparite\_rom as

select titlu as 'Carti tiparite in limba romana'

from carte,tiparita,opera

where carte.isbn=tiparita.isbn and opera.id\_opera=tiparita.opera and carte.limba='Romana';

1. /\* Toate informațiile despre cărți\*/

drop view if exists Carti;

create view Carti as

select carte.isbn,opera.titlu,opera.autor,carte.categorie,carte.status,carte.locatie\_in\_biblioteca,carte.tarif\_penalizare

from carte,opera,tiparita

where carte.isbn=tiparita.isbn and tiparita.opera=opera.id\_opera

group by opera.titlu

union

select carte.isbn,opera.titlu,opera.autor,carte.categorie,carte.status,carte.locatie\_in\_biblioteca,carte.tarif\_penalizare

from carte,opera,manuscris

where carte.isbn=manuscris.isbn and manuscris.opera=opera.id\_opera

group by opera.titlu asc;

1. /\* Informații împrumuturi \*/

drop view if exists Imprumuturi;

create view Imprumuturi as

select imprumut\_curent.id\_imprumut,imprumut\_curent.data\_inceput,imprumut\_curent.data\_final,membru.nume,membru.prenume,opera.titlu,opera.autor

from imprumut\_curent,membru,carte,opera,tiparita

where imprumut\_curent.membru=membru.id\_membru and imprumut\_curent.ISBN\_carte=carte.isbn and carte.isbn=tiparita.isbn and tiparita.opera=opera.id\_opera

group by membru.nume asc

union

select imprumut\_curent.id\_imprumut,imprumut\_curent.data\_inceput,imprumut\_curent.data\_final,membru.nume,membru.prenume,opera.titlu,opera.autor

from imprumut\_curent,membru,carte,opera,manuscris

where imprumut\_curent.membru=membru.id\_membru and imprumut\_curent.ISBN\_carte=carte.isbn and carte.isbn=manuscris.isbn and manuscris.opera=opera.id\_opera

group by membru.nume asc;

1. /\* Informații rezervări \*/

drop view if exists Rezervari;

create view Rezervari as

select rezervare\_online.id\_rezervare,rezervare\_online.data\_inceput,rezervare\_online.data\_final,membru.nume,membru.prenume,opera.titlu,opera.autor

from rezervare\_online,membru,carte,opera,tiparita

where rezervare\_online.membru=membru.id\_membru and rezervare\_online.ISBN=carte.isbn and carte.isbn=tiparita.isbn and tiparita.opera=opera.id\_opera

group by membru.nume asc

union

select rezervare\_online.id\_rezervare,rezervare\_online.data\_inceput,rezervare\_online.data\_final,membru.nume,membru.prenume,opera.titlu,opera.autor

from rezervare\_online,membru,carte,opera,manuscris

where rezervare\_online.membru=membru.id\_membru and rezervare\_online.ISBN=carte.isbn and carte.isbn=manuscris.isbn and manuscris.opera=opera.id\_opera

group by membru.nume asc;

1. /\* Informații istoric împrumuturi \*/

drop view if exists Istoric;

create view Istoric as

select istoric\_imprumut.id\_istoric,istoric\_imprumut.data\_returnarii,membru.nume,membru.prenume,opera.titlu,opera.autor

from istoric\_imprumut,membru,carte,opera,tiparita

where istoric\_imprumut.membru=membru.id\_membru and istoric\_imprumut.ISBN=carte.isbn and carte.isbn=tiparita.isbn and tiparita.opera=opera.id\_opera

group by istoric\_imprumut.id\_istoric asc

union

select istoric\_imprumut.id\_istoric,istoric\_imprumut.data\_returnarii,membru.nume,membru.prenume,opera.titlu,opera.autor

from istoric\_imprumut,membru,carte,opera,manuscris

where istoric\_imprumut.membru=membru.id\_membru and istoric\_imprumut.ISBN=carte.isbn and carte.isbn=manuscris.isbn and manuscris.opera=opera.id\_opera

group by istoric\_imprumut.id\_istoric asc;

1. /\* Informatii tiparite \*/

drop view if exists Informatii\_tiparite;

create view Informatii\_tiparite as

select opera.titlu,opera.autor,carte.limba,carte.nr\_total\_copii,carte.nr\_copii\_disponibile,carte.categorie,carte.locatie\_in\_biblioteca,carte.tarif\_penalizare,tiparita.editura

from carte,opera,tiparita

where carte.isbn=tiparita.isbn and tiparita.opera=opera.id\_opera

group by opera.titlu;

