Proiect 1 - Gestionarea Comenzilor de Pizza

Introducere:
Să se dezvolte un sistem software implementat în C# (aplicație consolă) pentru gestionarea comenzilor de pizza ale unei pizzerii. Sistemul va permite administrarea meniului, gestionarea comenzilor și procesarea plăților. În același timp, aplicația va oferi funcționalități de raportare pentru administratori.

Contextul Sistemului

1. **Pizzeria**:
   - Are un nume, o adresă, și un meniu cu sortimente de pizza.
   - Acceptă comenzi plasate de clienți, care pot fi pentru livrare sau ridicare din locație.
   - Stochează toate comenzile înregistrate într-o bază de date sau fișier.
2. **Pizza**:
   - Fiecare tip de pizza are un nume, dimensiune (mică, medie, mare – tip enumerare), preț de bază, și o listă de ingrediente.
   - Se pot crea două categorii principale de pizza:
     - **Standard**: are un preț fix în funcție de dimensiune.
     - **Personalizată**: clientul poate alege ingredientele dorite; prețul se calculează în funcție de ingredientele selectate + 30 RON.
   - Ingredientele au un nume și un cost asociat.
3. **Clienți**:
   - Fiecare client are un nume, un număr de telefon, și un istoric de comenzi. Numărul de telefon trebuie să respecte formatul unui număr de telefon din România.
   - Dacă clientul nu este autentificat, poate doar vizualiza meniul.
   - Clienții fideli (care au plasat cel puțin 5 comenzi) beneficiază de o reducere de 10%.
4. **Comenzile**:
   - O comandă include: clientul, lista de pizza comandate, metoda de livrare (ridicare sau livrare la domiciliu) și costul total.

- Livrarea la domiciliu adaugă un cost fix de 10 RON.
- Comenzile trebuie validate înainte de procesare (ex: pizza din comandă trebuie să existe în meniu).

Funcționalități

Aplicația va permite:

1. **Autentificare și înregistrare**:
   - Înregistrarea unui client nou.
   - Autentificarea utilizatorilor existenți.
2. **Gestionarea meniului (doar pentru administratori)**:
   - Vizualizare lista completă a sortimentelor disponibile.
   - Adăugare/ștergere sortimente de pizza.
   - Modificarea informațiilor despre un sortiment (ex: lista de ingrediente, preț).
   - Vizualizare lista ingrediente
   - Modificare preț pentru un anumit ingredient
   - Adăugare/ștergere ingredient
3. **Vizualizare istoric comenzi:**
   - Istoric comenzi pizzerie
   - Vizualizarea istoricului comenzilor pentru un anumit client.
4. **Gestionarea comenzilor**:
   - Vizualizarea meniului și plasarea unei comenzi de către clienți.
   - Calcularea prețului total al comenzii (inclusiv eventuale reduceri sau costuri de livrare).
5. **Raportare și statistici (doar pentru administratori)**:
   - Vizualizarea comenzilor finalizate într-o anumită zi.
   - Raport cu cele mai populare sortimente de pizza comandate.
   - Raport cu totalul veniturilor generate de comenzi într-o anumită perioadă.

Cerințe Tehnice

1. **Pentru nota maximă 8**:
   - Crearea unui model OO clar, utilizând corect conceptele POO (încapsulare, moștenire, polimorfism, compoziție).
   - Salvarea și încărcarea stării aplicației într-un fișier (informații despre meniul și comenzile existente, etc).
   - Tratarea situațiilor de eroare (ex: fișiere lipsă, date invalide).
   - Utilizarea GitHub pentru versionare și colaborare.
   - Documentarea și prezentarea evoluției aplicației.
2. **Pentru nota maximă 10**:
   - Toate cerințele de mai sus.
   - Izolarea funcționalităților externe folosind clase wrapper (pentru interacțiunea cu fișiere și consola).
   - Utilizarea **.NET Core GenericHost** pentru gestionarea dependențelor.
   - Logging folosind **ILogger** pentru colectarea informațiilor despre erori sau evenimente.
3. **Cerințe opționale (puncte bonus)**:
   - Stocarea alternativă a datelor într-o bază de date SQL.
   - Utilizarea expresiilor LINQ pentru manipularea colecțiilor.
   - Adăugarea unei funcționalități de notificare (email sau SMS) pentru confirmarea comenzilor.
   - Implementarea unui modul de feedback pentru clienți (evaluarea comenzilor sau a sortimentelor).
   - Crearea unui „framework” pentru a extinde aplicația la alte tipuri de restaurante.
   - Crearea de teste unitare pentru validarea funcționalității claselor principale.

