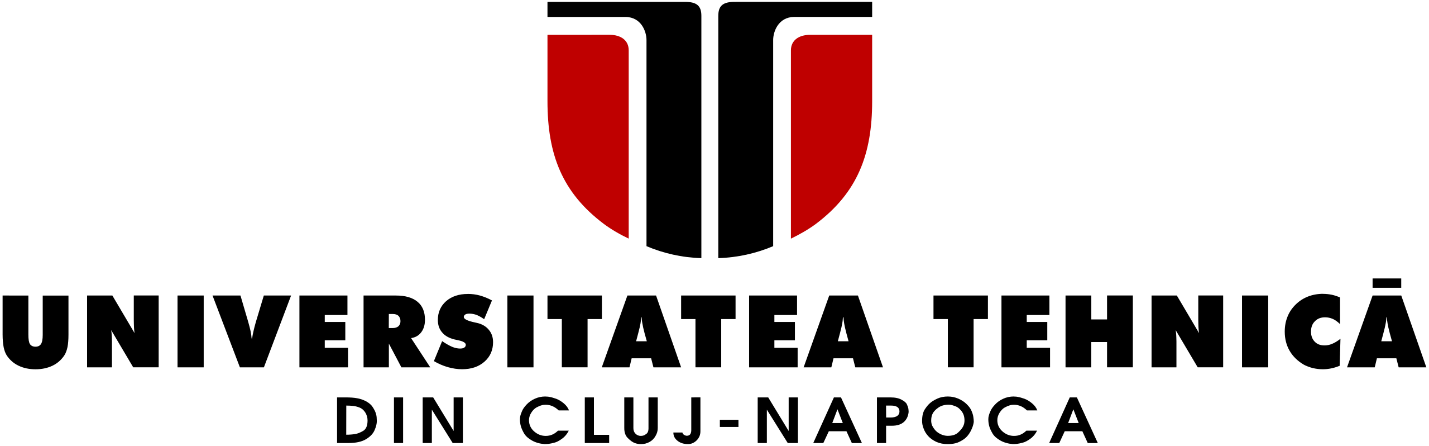
****

**PROIECT**

la disciplina

Introducere in Baze de Date

**Titlu Platforma de studiu**

Alexandra Coltea

Madalina Strugar

**An academic : 2022 – 2023**

PROIECT de SEMESTRU

Catedra de Calculatoare

Disciplina : Introducere in Baze de Date

Coordonator: s.l. ing. Cosmina IVAN

Data 13.01.2023

**Cuprins**

**1. Introducere**

Introducere, argumente, scop si obiective specifice

## 2. Analiza cerintelor utilizatorilor ( Specificatiile de proiect)

* Ipoteze specifice domeniului ales pentru proiect ( cerinte, constrangeri )
* Organizare structurata( tabelar) a cerintelor utilizator
* Determinarea si caracterizarea de profiluri de utilizatori ( admin, user intern, user extern…diversi alti “actori”)

## 3. Modelul de date si descrierea acestuia

* Entitati si atributele lor ( descriere detaliata **– implementarea fizica**)
* Diagrama EER/UML pentru modelul de date complet
* Normalizarea datelor

## 4. Detalii de implementare

* Descrierea functionala a modulelor ( organizarea logica a acestora- de ex . structura claselor Java, cod HTML, JSP, ASP, PhP)
* Manual de utilizare/instalare (diferentiat pe tipuri de actori)
* Elemente de securizare a aplicatiei

**5. Concluzii limitari si dezvoltari ulterioare**

**6. Bibliografie**

**1. Introducere**

Proiectul implementeaza o aplicatie de gestiune a unei platforme de studiu. Acesta implementeaza o baza de date MySQL care este accesata cu ajutorul unei interfete grafice. Interfata ofera utilizatorilor funcţionalităţi, corespunzător cu drepturile pe care aceştia le deţin în sistem, descrise prin intermediul unor roluri. Aplicatia are 4 tipuri de utilizator, fiecare are anumite drepturi de acces si editare la baza de date. Cele 4 tipuri de utilizator sunt: super-administrator, administrator, profesor si student.

Logarea in aplicatie se face pe baza numelui si a unei parole.

## 2. Analiza cerintelor utilizatorilor ( Specificatiile de proiect)

2.a Ipoteze specifice domeniului ales pentru proiect ( cerinte, constrangeri )

Aplicatia va permite gestiunea cu usurinta a activitatilor didactice si astfel a interacțiunilor dintre studenti si profesori.

