**Detalii de implementare pentru proiect final C#**

1. Intrucat string-ul de conectare a aplicatiei la serverul bazei de date este hard-codat, acesta va trebui schimbat pentru a se potrivi cu versiunea VS de pe masina locala. Fisierul **DAL/Class1.cs**, linia 17:

@"Data Source=(localdb)\MSSQLLocalDB;Initial Catalog=master; Integrated Security=True;Connect Timeout=30; Encrypt=False; TrustServerCertificate=True; ApplicationIntent=ReadWrite; MultiSubnetFailover=False"

De asemenea, serverul local MSSQL va trebui sa contina un obiect (schema) numit Training, cu un tabel numit Agenda, structura si datele din tabel fiind precizate in fisierul **Training.sql** atasat acestui proiect

1. metodele care populeaza obiectul dataGridView1 la incarcare (**FrmMainWindow\_Load**) si la click pe butonul **button1** (Load) sunt**:**

publicList<Abonati> Select() (**BAL/Class1.cs, linia 40**) este folosita in prima situatie.

publicDataTable Select2() (acelasi fisier**, linia 71**) estefolosita in cea de-a doua situatie.

Comportamentul acestor metode diferite este, de asemenea, diferit. La incarcarea ferestrei FrmMainWindow, grid-ul este populat dar **nu este editabil inca** (dataGridView1.ReadOnly = true, fisierul **Form1.cs**, **linia 73**). In plus, butonul **Save** (**button2**) este inactiv. Grid-ul va trece pe modul editabil iar butonul **Save** va deveni **enabled** dupa ce user-ul face click pe **button1**.

1. Constructorul fara parametri al obiectului de clasa **DAbonati (DAL/Class1.cs, linia 21)** este cel care realizeaza conexiunea la baza de date in modul disconnected layer. Tot acest constructor populeaza campul DataTable dt, care va fi utilizat in **BAL** (fie va fi procesat pentru crearea unei liste generice cu tipul “Abonati”, **List<Abonati>,** fie va fi pasat mai departe pentru initializarea **dataGridView1.DataSource**) in scopul popularii, in GUI, a grid-ului.

Constructorul **cu parametri** din DAL (**DAL/Class1.cs, linia 40**) este utilizat pentru interogari **Select** in baza de date folosind clauza **Where.** Interogarile sunt initiate in FrmSearchWindow (fisierul **Form2.cs**)

Metoda publicDataTableSelect() (**DAL/Class1.cs, linia 66**) este utilizata atat de catre metoda care populeaza grid-ul la incarcarea aplicatiei, cat si de cealalta metoda din BAL, care populeaza gridul la click pe **button1** (Load).

1. Exceptiile sunt in general tratate local (cele mai multe GUI, fisierul **Form1.cs**) dar sunt si cazuri in care sunt “aruncate” mai departe, pentru a fi tratate in codul apelant (fisierul **BAL/Class1**, **liniile 58 – 68**). Exceptia ridicata de incercarea de a altera cheia primara cu valori non-integer este tratata de un handler **pe linia 121**, in fisierul **Form1.cs**.
2. Update-ul bazei de date este initiat intr-un singur loc, in dataGridV, iar modificarile sunt salvate dupa apasarea butonului Save (fisierul **Form1.cs**, **linia 105**). Update-ul efectiv se realizeaza in **DAL/Class1.cs, liniile 71 – 74**.
3. In afara de popularea grid-ului cu datele din tabelul “Agenda” si de posibilitatile de a modifica baza de date, celelalte cerinte ale proiectului sunt implementate dupa cum urmeaza:
4. **Meniu cu trei rubrici** (MenuStrip + 3 x ToolStripMenuItem): fisierul **Form1.cs**, **liniile** **128 – 162**;
5. **Serializare si salvare** date din grid in fisier serializat, acelasi fisier, **liniile 166 – 188**;
6. **Exit aplicatie** cu optiune Da/Nu, acelasi fisier, **liniile 155 – 162**;
7. **Meniu cu o rubrica** (MenuStrip + ToolStripMenuItem) pentru cautare persoana. Aici este lansata (la click pe “Cautare persoana”) fereastra FrmSearchWindow – liniile **197 – 203** din **Form1.cs.**
8. **FrmSearchWindow** este implementata in **Form2.cs,** are propriul dataGridView, needitabil, dispune de mai multe controale pentru definirea cautarii (select cu parametri) si, fiind parte a GUI, poate apela direct doar obiecte/metode din BAL.
9. **Referintele** sunt puse pentru a respecta implementarea pe 3 layere: GUI (adica FrmMainWindow si FrmSearchWindow) referentiaza BAL, care referentiaza DAL.
10. Codul este comentat in mai multe locuri, pentru a usura lizibilitatea.