1. /\* Informatii manuscrise \*/

drop view if exists Informatii\_manuscrise;

create view Informatii\_manuscrise as

select opera.titlu,opera.autor,carte.limba,carte.nr\_total\_copii,carte.nr\_copii\_disponibile,carte.categorie,carte.locatie\_in\_biblioteca,carte.tarif\_penalizare,manuscris.anul\_scrierii

from carte,opera,manuscris

where carte.isbn=manuscris.isbn and manuscris.opera=opera.id\_opera

group by opera.titlu;

1. /\* Informatii membrii \*/

drop view if exists Informatii\_membrii;

create view Informatii\_membrii as

select id\_membru,nume,prenume,ocupatie,adresa\_email

from membru

group by id\_membru;

1. /\* Informatii acces \*/

drop view if exists Informatii\_acces;

create view Informatii\_acces as

select membru.nume as Nume\_Membru,membru.prenume as Prenume\_Membru,acces\_sala.sala,angajat.nume as Nume\_angajat,angajat.prenume as Prenume\_angajat,acces\_sala.data\_acces

from membru,acces\_sala,angajat

where membru.id\_membru=acces\_sala.membru and angajat.id\_angajat=acces\_sala.angajat\_responsabil

order by acces\_sala.data\_acces asc;

*Triggere*

1. /\* Trigger inserare în istoric după restituire carte \*/

drop trigger if exists inserare\_istoric;

delimiter //

create trigger inserare\_istoric after delete on imprumut\_curent

for each row begin

insert into istoric\_imprumut(data\_returnarii,membru,isbn)

values (curdate(),old.membru,old.isbn\_carte);

end //

delimiter ;

1. /\* Trigger pentru anunț penalizare la returnare \*/

drop trigger if exists penalizare;

delimiter //

create trigger penalizare after delete on imprumut\_curent

for each row begin

set @tarif=null;

select carte.tarif\_penalizare into @tarif from carte where carte.isbn=old.isbn\_carte;

if ((curdate()-old.data\_final)>0) then

set @message\_text=concat('Aveti de platit penalizare: ',(curdate()-old.data\_final)\*@tarif,' lei');

signal sqlstate value '45000'

set message\_text=@message\_text;

end if;

end //

delimiter ;

1. /\* Trigger verificare constrângeri vârstă membrii noi\*/

drop trigger if exists verificare\_membru;

delimiter //

create trigger verificare\_membru before insert on membru

for each row begin

if(new.data\_nasterii<'1900-01-01'or new.data\_nasterii>'2013-01-01') then

signal sqlstate '45000'

set message\_text='Varsta nu este corespunzatoare pt a deveni membru al bibliotecii';

end if;

end //

delimiter ;

*Proceduri stocate*

1. /\* Adăugare membru nou \*/

drop procedure if exists adaugare\_membru\_nou;

delimiter //

create procedure adaugare\_membru\_nou(nume varchar(40),prenume varchar(40),strada varchar(40),numar int,oras varchar(40),ocupatie varchar(40),data\_nasterii date,sex char(1),adresa\_email varchar(40))

begin

start transaction;

set @email=null;

select @email:=membru.adresa\_email from membru where membru.adresa\_email=adresa\_email;

if((@email is null) and (nume is not NULL) and (prenume is not NULL) and (strada is not NULL) and (numar is not NULL) and (oras is not NULL) and (ocupatie is not NULL) and (data\_nasterii is not NULL) and (sex is not NULL) and (adresa\_email is not NULL)) then

begin

insert into membru(nume,prenume,strada,numar,oras,ocupatie,data\_nasterii,sex,adresa\_email) values

(nume,prenume,strada,numar,oras,ocupatie,data\_nasterii,sex,adresa\_email);

commit;

end;

else rollback;

end if;

end //

delimiter ;

1. /\* Adăugare împrumut nou\*/

drop procedure if exists adaugare\_imprumut\_nou;

delimiter //

create procedure adaugare\_imprumut\_nou(membru int,isbn int)

begin

start transaction;

set @id\_membru=null;

select @id\_membru:=membru.id\_membru from membru where membru.id\_membru=membru;

if(@id\_membru is not null) then

begin

set @isbn\_c=null;

select @isbn\_c:=carte.isbn from carte where carte.isbn=isbn;

if(@isbn\_c is not null) then

begin

set @deja\_imprumutat=null;

select @deja\_imprumutat:=carte.status from carte where carte.isbn=isbn;

if(@deja\_imprumutat='Disponibil') then

begin

insert into imprumut\_curent(data\_inceput,data\_final,membru,isbn\_carte) values

(curdate(),curdate()+14,membru,isbn);