Utilizatorul de tip administrator poate adăuga, modifica şi şterge informaţii în baza de date. El are acces la datele personale si la cele la profesorilor si studentilor. Utilizatorul de tip super-administrator are drept de acces asupra intregii baze de date.

Utilizatorul de tip profesor poate preda mai multe materii, la fiecare materie avand optiunea de a alege dintre 3 tipuri de activitati: curs, seminar si laborator. Profesorul stabileste din interfata grafica impartirea procentuala pe tipurile de activitati. Acestia pot evalua studentii cu note pentru fiecare tip de activitate. Profesorii pot accesa un catalog, unde pot filtra studentii dupa cursuri si le pot adauga note.

Utilizatorul de tip student se poate inscrie la mai multe materii, unde isi poate alege activitatile la care vrea sa participe. La fiecare activitate la care s-a inscris poate fi evaluat de catre profesorul responsabil. Studentii pot renunta la cursuri si isi pot vedea notele. Totodata, studentii se pot inscrie in grupuri de studiu pentru o anumita materie, daca sunt inscrisi la materia respectiva. Acestia pot sa vada toti membrii grupului si sa lase mesaje. Pe grup, studentii pot adauga activitati.

La logare, studentii si profesorii pot sa isi vada activitatile din ziua curenta sau pot accesa o pagina cu toate activitatile la care sunt asignati / inscrisi.

2.b Organizare structurata( tabelar) a cerintelor utilizator

Initial am creat entitatea utilizator care contine toate atributele comune tipurilor de utilizator. Apoi am creat alte 3 entitati, administrator, profesor, student care mostenesc din tabela utilizator si au in plus atributele particulare fiecarui tip. Ulterior am creat tabele cursuri, grup de studiu, activitati si activitati grup. Apoi am rezolvat relatiile many to many folosind tabele de legatura(profesor\_curs).

2.c Determinarea si caracterizarea de profiluri de utilizatori

Super-administrator:

* Are aceleasi drepturi ca si utilizatorul de tip administrator
* In plus, are acces si drept de modificare/stergere asupra datelor tuturor tipurilor de utilizator

Administrator:

* Adaugare,stergere,modificare date utilizatori
* Cautare utilizatori si filtrare dupa tip(ex afisam doar studentii)
* Asignarea profesorilor la cursuri
* Cautare cursuri dupa nume(afisam numele profesorului si numele studenitilor participanti)

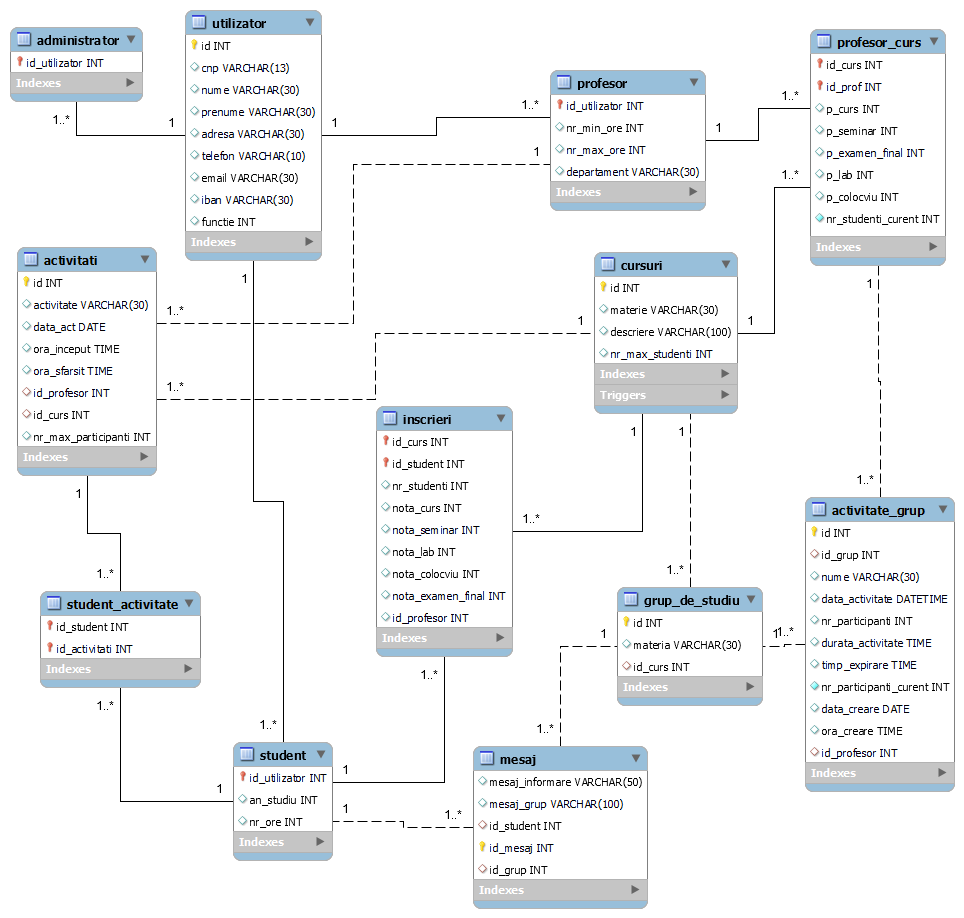
Profesor:

