

### Tema 3

Sa se realizeze o aplicatie pentru **gestiunea unui hotel**, folosind WPF, C# si SQL Server.

Aplicatia va avea 4 tipuri de utilizatori: utilizator neautentificat (care vizualizeaza informatii in aplicatie fara a avea cont de utilizator), administrator, client si angajat.

Pe prima pagina a aplicatiei va fi un modul de cautare si un modul de creare cont / autentificare.

**Administratorul** este cel care poate adauga, modifica si sterge camere, lista de dotari pentru tipurile de camere, lista de imagini<sup>1</sup> asociate camerelor, serviciile suplimentare ale hotelului, preturi pentru tipurile de camere si pentru serviciile suplimentare, oferte.

**Utilizatorul neautentificat** poate vizualiza tipurile de camere impreuna cu informatiile aferente (dotari, imagini, preturi, numar de camere disponibile). Acesta poate face si o cautare pentru o anumita perioada si va vizualiza pretul pentru toata perioada aleasa pentru toate tipurile de camere disponibile in perioada respectiva.

**Clientul (utilizator autentificat)**, pe langa functionalitatea utilizatorului neautentificat, mai poate face rezervare, poate vizualiza ofertele hotelului si poate vizualiza istoricul rezervarilor sale. Pentru a efectua o rezervare se va specifica numarul de camere, tipul de camera (tipurile – in cazul in care rezerva mai multe camere de tipuri diferite), perioada in care doreste sa se cazeze (daca a ales mai multe camere, toate vor fi in aceeasi perioada), serviciile suplimentare pentru care opteaza (mic dejun, demipensiune, all inclusive, transport aeroport etc.). Clientul autentificat poate selecta si oferta (Ex: „Oferta de Pasti” 22 – 25 aprilie → 750 lei pentru 3 nopti cu mic dejun si cina incluse, „Oferta de 1 Mai” 30 apr – 2 mai → 400 lei pentru 2 nopti cu gratar inclus. Dupa selectarea tuturor optiunilor, clientul va vizualiza si costul total al rezervarii. Clientul va putea rezerva o camera doar daca aceasta este disponibila in perioada aleasa. ). Ofertele pot fi selectate deasemenea, doar in cazul in care mai exista camere disponibile in acea perioada. Clientul va putea deasemenea sa anuleze o rezervare activa (**rezervare activa = rezervare care nu a ajuns la data cazarii si care nu a fost platita**).

Dupa trecerea datei rezervarii, aceasta nu va mai putea fi modificata de catre nici un utilizator.

**Angajatul hotelului**, pe langa functionalitatile de utilizator neautentificat, va mai putea vizualiza rezervarile si va putea modifica starea unei rezervari active (prin **starea unei rezervari** se va intelege: **activ, anulat, achitat**)

---

<sup>1</sup> Imaginile se vor salva in baza de date

Se vor mai avea in vedere urmatoarele lucruri:

1. Preturile se vor modifica in timp, asa ca, daca perioada de rezervare contine o schimbare de pret, se va calcula cu doua preturi. Istoricul preturilor va fi inregistrat cu perioadele lor de valabilitate.
2. Dupa ce se face rezervarea unei/unor camere se va actualiza in baza de date numarul de camere disponibile pe intervalul de timp in care au fost rezervate.
3. Nu se va face stergerea fizica a datelor din baza de date, ci stergere logica (veti pune in tabelele din baza de date un camp „Deleted” care va avea valorile True sau False). Se vor lua in considerare pentru afisari si calcul de preturi doar datele din tabele, pentru care acel camp are valoarea False.

#### **Barem orientativ:**

1. Operatiile de CRUD (Create, Read, Update, Delete) pe tabele (functionalitatea administratorului) – **4p**
2. Functionalitate utilizator neautentificat (vizualizarea camerelor disponibile cu toate informatiile aferente) – **2p**
3. Functionalitate utilizator autentificat (rezervarea clasica; rezervare cu oferta; actualizarea bazei de date in urma rezervarilor; anulare rezervare; vizualizare istoric rezervari) – **2p**
4. Functionalitate Angajat – **1p**
5. Oficiu – **1p**

#### **Observatii:**

1. Timp de lucru – 3 saptamani. Predarea temei se va face in **saptamana 23 – 27 mai**, fiecare student la grupa sa.
2. Baza de date trebuie sa fie cel putin in **forma normala 3** (-3p in caz contrar)
3. Aplicatia trebuie **structurata pe niveluri** (-2p in caz contrar)
4. **Nu se vor lua in considerare** aplicatiile care nu sunt realizate pe modelul **MVVM** si care nu utilizeaza **Data Binding**.
5. Cei care folosesc Entity Framework vor realiza minim **5 proceduri stocate, dintre care minim doua mai complexe**, iar cei care utilizeaza ADO.NET vor opera pe baza de date

(inserare, modificare, stergere, selectie, etc) **doar prin proceduri stocate (-2p** in caz contrar)

6. Daca tema nu a obtinut **minim nota 5**, nu se va mai incheia situatia laboratorului.
7. Evitare **SQL Injection (-3p** in caz contrar)
8. Nu se vor afisa valorile id-urilor din tabele pe interfata grafica (**-2p** in caz contrar)