##scadem numarul de copii disponibile

update carte set nr\_copii\_disponibile=nr\_copii\_disponibile-1 where carte.isbn=isbn;

set @nr\_c=null;

select @nr\_c:=carte.nr\_copii\_disponibile from carte where carte.isbn=isbn;

if(@nr\_c=0) then

begin

update carte set status='Imprumutat' where carte.isbn=isbn;

commit;

end;

end if;

commit;

end;

else rollback;

end if;

commit;

end;

else rollback;

end if;

commit;

end;

else rollback;

end if;

end //

delimiter ;

1. /\* Restituire împrumut \*/

drop procedure if exists restituire\_imprumut;

delimiter //

create procedure restituire\_imprumut(membru int,id\_imprumut int)

begin

start transaction;

set @id\_membru=null;

select @id\_membru:=membru.id\_membru from membru where membru.id\_membru=membru;

if(@id\_membru is not null) then

begin

set @imprumut=null;

select @imprumut:=imprumut\_curent.id\_imprumut from imprumut\_curent where imprumut\_curent.id\_imprumut=id\_imprumut;

if(@imprumut is not null) then

begin

set @isbn=null;

select @isbn:=isbn\_carte from imprumut\_curent where imprumut\_curent.id\_imprumut=id\_imprumut;

delete from imprumut\_curent where imprumut\_curent.id\_imprumut=id\_imprumut;

##crestem numarul de copii disponibile

update carte set nr\_copii\_disponibile=nr\_copii\_disponibile+1 where carte.isbn=@isbn;

#actualizam disponibilitatea cartiie

set @nr\_c=null;

select @nr\_c:=carte.nr\_copii\_disponibile from carte where carte.isbn=@isbn;

if(@nr\_c>0) then

begin

update carte set status='Disponibil' where carte.isbn=@isbn;

commit;

end;

end if;

commit;

end;

else rollback;

end if;

commit;

end;

else rollback;

end if;

end //

delimiter ;

1. /\* Adăugare rezervare nouă \*/

drop procedure if exists adaugare\_rezervare\_noua;

delimiter //

create procedure adaugare\_rezervare\_noua(membru int,isbn int)

begin

start transaction;

set @id\_membru=null;

select @id\_membru:=membru.id\_membru from membru where membru.id\_membru=membru;

if(@id\_membru is not null) then

begin

set @isbn\_c=null;

select @isbn\_c:=carte.isbn from carte where carte.isbn=isbn;

if(@isbn\_c is not null) then

begin

set @deja\_imprumutat=null;

select @deja\_imprumutat:=carte.status from carte where carte.isbn=isbn;

if(@deja\_imprumutat='Disponibil') then

begin

insert into rezervare\_online(data\_inceput,data\_final,membru,isbn) values

(curdate(),curdate()+1,membru,isbn);

##scadem numarul de copii disponibile

update carte set nr\_copii\_disponibile=nr\_copii\_disponibile-1 where carte.isbn=isbn;

set @nr\_c=null;

select @nr\_c:=carte.nr\_copii\_disponibile from carte where carte.isbn=isbn;

if(@nr\_c=0) then

begin

update carte set status='Rezervat' where carte.isbn=isbn;

commit;

end;

end if;

commit;

end;

else rollback;

end if;

commit;

end;

else rollback;

end if;

commit;

end;

else rollback;

end if;

end //

delimiter ;

1. /\* Anulare rezervare online \*/

drop procedure if exists anulare\_rezervare;

delimiter //

create procedure anulare\_rezervare(membru int,id\_rezervare int)

begin

start transaction;

set @id\_membru=null;

select @id\_membru:=membru.id\_membru from membru where membru.id\_membru=membru;

if(@id\_membru is not null) then

begin

set @rezervare=null;

select @rezervare:=rezervare\_online.id\_rezervare from rezervare\_online where rezervare\_online.id\_rezervare=id\_rezervare;

if(@rezervare is not null) then

begin

set @isbn=null;

select @isbn:=isbn from rezervare\_online where rezervare\_online.id\_rezervare=id\_rezervare;

delete from rezervare\_online where rezervare\_online.id\_rezervare=id\_rezervare;

##crestem numarul de copii disponibile

update carte set nr\_copii\_disponibile=nr\_copii\_disponibile+1 where carte.isbn=@isbn;

#actualizam disponibilitatea cartiie

set @nr\_c=null;

select @nr\_c:=carte.nr\_copii\_disponibile from carte where carte.isbn=@isbn;

if(@nr\_c>0) then

begin

update carte set status='Disponibil' where carte.isbn=@isbn;

commit;

end;

end if;

commit;

end;

else rollback;

end if;

commit;

end;

else rollback;

end if;

end //

delimiter ;