* Programare activitati(curs,seminar etc) intr-un calendar si specifica numarul de participanti
* Stabilieste impartirea procentuala pe tipurile de activitati (ex. 20% seminar, 35% laborator, 45% curs/examenul de la curs)
* Poate accesa un catalog si pot adauga note

Student:

* Se poate inscrie la cursuri
* Poate renunta la cursuri
* Isi poate vedea notele
* Alege activitati(care nu se suprapun cu alte activitati si nu s-a atins numarul maxim de studenti inscrisi(primeste mesaj daca nu se poate inscrie)
* Inscriere grup de studiu(pentru materiile la care e inscris)
* Vizualizare membrii grupului de studiu
* Poate sa lase mesaje
* Adaugare activitati pentru grup(sa defineasca un numar minim de participanti si o perioada in care ceilalti studenti pot sa anunte

## 3. Modelul de date si descrierea acestuia



Entitatea Utilizator

* id int - cheie primara
* cnp,nume,prenume,adresa,telefon,email,iban varchar - date personale despre utilizator
* functie int - tipul de utilizator(admin=0,profesor=1,student=2,super-administrator=3)

Entitatea Profesor

* id\_utilizator int - cheie primara si straina
* nr\_ore\_min,nr\_ore\_max int
* departament varchar

Entitatea Student

* id\_utilizator int - cheie primara si straina
* an\_studiu int
* nr ore int

Entitatea Cursuri

* id int - cheie primara
* materie varchar
* departament varchar
* nr max studenti

Entitatea Activitati

* id int - cheie primara
* activitate varchar
* data\_act date
* ora inceput, ora sfarsit time
* id profesor,id curs int - chei straine
* nr max participanti int

Entitatea Inscrieri

* id\_curs,id\_student chei primare si straine
* nr\_studenti ,id\_profesor int
* nota\_curs,nota\_seminar,nota\_laborator,nota\_examen\_final int

Entitatea Grup de studiu

* id int cheie primara
* materia varchar
* id\_curs int cheie straina

Entitatea Activitate grup

* id int cheie primara
* id\_grup,id\_prof int chei straine
* nume varchar
* data\_activitate datetime
* nr\_participanti,nr\_participanti\_curent int
* durata\_activitate,timp\_expirare ,ora\_creare time
* data\_creare date

Normalizarea datelor

Normalizarea este procesul reversibil de transformare a unei relaţii în relaţii de structură mai simplă. (Procesul este reversibil în sensul că nici o informaţie nu este pierdută în timpul transformării). Scopul normalizării este de a suprima redundanţele logice, de a evita anomaliile la reactualizare şi rezolvarea problemei reconexiunii.

Codd a definit iniţial 3 forme normale, notate prin FN1, FN2 şi FN3

Forma de normalizare 1 – FN1

Dintre toate formele normale, doar FN1 are caracter de obligativitate. Se spune că o bază de date este normalizată daca toate relaţiile se află măcar în FN1. O relaţie este în FN1 dacă domeniile pe care sunt definite atributele relaţiei sunt constituite numai din valori atomice. Un tuplu nu trebuie să conţină atribute sau grupuri de atribute repetitive.

Forma de normalizare 2 – FN2

FN2 este strâns legată de noţiunea de dependenţă funcţională.

O relaţie se află în a doua formă normală FN2 dacă se află în forma normală FN1 şi fiecare atribut care nu este cheie este dependent de întreaga cheie primară.

