Facultatea de Automatică și Calculatoare

Departamentul de Calculatoare

Proiect

La disciplina

Baze de Date

Anul 2021-2022

Platformă de Studiu

4 ianuarie 2022

Budiul Cristian-Carol

Giesswein Alexia

Kovács Paul-Adrian

Improvised checklist TODO delete it once it s done

**I. SCOP si CERINTE GENERALE**

**1. Aplicatia - arhiva cu cod si descriere Readme a acesteia si a modului de lansare in executi**

**2. Un Document (sursa si tiparit), care sa cuprinda conform unei structuri propuse (vezi Model ) :**

1. tema proiectului – sumar executiv

2. detalii de proiectare conceptuală a bazei de date (diagrama EER/UML)

3. solutia de transformare in relational ( daca este necesara, many to many, reflexivitate, alegere chei)

4. descrierea bazei de date relationale – tabele (8 tabele /atribute/chei/indecsi )

5. argumentarea nivelului de normalizare

6. 20 interogari asupra datelor(fisier .txt), cod SQL si minim 5 interogări în algebra relationala

7. Realizarea de obiecte de tip vedere(min 3), trigere (min 2), procedurile stocate (min 8), securizarea accesului

8. implementarea a minim 2 formulare de acces la baza de date și cel puțin 2 rapoarte ce extrag date din mai mult de un tabel.

9. descrierea aplicaţiei din punct de vedere functional (module PhP/Java/Net) – descrierea minimala a fiecarei clase implementate si un model al ierarhiei de clase a proiectului

10. manual de instalare/configurari /utilizare

11. concluzii si dezvoltari ulterioare

**II. ETAPE in derularea proiectului**

1.1.Implementarea schemei in SGBD.

Scop: identificarea structurii de date a aplicației, realizarea modelului abstract, conceptual în EER/UML. formularea cerintelor de utilizare pentru aplicatie

Cerinte asupra datelor si diagrama EER/UML: alegeti o aplicatie care sa cuprinda min 6-8 entitati si relatii (nontriviala).

1.2.Construirea , popularea si interogarea bazei de date

Scop: translatare diagrama EER/UML in scheme relationale, implementarea schemei intr-un SGBD: MSSQL/MySQL , popularea schemei BD.

Explicati modul in care ati realizat translatarea/reprezentarea in schema relationala. Pentru situatiile ce comporta posibil mai multe solutii argumentati alegerea.

Argumentati nivelul de normalizare a schemelor de relatii.

Scrieti un script SQL ce creeaza tabelele bazei de date, specificand tipurile atributelor, cheile primare, constrangeri not null si unique, valori default ,constrangeri de cheie straina si politicile necesare de reactie la modificari (ON DELETE NO ACTION este default), etc, eventual alte constranegri suplimentare necesare si argumentate( tip check, sau triggere)

Incarcarea datelor. Scrieti un script suplimentar ce contine elemente INSERT pentru relatii, astfel incat constrangerile specificate sa nu fie violate.

Identificati in setul de interogari posibile si necesare, acelea care poseda urmatoarele caracteristici de implementare SQL : agregare cu group by si having, imbricare SELECT cu agregare, imbricare cu negare (NOT EXISTS si NOT IN) , joinuri pe aceeasi tabela/tabele diferite.

1.3. Vederi, proceduri, trigere. Se vor implementa pentru aplicatie numarul solicitat de componente programatice .

1.4.Integrarea elementelor de securitate si a minim doi utilizatori de sistem Implementarea interfetei aplicatiei.Se poate opta pentru implementare PhP/Java

**III.STRUCTURĂ de PROIECT ( se va utiliza modelul atasat)**

UTCN

Facultatea de Automatica si Calculatoare

Departamentul de Calculatoare

Disciplina Baze de Date Anul xxxx

Denumire proiect (alegeti un acronim interesant)

Data

Echipa de proiect

**Cuprins**

1. **Introducere** Introducere, argumente, scop si obiective specifice
2. **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 , diversi alti “actori”)

**3. Modelul de date si descrierea acestuia**

Entitati si atributele lor ( descriere detaliata)

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, module PhP)