1. /\* Adăugare operă nouă \*/

drop procedure if exists adaugare\_opera;

delimiter //

create procedure adaugare\_opera(titlu varchar(45),autor varchar(45))

begin

start transaction;

set @title=null;

select @title:=opera.titlu from opera where opera.autor=autor;

if(@title<>titlu || @title is null and titlu is not null and autor is not null) then

begin

insert into opera(titlu,autor) values

(titlu,autor);

commit;

end;

else rollback;

end if;

end //

delimiter ;

1. /\* Adăugare tiparită nouă \*/

drop procedure if exists adaugare\_tiparita;

delimiter //

create procedure adaugare\_tiparita(isbn int ,limba varchar(40),nr\_total\_copii int ,nr\_copii\_disponibile int ,categorie varchar(40),status varchar(40),locatie\_in\_biblioteca varchar(40),tarif\_penalizare double,opera int,editura varchar(45))

begin

start transaction;

set @carte=null;

select @carte:=isbn from carte where carte.isbn=isbn;

if(@carte is null ) then

begin

set @opera=null;

select @opera:=id\_opera from opera where opera.id\_opera=opera;

if(@opera is not null) then

begin

insert into carte(isbn,limba,nr\_total\_copii,nr\_copii\_disponibile,categorie,status,locatie\_in\_biblioteca,tarif\_penalizare) values

(isbn,limba,nr\_total\_copii,nr\_copii\_disponibile,categorie,status,locatie\_in\_biblioteca,tarif\_penalizare);

insert into tiparita(isbn,opera,editura) values

(isbn,opera,editura);

commit;

end;

else rollback;

end if;

commit;

end;

else rollback;

end if;

end //

delimiter ;

1. /\* Adăugare manuscris nou \*/

drop procedure if exists adaugare\_manuscris;

delimiter //

create procedure adaugare\_manuscris(isbn int ,limba varchar(40),nr\_total\_copii int ,nr\_copii\_disponibile int ,categorie varchar(40),status varchar(40),locatie\_in\_biblioteca varchar(40),tarif\_penalizare double,opera int,anul year)

begin

start transaction;

set @carte=null;

select @carte:=isbn from carte where carte.isbn=isbn;

if(@carte is null ) then

begin

set @opera=null;

select @opera:=id\_opera from opera where opera.id\_opera=opera;

if(@opera is not null) then

begin

insert into carte(isbn,limba,nr\_total\_copii,nr\_copii\_disponibile,categorie,status,locatie\_in\_biblioteca,tarif\_penalizare) values

(isbn,limba,nr\_total\_copii,nr\_copii\_disponibile,categorie,status,locatie\_in\_biblioteca,tarif\_penalizare);

insert into manuscris(isbn,opera,anul\_scrierii) values

(isbn,opera,anul);

commit;

end;

else rollback;

end if;

commit;

end;

else rollback;

end if;

end //

delimiter ;

1. /\* Adăugare membru în sala de studiu \*/

drop procedure if exists adaugare\_acces;

delimiter //

create procedure adaugare\_acces(membru int,sala int ,angajat int)

begin

start transaction;

set @id\_membru=null;

select @id\_membru:=membru.id\_membru from membru where membru.id\_membru=membru;

set @id\_sala=null;

select @id\_sala:=sala\_studiu.nr\_sala from sala\_studiu where sala\_studiu.nr\_sala=sala;

set @id\_angajat=null;

select @id\_angajat:=angajat.id\_angajat from angajat where angajat.id\_angajat=angajat;

if(@id\_membru is not null and @id\_sala is not null and @id\_angajat is not null) then

begin

insert into acces\_sala(membru,sala,angajat\_responsabil,data\_acces) values

(membru,sala,angajat,curdate());

commit;

end;

else rollback;

end if;

end //

delimiter ;

*Utilizatori*

use biblioteca;

DROP USER 'member'@localhost;

CREATE USER 'member'@'localhost';

GRANT SELECT ON biblioteca.\* TO 'member'@'localhost';

GRANT INSERT ON biblioteca.\* TO 'member'@'localhost';

GRANT UPDATE ON biblioteca.\* TO 'member'@'localhost';

GRANT DELETE ON biblioteca.\* TO 'member'@'localhost';

GRANT SHOW VIEW ON biblioteca.\* TO 'member'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.adaugare\_imprumut\_nou TO 'member'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.adaugare\_membru\_nou TO 'member'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.adaugare\_rezervare\_noua TO 'member'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.restituire\_imprumut TO 'member'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.anulare\_rezervare TO 'member'@'localhost';

DROP USER 'employee'@localhost;

CREATE USER 'employee'@'localhost';

GRANT SELECT ON biblioteca.\* TO 'employee'@'localhost';

GRANT INSERT ON biblioteca.\* TO 'employee'@'localhost';