Forma de normalizare 3- FN3

A treia regulă de normalizare cere ca toate câmpurile din tabele să fie independente între ele. O relaţie este în forma normală trei FN3 dacă se găseşte în FN2 şi fiecare atribut care nu este cheie (nu participă la o cheie) depinde direct de cheia primară.

Forma de normalizare Boyce-Codd (BCNF – Boyce-Codd Normal Form)

BCNF e o versiune putin mai restrictiva de forma normala 3. In cazul unei forme normale 3, toate atributele depinde de o cheie, o cheie in intregime si numai de o cheie.

Baza de date a fost normalizata prin perechi de chei straine care devin chei primare pentru tabele copii.

Tipuri de relatii prezente in baza de date

* relatia 1:1 Într-o relație unu-la-unu, o înregistrare dintr-un tabel este asociată cu una și numai o singură înregistrare dintr-un alt tabel. Un astfel de exemplu in baza de date prezentata se poate exemplifica intre tabelele student-utilizator.
* Relatia 1:n Într-o relație one-to\_many, o înregistrare dintr-un tabel poate fi asociată cu una sau mai multe înregistrări dintr-un alt tabel. De exemplu, fiecare client poate avea mai multe comenzi de vânzare. Spre exemplu, aceasta relatie este prezenta intre tabele cursuri-activitati.
* Relatia m:n O relație many-to-many apare atunci când mai multe înregistrări dintr-un tabel sunt asociate cu mai multe înregistrări dintr-un alt tabel. Spre exemplu, inainte de normalizare, aceasta relatia exista intre tabele profesor-cursuri. Acesta a fost rezolvata prin intermediul tabelului profesor\_curs.

## 4. Detalii de implementare

In constructia aplicatiei s-au utilizat limbajele MySQL si Java(Swing). Initial am realizat arhitectura bazei de date prin crearea tabelelor si legaturilor dintre acestea, iar apoi am implementat operatiile utilizand proceduri. Ulterior am facut legatura bazei de date cu compilatorul Java, unde am apelat procedurile si am realizat interfata grafica penrtu utilizarea user-friendly a aplicatiei.

4.a Instrumente suport de implementare

* MySQL Workbench

MySQL Workbench este un instrument vizual unificat pentru arhitecți de baze de date, dezvoltatori și DBA. MySQL Workbench oferă modelare de date, dezvoltare SQL și instrumente complete de administrare pentru configurarea serverului, administrarea utilizatorilor, backup și multe altele. MySQL Workbench este disponibil pe Windows, Linux și Mac OS X.

* EclipseIDE

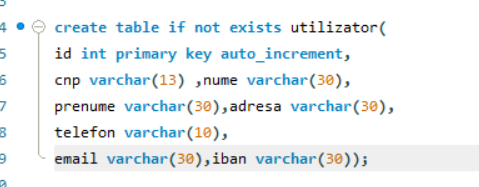
Eclipse este definit ca fiind o platformă pentru dezvoltarea aplicațiilor bazate pe calculator care utilizează diverse limbaje de programare, cum ar fi JAVA, Python, C/C++, Ruby și multe altele. Eclipse este un IDE (kit de dezvoltare integrat) și în principal programarea bazată pe JAVA se realizează pe această platformă. Există mai multe plug-in-uri și alte plug-in-uri suplimentare pot fi instalate în platformă. Pot fi dezvoltate aplicații client avansate. JDT este utilizat pentru realizarea programării în Eclipse IDE.