Elemente de utilizare/instalare (diferentiat pe tipuri de actori)

Elemente de securizare a aplicatiei

**5. Concluzii limitari si dezvoltari ulterioare**

**6. Bibliografie**

Cuprins

[1 Introducere 6](#_Toc91782670)

[2 Analiza cerințelor utilizatorilor 6](#_Toc91782671)

[2.1 Cerințe 6](#_Toc91782672)

[2.2 Determinarea și caracterizarea de profiluri de utilizatori 7](#_Toc91782673)

[2.3 Organizare structurată a cerințelor utilizatorilor 7](#_Toc91782674)

[3 Modelul de date 8](#_Toc91782675)

[4 Detalii de implementare 8](#_Toc91782676)

[5 Posibilități de dezvoltare ulterioare 8](#_Toc91782677)

[6 Concluzii 8](#_Toc91782678)

[7 Bibliografie 8](#_Toc91782679)

# Introducere

Proiectul în cauză reprezintă un sistem informatic destinat gestiunii unei platforme de studiu. Aplicația va folosi un sistem de gestiune pentru baze de date MySQL, iar interacțiunea cu aceasta va fi realizata prin interfață grafică.

# Analiza cerințelor utilizatorilor

## Cerințe

Aplicația va permite gestiunea cu usurință a activitaților didactice. Vor exista mai mutle materii care au una sau mai multe tipuri de activități (curs, seminar, laborator), iar orice materie poate fi predată de mai mulți profesori.

**Funcționalități generale**

* Accesarea aplicație printr-un proces de autentificare de către mai multe tipuri de utilizatori
* Funționalitate pentru deautentificare, astfel încât și un alt utilizator să îl poată folosi ulterior, fără a fi necesară repornirea sa
* Posibilitatea utilizatorilor de a vizualiza datele personale
* Activitățile se desfașoară recursiv între 2 date

**Funcționalități specifice unui administrator**

* Adăugare, modificare și ștergere informații în baza de date legate de utilizatori
* Căutare utilizatori dupa nume
* Filtrare utilizatori dupa tip
* Căutare materii după nume
* Vizualizare studenti înscriși la o materie
* Asignare profesori la materii

**Funcționalități specifice unui profesor**

* Stabilirea împărțirii procentuale pe tipurile de activitați
* Programare activități (cu număr maxim de participanți, doar în viitor)
* Accesare catalog (cu filtru dupa materie)
* Adaugare note în catalog
* Descărcare catalog sub forma de fisier
* Vizualizare activități proprii din ziua curentă (cu posibilitatea de descărcare)
* Vizualizare toate activitățiile proprii (cu posibilitatea de descărcare)

**Funcționalități specifice unui student**

* Însciere la materie (dacă sunt locuri libere și nu există suprapuneri)
* Renuțare la materie
* Vizualizare note
* Înscriere în grup de studiu pentru o anumită materie (dacă este înscris la acea materie)
* Vizualizare membri grup
* Planificare activitate in grup (Număr minim participanți, data, timp de expirare)
* Vizualizare activități proprii din ziua curentă (cu posibilitatea de descărcare)
* Vizualizare toate activitățiile proprii (cu posibilitatea de descărcare)

## Determinarea și caracterizarea de profiluri de utilizatori

Aplicaţia va putea fi accesată, pe baza unui proces de autentificare, de către mai multe tipuri de utilizatori: studenti, profesori, administratori.

Pentru fiecare tip de utilizator se vor reține date personale: CNP, nume, prenume, adresă, număr de telefon, email, cont IBAN, numărul de contract.

Pentru un utilizator de tip student se va reține si anul de studiu și numărul de ore pe care trebuie să le sustină.

Pentru un utilizator de tip profesor se vor reține și materiile predate, numărul minim și numărul maxim de ore pe care le poate preda și departamentul din care face parte.