GRANT UPDATE ON biblioteca.\* TO 'employee'@'localhost';

GRANT DELETE ON biblioteca.\* TO 'employee'@'localhost';

GRANT SHOW VIEW ON biblioteca.\* TO 'employee'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.adaugare\_membru\_nou TO 'employee'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.adaugare\_tiparita TO 'employee'@'localhost';

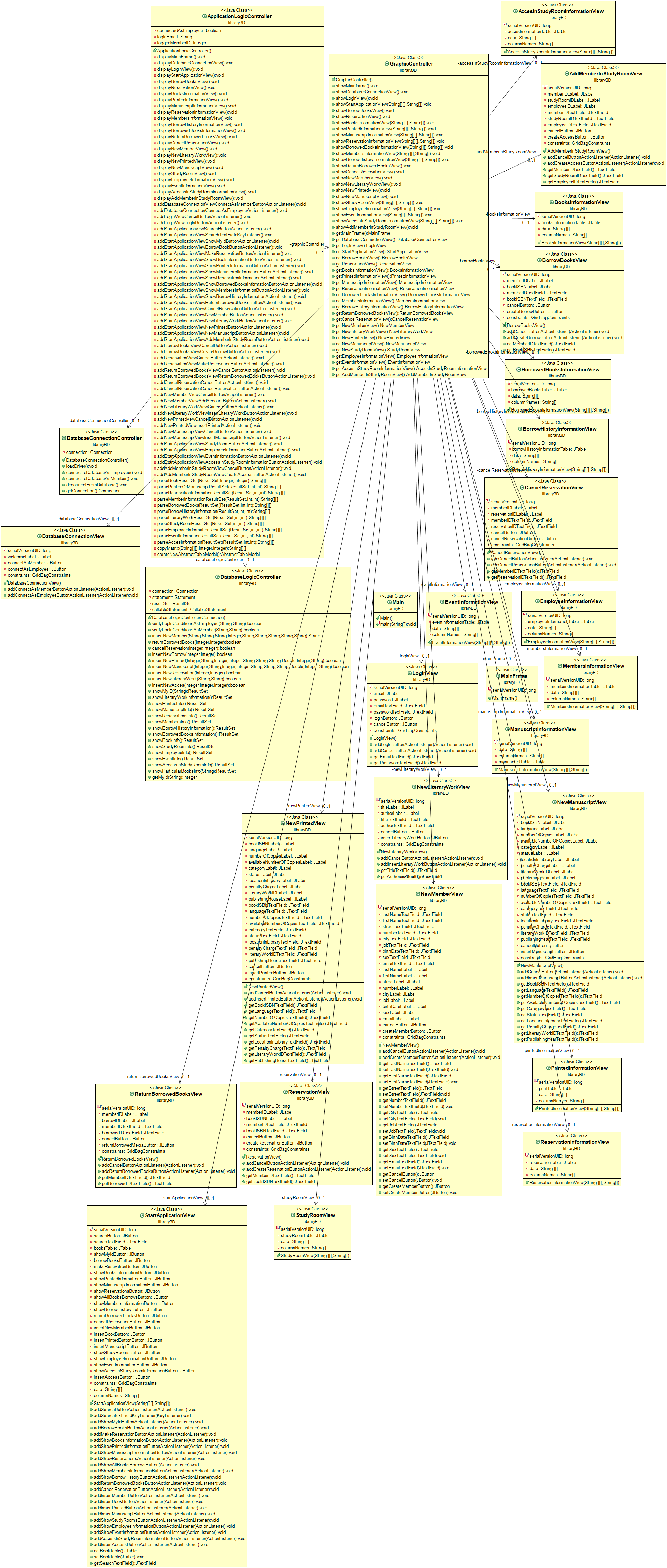
GRANT EXECUTE ON PROCEDURE biblioteca.adaugare\_manuscris TO 'employee'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.adaugare\_opera TO 'employee'@'localhost';

GRANT EXECUTE ON PROCEDURE biblioteca.adaugare\_acces TO 'employee'@'localhost

4.Interfata Java

4.1 Structura claselor in Java.Diagrama UML



Clasele din Java sunt:

* AccessInStudyRoomInformationView
* AddMemberINStudyRoomView
* ApplicationLogicController
* BooksInformationView
* BorrowBooksView
* BorrowedBooksInformationView
* BorrowHistoryInformationView
* CancelReservationView
* DatabaseConnectionController
* DatabaseConnectionView
* DatabaseLogicController
* EmployeeInformationView
* EventInformationView
* GraphicController
* LogInView
* Main
* MainFrame
* ManuscriptInformationView
* MembersInformationView
* NewLiteraryWorkView
* NewManuscriptView
* NewMemberview
* NewPrintedView
* PrintedInformationView
* ReservationInformationView
* Reservationview
* ReturnBorrowedBooksView
* StartApplicationView
* StudyRoomView