4.b Limbaje

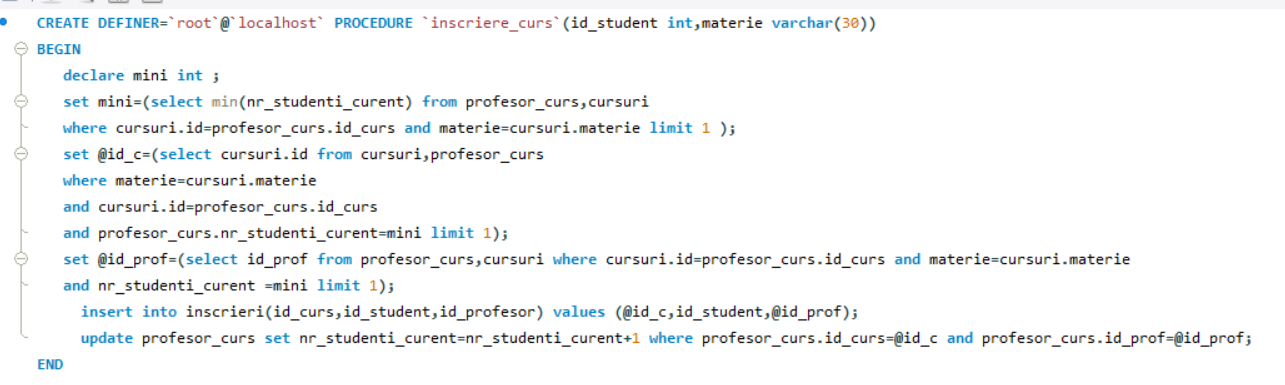
* MySQL

MySQL este un sistem de gestionare a bazelor de date relaționale (RDBMS) utilizat pe scară largă. MySQL este un sistem popular de gestionare a bazelor de date relaționale cu sursă deschisă. Acesta este utilizat pentru a stoca, organiza și extrage date într-o manieră structurată. MySQL este utilizat în mod obișnuit în aplicațiile web și poate fi integrat cu limbaje de programare precum PHP, Java și Python. Este cunoscut pentru fiabilitatea, ușurința de utilizare și performanța sa.

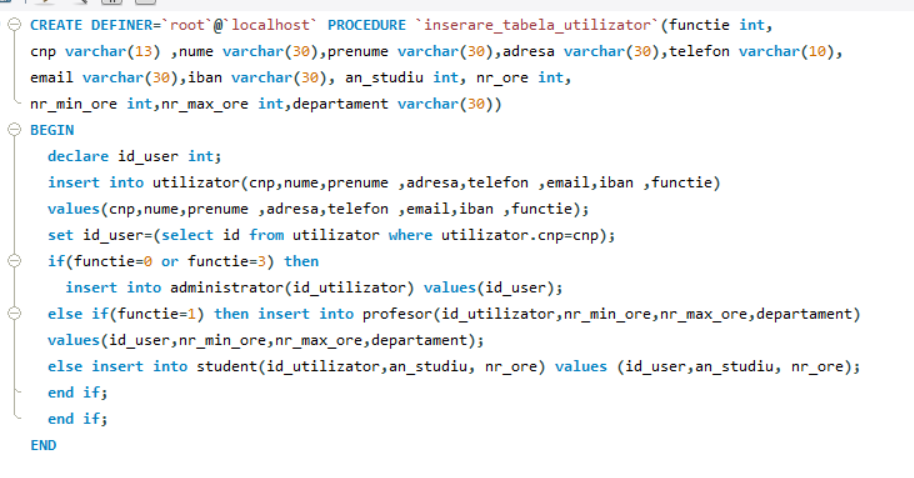
Exemplu: Creare tabel



Proceduri



Aceasta procedura realizeaza inscrierea unei student la o materie. Asignarea se face la profesorul cu cei mai putini studenti, astfel in mini selectam numarul minim de studenti dintre numarul de studenti inscrisi la fiecare profesor care preda materia respectiva.

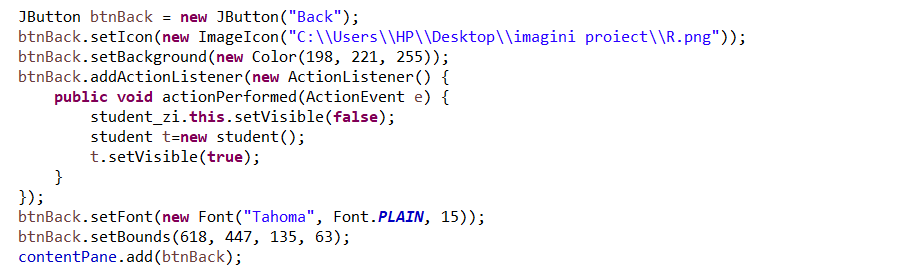


Aceasta procedura creeaza un utilizator nou in momentul in care o persoana isi creeaza un cont nou. Datele comune sunt introduse in tabela utilizator si in functie de tipul de user ales se introduc datele suplimentare in tabelele corespunzatoare.