Va exista şi un rol de super-administrator care poate opera inclusiv asupra utilizatorilor de tip administrator.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Student | Profesor | Admin |
| Date personale | da | da | da |
| Date despre alți utilizatori | nu | nu | da |
| Date despre materii | doar ale sale | doar ale sale | da |
| Alocare/Dealocare student la activitați | da | nu | da |
| Alocare/Dealocare profesori la materii | nu | nu | da |

## Organizare structurată a cerințelor

|  |
| --- |
| Gestiunea utilizatorilor |
| 1. Autentificare |
| 1. Deautentificare |
| 1. Implementare sistem de control al drepturilor de acces al utilizatorilor în cadrul sistemului informatic pe baza rolurilor |

|  |
| --- |
| Funcționalități pentru utilizatorul de tip admin |
| 1. Operații CRUD pe utilizatori |
| 1. Operații CRUD pe materii |
| 1. Operații CRUD pe grupuri de studii |
| 1. Căutare utilizatori dupa nume |
| 1. Filtrare utilizatori dupa tip |
| 1. Căutare materie |
| 1. Căutare materie după nume |
| 1. Asignare profesori la materii |
| 1. Vizualizare studenti înscriși la o materie |
| 1. Adăugare activitate grup |

|  |
| --- |
| Funcționalități pentru utilizatorul de tip student |
| 1. Căutare materie |
| 1. Încriere materie |
| 1. Vizualizare note |
| 1. Vizualizare grupuri |
| 1. Înscriere în grup de studiu |
| 1. Părasire grup de studiu |
| 1. Vizualizare membri din grup |
| 1. Adăugare activitate grup |
| 1. Vizualizare/descărcare activități curente/din viitor |
| 1. Mesaje pe grup |

|  |
| --- |
| Funcționalități pentru utilizatorul de tip profesor |
| 1. Adăugare activatăți |
| 1. Programare activități |
| 1. Gestionare ponderi note |
| 1. Notare studenți |
| 1. Vizualizare studeți |
| 1. Descărcare cataloage |
| 1. Vizualizare/descărcare activități curente/din viitor |

# Modelul de date

## Enititati si atributele lor

* persoane -> student/profesor/admin

-> atribute: cnp, nume, prenume, adresa, numar de telefon, email, iban, numar contract

* admini -> atribute: cnp, toate atributele de la persoane

-> cheie straina catre persoane

* studenti -> atribute: cnp, an de stiudiu, numar de ore, toate atributele de la persoane

-> cheie straina catre persoane

* profesori -> atribute: cnp, nr minim de ore, numar maxim de ore, departament, toate atributele de la persoane

-> cheie straina catre persoane

* materii -> atribute: id, nume, descriere, procent curs, procent seminar, procent laborator, numar maxim de studenti,

recurenta curs, recurenta seminar, recurenta laborator

* materii\_studenti -> legatura intre materii si studenti

-> atribute: id materie, cnp student, categorie(curs/seminar/laborator), nota

-> chei straine catre materii si persoane

* materii\_profesor -> legatura intre materii si profesori

-> atribute: id materie, cnp profesor

-> chei straine catre materii si persoane

* calendar -> atribute: id, data programarii, durata, id materie, categorie(curs/seminar/laborator), numar maxim

-> cheie straina catre materii

* calendar\_studenti -> legatura intre calendar si studenti

-> atribute: cnp student, id calendar

-> chei straine catre calendar si persoane

* grup\_studiu -> atribute: id, id materie

-> cheie straina catre materii

* grup\_studiu\_studenti -> legatura intre grup\_studiu si studenti

-> atribute: cnp student, id grup

-> chei straine catre grup\_studiu si persoane

* grup\_studiu\_mesaje -> mesajele de pe grupuri

-> atribute: id grup, cnp student, mesaj, data si oraa trimiterii mesajului

-> chei straine catre grup\_studiu si persoane

* grup\_studiu\_activitati -> atribute: id, id grup, cnp profesor, nume, descriere, data programarii, durata, data expirarii, numar minim

-> chei straine catre grup\_studiu si persoane

* grup\_studiu\_activitati\_studenti -> legatura intre activitati si studenti

-> atribute: id activitate, , cnp student

-> chei straine catre grup\_studiu\_activitati si persoane

## Diagrama EER/UML pentru modelul de date complet

## Normalizarea datelor

# Detalii de implementare

# Posibilități de dezvoltare ulterioare

# Concluzii

# Bibliografie