4.2 Manual de utilizare/instalare si elemente de securitate

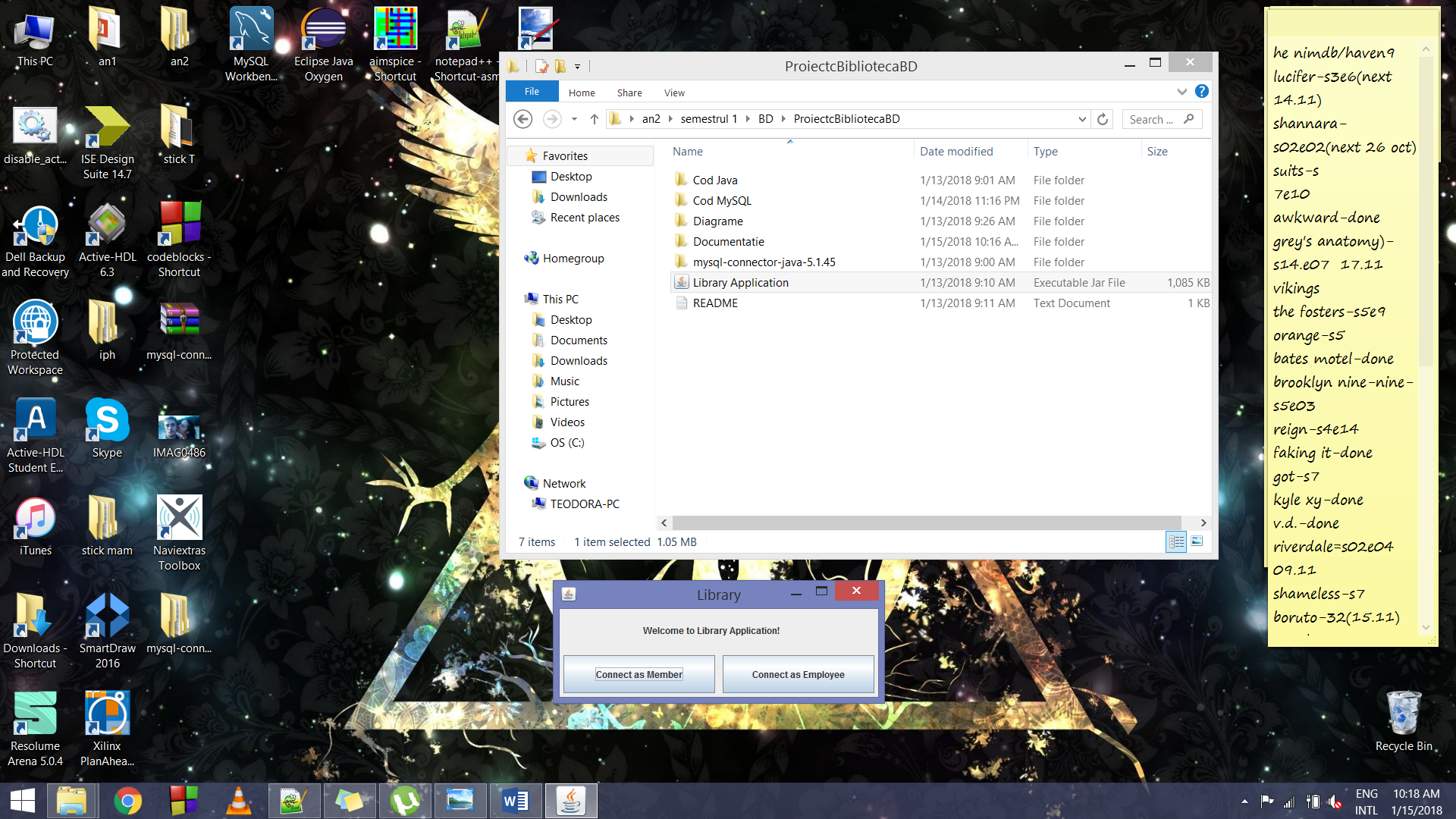
Pentru realizarea unei logari de tip test:

* Membru: UserName: orice adresa de email din baza de date (de ex. [nas.razvan@gmail.com](mailto:nas.razvan@gmail.com)), fara parola
* Angajat: Username:employee, fara parola