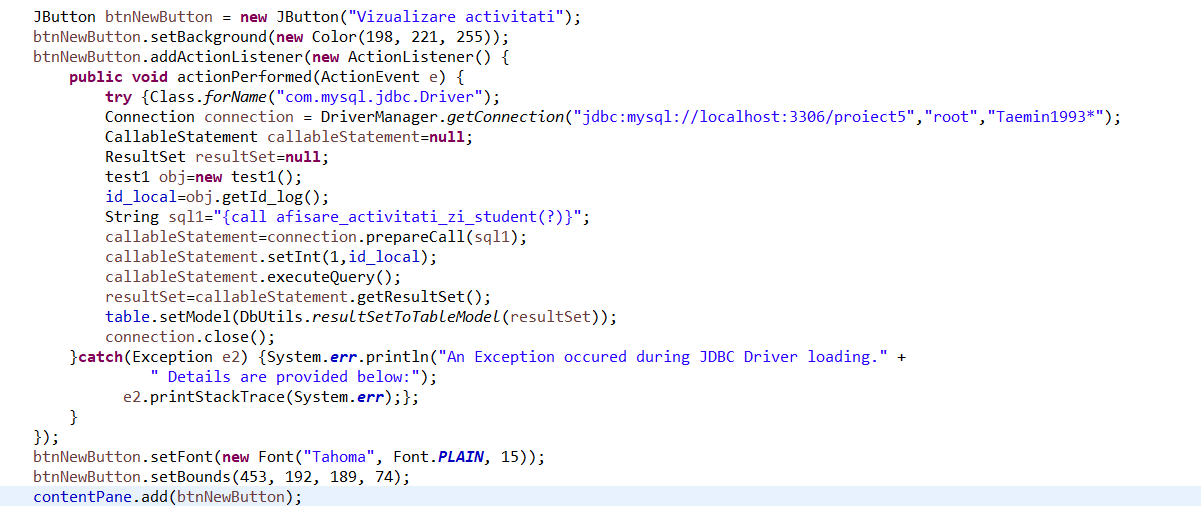
* Java

Java este un limbaj de programare de uz general, bazat pe clase, orientat pe obiecte și conceput pentru a avea cât mai puține dependențe de implementare posibil. Acesta este conceput pentru a permite dezvoltatorilor de aplicații să "scrie o singură dată, rulează oriunde" (WORA), ceea ce înseamnă că un cod Java compilat poate rula pe toate platformele care acceptă Java fără a fi nevoie de recompilare. Aplicațiile Java sunt de obicei compilate în cod byte care poate rula pe orice mașină virtuală Java (JVM), indiferent de arhitectura calculatorului. Java este unul dintre cele mai populare limbaje de programare utilizate, în special pentru aplicațiile web client-server, cu un număr de 9 milioane de dezvoltatori.

Exemplu Buton Back(Intoarcere)



Exemplu Apelare procedura



**5. Concluzii limitari si dezvoltari ulterioare**

5.a Concluzii

În concluzie, proiectul Java și MySQL pentru o platformă de studiu demonstrează puterea și flexibilitatea acestor două tehnologii atunci când sunt utilizate împreună. Java oferă capacitatea de a crea aplicații web robuste, scalabile și sigure, în timp ce MySQL oferă o modalitate fiabilă și eficientă de stocare și gestionare a datelor. Platforma de studiu utilizează aceste capacități pentru a le oferi studenților un mediu de învățare interactiv și ușor de utilizat. Caracteristicile platformei includ o bază de date centralizată pentru stocarea informațiilor despre studenți, profesori și administratori, un sistem de conectare securizat și posibilitatea de a crea și gestiona cursuri. În plus, platforma oferă posibilitatea de a urmări progresul studenților și de a le oferi feedback în timp real. În general, platforma de studiu Java și MySQL este un instrument valoros atât pentru educatori, cât și pentru studenți, oferind o experiență de învățare cuprinzătoare și atractivă.

5.b Dezvoltari ulterioare

Învățare interactivă: Platforma ar putea sa ofere caracteristici interactive de învățare, cum ar fi teste, examene și feedback în timp real pentru studenți.

Multimedia: Platforma poate suporta conținut multimedia, cum ar fi videoclipuri, imagini și fișiere audio.

Compatibilitate mobilă: Platforma poate fi optimizată pentru dispozitive mobile, permițând studenților să acceseze cursurile și să participe la activități interactive din mers.

**6. Bibliografie**

* http://www.mysql.com
* <http://www.w3schools.com/sql>
* <http://www.codebind.com/category/java-tutorials/>
* <https://stackoverflow.com/questions/tagged/java>
* <https://www.tutorialspoint.com/java/util/timer_schedule_period.htm>
* <https://1bestcsharp.blogspot.com/2015/02/java-how-to-get-selected-row-values.html>
* https://docs.oracle.com/javase/tutorial/jdbc/basics/storedprocedures.html