

## **Dezvoltarea Aplicatiilor Web – Examen – 11.01.2021**

Sa se creeze o baza de date cu informatii despre abonamente si clientii care detin aceste abonamente. Un client detine mai multe abonamente, dar un abonament apartine unui singur client.

Creati cele doua modele si **adaugati cateva inregistrari** cu ajutorul IDE-ului (3 in fiecare tabel). Creati constrangerile de cheie primara, cheie externa. Denumirile tabelelor si atributelor trebuie sa respecte cerinta.

1. **(1p)** Baza de date contine urmatoarele doua tabele:

**+ Subscription:**

- Id - int - id abonament (cheie primara);
- Descriere - string – o scurta descriere a abonamentului;
- Numar - int – numarul abonamentului;
- DataEmitere - datetime – data la care a fost emis abonamentul;
- ClientId - int - id-ul clientului care detine abonamentul(cheie externa);

**+Client:**

- Id - int - id client (cheie primara);
- Nume - string – numele clientului;

- ID-urile **nu vor fi** vizibile, editabile sau selectabile din interfata web.
- In final sa existe **cel putin 3 intrari in tabele**.

2. **(1p)** Asigurati-va, la fiecare pas, de validitatea datelor de intrare. Astfel, toate campurile sunt obligatorii si veti afisa mesaj pentru fiecare camp (ex de mesaj: Decrierea abonamentului este obligatorie, etc). Numarul abonamentului trebuie sa fie unic. De asemenea, data se va adauga manual de catre utilizator sau se va selecta dintr-un calendar si se va emite abonamentul folosind o data din viitor, inclusiv data curenta.

3. **(4.5p)** Creati o aplicatie web care implementeaza cele 4 operatii fundamentale (Create, Read, Update, Delete) pe entitatea **Subscription**. Se vor utiliza toate verbele http invatate:

- **Index (0.5p)** – Sa se afiseze abonamentele impreuna cu numele clientului care detine abonamentul.
- **Show (1p)** – Sa se afiseze detaliile despre un singur abonament (in functie de abonamentul selectat) impreuna cu numele clientului. Metoda se va numi Show si va fi accesata prin intermediul rutei **afisareAbonament**.

- **New (1.25p)** – Sa se implementeze posibilitatea adaugarii unui nou abonament. Metoda se va numi New. Se vor folosi urmatoarele helperele Html: **Label**(@HtmlLabel), **TextBox** pentru campurile de tip Text. Data sa va adauga manual de catre utilizator sau se va selecta dintr-un calendar.  
Pentru selectarea numelui clientului, se va folosi un DropDown unde cu ajutorul unei metode vor fi selectati toti clientii existenti in baza de date.
  - **Edit (1.25p)** – Sa se implementeze posibilitatea editarii unui abonament, unde numele clientului va fi selectat dintr-un dropdown (la fel ca la punctul anterior). In momentul in care un utilizator doreste sa editeze un abonament, datele actuale ale acestuia vor fi afisate in formular. Dupa modificarea abonamentului se va afisa un mesaj de confirmare cu textul: “Abonamentul a fost modificat cu succes!”. Metoda se va numi Edit.
  - **Delete (0.5p)** – Sa existe posibilitatea stingerii unui abonament. Metoda se va numi Delete.
4. **(0.5p)** Pentru a facilita navigarea sa existe o pagina **Layout** (se poate modifica Layout-ul existent) care sa contina link catre paginile New si Index. Acestea se vor numi: “Adaugare abonament”, “Afisare abonamente”.
5. **(2p)** Sa existe o pagina – al carei link se va numi “Cautare abonamente” si va fi accesat din Layout-ul de la punctul anterior – in care sa existe posibilitatea cautarii abonamentelor dupa descriere sau dupa data, campul fiind obligatoriu. Pentru fiecare cautare se vor afisa: descrierea abonamentului, data emiterii si numele clientului. De asemenea, sa existe posibilitatea sortarii rezultatelor crescator sau descrescator in functie de data emiterii.

**ATENTIE:**

- Pentru **crearea unui Controller se foloseste optiunea MVC 5 Controller-Empty**
- Pentru **crearea unui View se foloseste Template Empty (without model)**
- **NU** se genereaza cod (nici Controller-ul, nici View-ul)
- Aplicatia trebuie sa ruleze in browser
- Proiectul se va numi **NumePrenumeInitialaAlDoileaPrenumeGrupa** astfel: se utilizeaza doar un prenume scris in intregime, pentru al doilea prenume se utilizeaza doar initiala, iar pentru numarul grupei o sa scrieti doar ultimele doua cifre

**EX: Student: Popescu Marian Cristian, grupa: 242 -> PopescuMarianC42**

- Arhiva o sa fie de tip **.zip** si va avea acelasi nume ca proiectul
- In momentul in care incarcati arhiva in formular, acesta trebuie sa contina
  - numele si prenumele in **intregime** (ex: Popescu Marian Cristian)
  - grupa (242)
  - adresa de e-mail (nume@s.unibuc.ro)

iar trimiterea sursei sa se faca utilizand **adresa institutionala**

- **Nerespectarea instructiunilor anterioare atrage dupa sine nepromovarea examenului**
- Timp de lucru: **2 ore**
- Se acorda **1 punct din oficiu**