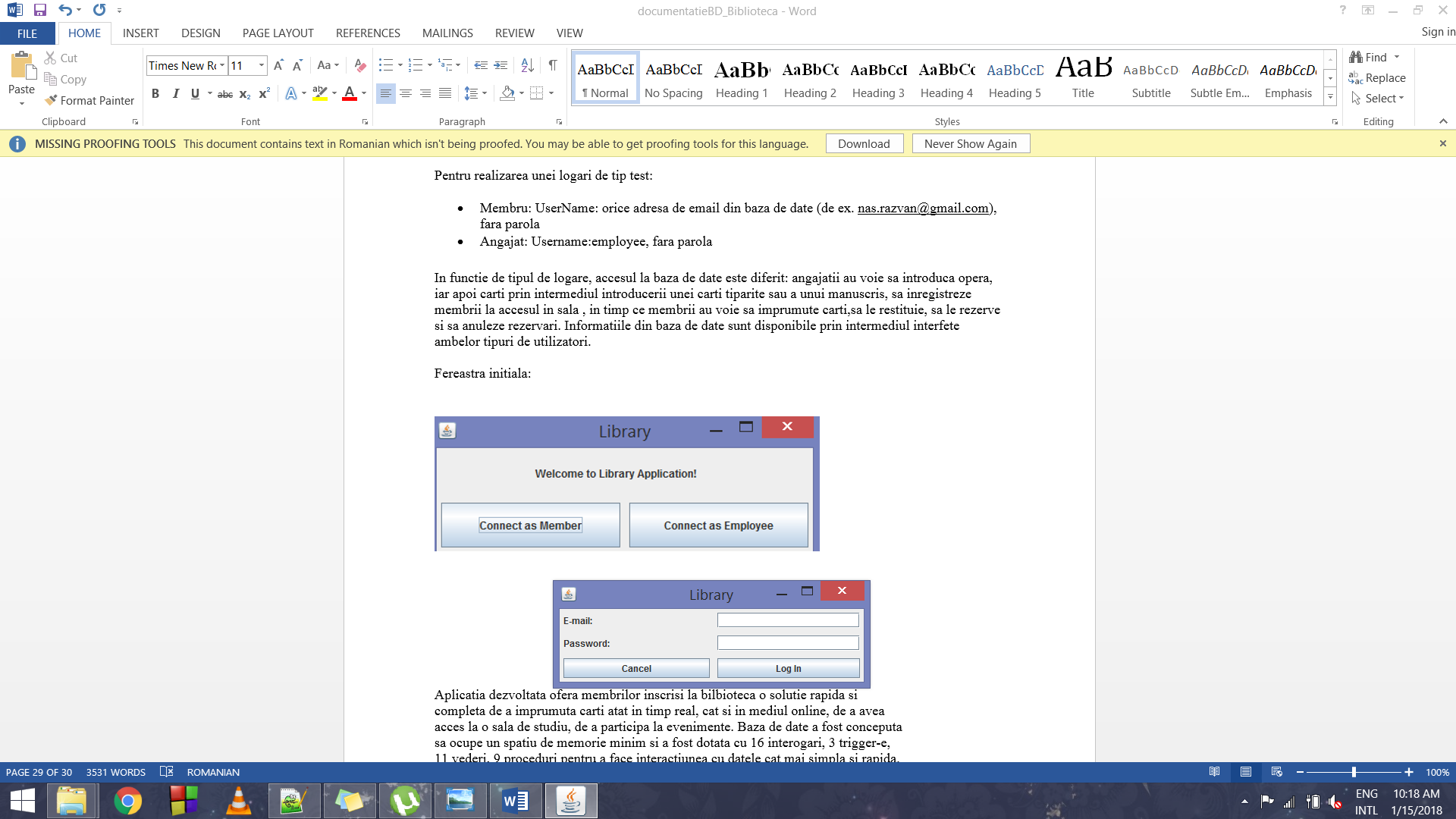
In functie de tipul de logare, accesul la baza de date este diferit: angajatii au voie sa introduca opera, iar apoi carti prin intermediul introducerii unei carti tiparite sau a unui manuscris, sa inregistreze membrii la accesul in sala , in timp ce membrii au voie sa imprumute carti,sa le restituie, sa le rezerve si sa anuleze rezervari. Informatiile din baza de date sunt disponibile prin intermediul interfete ambelor tipuri de utilizatori.

Acesta logare diferentiata pe tipuri de utilizatori reprezinta elementele de securitate ale aplicatiei, accesul la baza de date fiind diferit, protejand modificarea tabelelor de catre utilizatori neautorizati.

