

## Catalog universitar

### Context

În contextul provocărilor din ce în ce mai aspre ale educației secolului XXI și împinși de nevoia continuă de a evolua, ne-am propus să digitalizăm modalitatea de gestiune existentă pentru un catalog universitar cu scopul de a reduce în primul rând timpul extrem de mare pe care l-am pierde folosind birocrăția clasică(hârtii inutile, pierderea acestora, necesitatea deplasării într-un anumit loc special pentru a modifica/verifica notele și absențele studenților, indisponibilitatea orarului, în ceea ce privește profesorii, iar pentru studenți greutatea cu care se confruntă privind situația proprie). Ca urmare propunem crearea unei aplicații moderne de gestionarea a unui catalog de acest fel care sa permită accesul la multiple facilități.

#### Cerințe funcționale:

- Centralizarea catalogului

Descriere: catalogul propus va aduna toate datele necesare pentru acumularea într-un singur loc a tuturor entităților ce țin de partea educațională a unei universități(studenți, profesori, grupe, săli de clasa, facultăți, specializări, etc.).

- Aplicația să fie de tip client-server

Descriere: aplicația în sine va avea la bază 3 componente:

1. Clientul: aplicație bazată pe o interfață grafică, care permite gestionarea rapidă și eficientă a entităților catalogului în funcție utilizatorul logat. Acesta trimite cereri la server în scopul primirii unui răspuns la întrebările puse(query SQL).
2. Baza de date: stochează toate informațiile aferente entităților catalogului și trimite la cerere, dacă există, informații referitoare la acestea. De asemenea, ca orice bază de date, implementează operațiile de bază read, update, delete, create.
3. Server: face legătura între client și baza de date, primind cererea de la user și trimițând un răspuns de la baza de date înapoi acestuia.

- Existența conturilor de utilizator

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

1. Student: are cel mai mic nivel de permisiuni, poate să își vadă propriile materii, note, absențe, restanțe, media, orarul(cu sălile de clasă), grupa, posibilitatea de a primi notificări la eventuale modificări(posibil și pe mail).
2. Profesor: are un nivel mediu de permisiuni, poate să își vadă grupele de studenți la care predă, orarul(cu sălile de clasă), poate sa modifice notele unor studenți.

3. Admin: are un nivel ridicat de permisiuni, poate să modifice studenți și profesori asociați unor materii, ani de studiu, specializări etc., cât și conturile acestora, să adauge/șteargă studenți/profesori, să publice anunțuri.

- Grupa de studii

Descriere: va avea o denumire, va reține studenții, profesorii curenți care predau la grupă, eliberând o fișă de note pentru fiecare disciplină. Vor fi împărțite pe ani de studii, facultate, specializare.

- Fișa de note

Descriere: conține notele unei grupe, poate calcula(la cerere) media generală, numărul de restanțieri pe grupă, restanțele(studentul care are restanța, materia).

- Anul de studiu

Descriere: compus din toate grupele din componența acelui an, toți profesorii de la toate grupele, precum și materiile.

- Materia

Descriere: asociată unui profesor, unui student, an de studiu, specializări, unei facultăți.

- Funcție

Descriere: funcții didactice(asistent universitar, lector universitar, conferențiar universitar, profesor universitar, doctor) și de conducere(șef laborator, șef departament, secretar, decan).

- Specializarea

Descriere: conține toți anii de studiu(folosită cel mai des ca un criteriu de sortare al studenților).

- Facultatea

Descriere: conține specializările(de asemenea folosită ca un criteriu de sortare al studenților dar și al profesorilor).

Cerințe nefuncționale:

- Detaliere server

Descriere: server creat în limbaj de programare C++, acesta gestionează socket-urile(mecanismele prin care asigurăm conexiunea dintre client și server prin internet, utilizând o adresă IP, un port și un protocol) folosind librăria Winsock32, execută operațiile de binding, listen, accept, receive și send, aflându-se în stransă legătură cu baza de date.

- Baza de date

Descriere:

Aplicație SGBD utilizată: Microsoft SQL.

1. Pentru student: procesează cererea acestuia de vizualizare(a notelor, colegilor, etc.) operație de tip SELECT
2. Pentru profesor: procesează cererea acestuia de vizualizare(a studenților, a notelor, etc.) operație de tip SELECT, cererea de modificare situație(note, absente) operație de tip INSERT/UPDATE
3. Pentru admin: procesează cererea de editare/ștergere/adăugare/vizualizare(a grupelor, a profesorilor, a fișelor de note, a orarului etc.).

- Clientul

Descriere: aplicație creată utilizând mediul de programare Qt Creator 6.4. E format din 2 componente:

1. Interfața grafică(GUI) care va asigura întreaga funcționare a aplicației și prin care end-userul va putea relaționa cu celelalte componente ale aplicației. Va conține o serie de obiecte cu care orice user, fie student, profesor sau admin, va putea interacționa pentru realizarea diferitelor facilități puse la dispoziție prin realizarea cerințelor software de mai sus.
2. Backend-ul integrează codul pentru conectarea aplicației client cu serverul. Astfel, atunci când lansăm în execuție aplicația sub forma unui client care dorește să se logheze pe un cont de user, se va realiza automat conexiunea acestui client la server.