Catalog universitar

Context

In contextul provocarilor din ce in ce mai aspre ale educatiei sec XXI si impinsi de nevoia contiunua de a evolua, ne-am propus sa digitalizam modalitatea de gestiune existenta pentru un catlog universitar pentru a reduce in primul rand timpul extrem de mare pe care l-am pierde folosind birocatia clasica(hartii inutile, pierderea acestora, necesitatea deplasarii intr-un anumit loc special pentru a modifica/verifca notele si absentele studentilor, a accesa orarul, in ceea ce priveste profesorii, iar pentru studenti greutatea cu care se confrunta privind situatia proprie). Ca urmare propunem crearea unei aplicati moderne de gestionare unor moderne care sa permita accesul la multiple facilitati.

Cerinte functionale:

* Centralizarea catalogului

Descriere: catalogul propus va aduna toate datele necesare pentru acumularea tuturor entitatiilor ce tin de o universitate(studenti, profesori, grupe, sali de clasa, facultati, specializari, etc.).

* Aplicatia sa fie de tip client-server

Descriere: aplicatia in sine va avea la baza 3 compenete:

1. Clientul: aplicatie baza pe o interfata grafica, care permite gestionarea rapida si eficienta a entitatilor catalogului in functie de ce utilizator e logat. Acesta trimite cereri la server in scopul primirii unui raspuns la intrebarile puse(query SQL).
2. Baza de date: stocheaza toate informatiile aferente entitatiilor catalogului si trimite la cerere, daca exista, informatii referitoare la acestea. De asemenea ca orice baza de date implementeaza operatiile de baza read, update, delete, create.
3. Server: face legature intre client si baza de date, primind cererea de la user si trimite un raspuns de la baza de date inapoi acestuia.

* Existenta conturilor de utilizator

Descriere: conturile vor fi de 3 tipuri: student, profesor, admin

1. Student: are cel mai mic nivel de permisiuni, poate sa isi vada propriile materii, note, absente, restante, media, orarul(cu salile de clasa), grupa, podibilitatea de a primi notificari la eventuale modificari.
2. Profesor: are un nivel mediu de permisiuni, poate sa isi vada grupele de studenti, orarul(cu salile de clasa), poate sa modifice notele unor studenti.
3. Admin: are un nivel ridicat de premisiuni, poate sa modifice studentii si profesorii cat si conturile acestora, sa adauge studenti/profesori, sa publice anunturi.

* Grupa de studii

Descriere: va avea o denumire, va retine studentii, profesorii cunrenti, elibereaza o fisa de note. Vor fi impartite pe ani de studii, facultate, specializare.

* Fisa de note

Descriere: contine notele unei grupe, poate face returna media generala, numarul de restantieri per grupa, restantele(studentul care are restanta, materia).

* Anul de studiu

Descriere: compus din toate grupele specializarii, toti profesorii de la toate grupele, precum si materiile.

* Matereia

Descriere: asociata unui profesor, unui student, an de studiu, unei facultati.

* Functie

Descriere: imparite in functii didactice(asisitent universitar, lector universitar, conferentiar universitar, profesor universitar, doctor) si de conducere(sef laborator, sef departament, secretar, decan).

* Specializarea

Descriere: contine toti anii de studiu(folosita cel mai des ca un criteriu de sortare al studentilor).

* Facultatea

Descrire: contine specializarile(de asemenea folosita ca un criteriu de sortare al studentilor dar si al profesorilor).

Cerinte nefunctionale:

* Detaliere server

Descriere: server creat in limbaj de programare C++, acesta gestioneaza socket-urile(mecanismele prin care asiguram conexiunea dintre client si server prin internet, utilizand o adresa IP, un port si un protocol), executa operatiile de binding, listen, accept, receive si send, aflandu-se in stransa legatura cu baza de date.

* Baza de date

Descriere:

1. Pentru student: proceseaza cerea acestui de vizualizare(a notelor, colegilor, etc.) operatie de tip SELECT
2. Pentru profesor: proceseaza cererea acestuia de vizualizare(a studentilor, a notelor, etc.) operatie de tip SELECT, cererea de modificare situatie(note, absente) operaie de tip INSERT/UPDATE
3. Pentru admin: proceseaza cererea de editare/stergere/adaugare/vizualizare(a grupelor, a profesorilor, a fiselor de note, a orarului etc.).