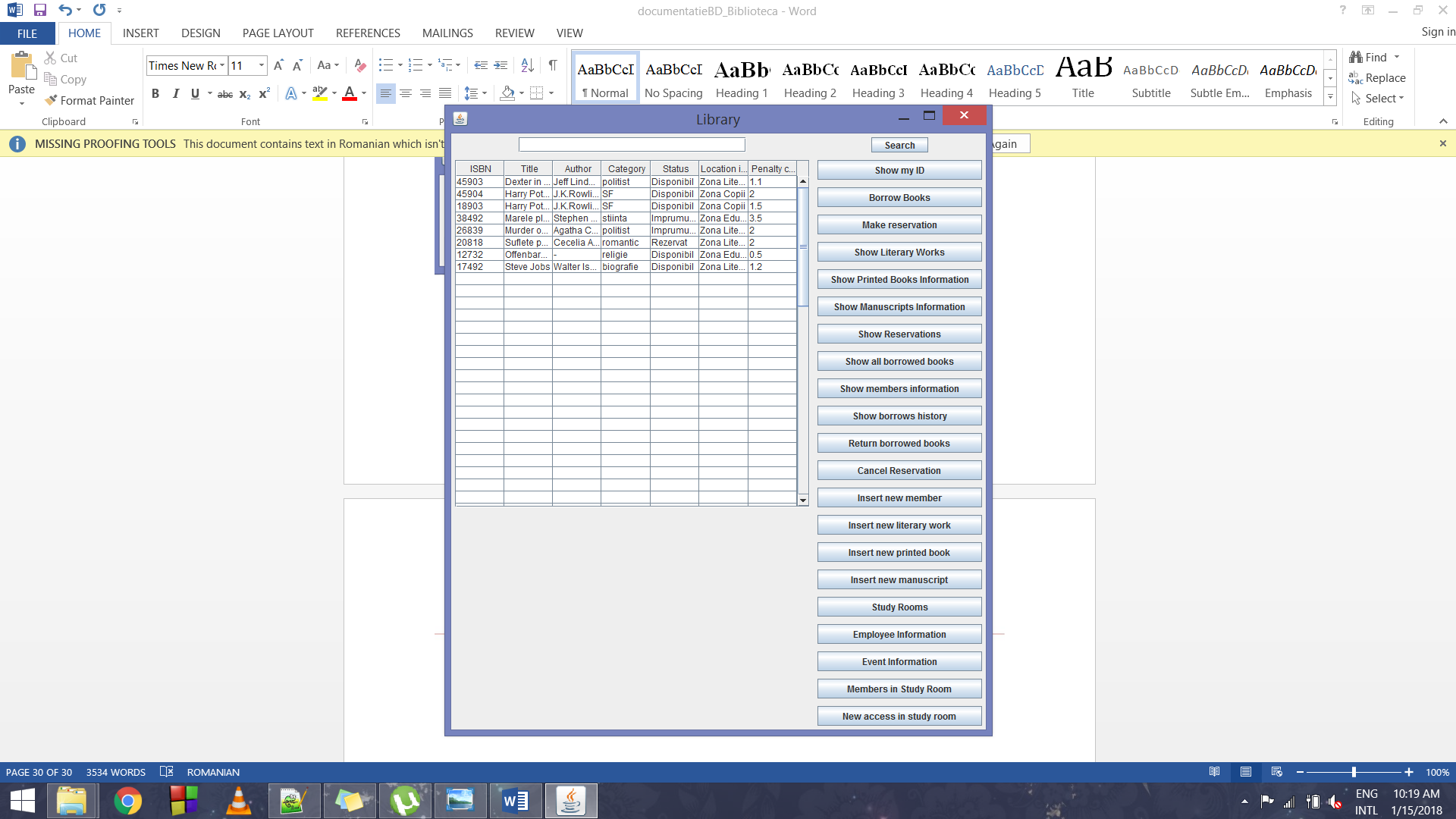
Fereastra initiala:



Fereastra de logare:



Fereastra principala a aplicatiei:



5.Concluzii si dezvoltari ulterioare

Aplicatia dezvoltata ofera membrilor inscrisi la bilbioteca o solutie rapida si completa de a imprumuta carti atat in timp real, cat si in mediul online, de a avea acces la o sala de studiu, de a participa la evenimente. Baza de date a fost conceputa sa ocupe un spatiu de memorie minim si a fost dotata cu 16 interogari, 3 trigger-e, 11 vederi, 9 proceduri pentru a face interactiunea cu datele cat mai simpla si rapida. Procedurile au fost create cu scopul de a nu fi nevoiti administratorii bazei de date sa-si creeze propriile subprograme ci doar sa apeleze cele deja existente, baza de date fiind usor de modificat.

Dezvoltarea aplicatiilor folosind Java si MySQL este posibila, dar este o varianta de implementare a aplicatiilor intre multe variante. Legat de aplicatia de fata, dupa cum am mai mentionat este demonstrativa, dar metodele folosite aici se pot folosi usor pentru a implementa o aplicatie reala.