Proiect 2 – Rezervări zboruri

Să se dezvolte un sistem software, implementat în C# (aplicație consolă), pentru gestionarea rezervării de zboruri ale unei companii aeriene. Sistemul trebuie să permită gestionarea zborurilor, înregistrarea clienților, realizarea de rezervări, calcularea costurilor.

Contextul Sistemului

1. **Compania aeriană**:
    - Are un nume, o flotă de avioane și o listă de rute disponibile. Pentru o rută se cunosc: oraș de plecare, oraș de destinație și nr. de kilometri.
    - Permite rezervarea biletelor pentru pasageri.
    - Înregistrează toate rezervările efectuate, inclusiv plățile.
2. **Zborurile**:
    - Fiecare zbor este identificat unic printr-un cod (ex. RO101).
    - Se cunosc ruta, ora plecării, durata zborului, capacitatea avionului, și numărul de locuri disponibile.
    - Se pot organiza două tipuri de zboruri:
        - **Zboruri interne**: preț bilet calculat ca (50 RON + 0.5 RON/km).
        - **Zboruri internaționale**: preț bilet calculat ca (200 RON + 1 RON/km).
3. **Pasagerii**:
    - Fiecare pasager are un nume, un cod unic (CNP), și o listă de rezervări anterioare. CNP-ul trebuie să respecte formatul unui CNP românesc.
    - Pasagerii trebuie să aibă un cont creat în sistem (username și parola unice).
    - Dacă pasagerul nu este autentificat, poate doar să vadă lista zborurilor disponibile.
4. **Rezervările**:
    - O rezervare este asociată unui pasager, unui zbor, și unui număr de locuri rezervate.
    - Se calculează costul total în funcție de numărul de locuri rezervate și tipul zborului.

Funcționalități

Sistemul trebuie să permită:

1. **Autentificare și înregistrare**:
    - Înregistrarea unui nou cont de pasager.
    - Autentificarea utilizatorilor existenți.
2. **Gestionare zboruri (doar pentru administratori)**:
    - Adăugare zbor nou.
    - Ștergere zbor.
    - Vizualizare lista completă de zboruri.
    - Actualizare informații despre zboruri (ex: ora plecării, capacitatea).
    - Vizualizare rute disponibile
    - Adăugare/ștergere rute
3. **Rezervări pentru pasageri**:
    - Vizualizare lista de zboruri disponibile (cu locuri libere).
    - Rezervare locuri pe un zbor specific.
    - Anulare rezervare.
    - Vizualizare istoricul rezervărilor proprii.
4. **Rapoarte și statistici (doar pentru administratori)**:
    - Vizualizare zboruri cu cele mai multe locuri rezervate.
    - Vizualizare veniturile generate de un zbor.
    - Generare raport zilnic al veniturilor totale.
    - Vizualizarea tuturor plăților efectuate de un pasager.

Cerințe Tehnice

1. **Nota maximă 8**:
    - Crearea unui model OO clar, utilizând corect principiile POO (încapsulare, moștenire, polimorfism, compoziție).
    - Salvarea și încărcarea stării aplicației într-un fișier.
    - Tratarea erorilor (ex: fișier inexistent, date invalide, locuri insuficiente pentru rezervare).
    - Utilizarea GitHub pentru colaborare între membri, cu commit-uri relevante și bine descrise.
    - Prezentarea design-ului și evoluției aplicației (slide-uri PPT).
2. **Nota maximă 10**:
    - Toate cerințele de mai sus.
    - Izolarea funcționalităților externe folosind clase învelitoare (wrapper) pentru interacțiunea cu fișiere și consola.
    - Utilizarea **.NET Core GenericHost** pentru gestionarea dependențelor.
    - Implementarea unui mecanism de logging folosind **ILogger**.
3. **Opțional (puncte bonus)**:
    - Stocarea alternativă a datelor într-o bază de date SQL.
    - Utilizarea expresiilor LINQ pentru manipularea colecțiilor.
    - Adăugarea unui mecanism de notificare prin email (simulat) pentru pasageri (ex: notificare pentru plata restantă).
    - Implementarea modelului arhitectural MVC.
    - Crearea unor teste unitare pentru validarea funcționalității aplicației.