

Laboratorul 6

Lucrul cu baze de date relaționate în Visual Basic .NET

Ce ne propunem astăzi?



În acest laborator ne propunem să implementăm în Visual Basic .NET o aplicație destinată unui cabinet medical de radiologie, care va trebui să păstreze o evidență a pacienților și a imaginilor radiologice ale acestora. Aplicația va permite adăugarea și vizualizarea tuturor imaginilor radiologice ale unui pacient. Toate aceste date vor fi păstrate într-o bază de date relaționată.

Aplicația va opera cu datele dintr-o bază de date (Pacienti.mdb) care conține două tabele relaționate (tabelele Pacienti și Radiografii). Pe fereastra principală va fi afișată sub formă tabelară o listă a tuturor pacienților care au efectuat consultații în cadrul cabinetului medical. Se va prevedea posibilitatea de căutare a pacienților după nume. Operațiile de adăugare de pacienți noi sau de modificare a datelor pacienților existenți se vor efectua direct în controlul tabelar (DataGridView). Pentru fiecare pacient se vor putea adăuga imagini radiologice care vor putea fi apoi vizualizate sub formă de miniaturi într-o listă. La click pe o miniatură, imaginea radiologică se va încărca la dimensiune completă într-un control de tip PictureBox (Figura 1).

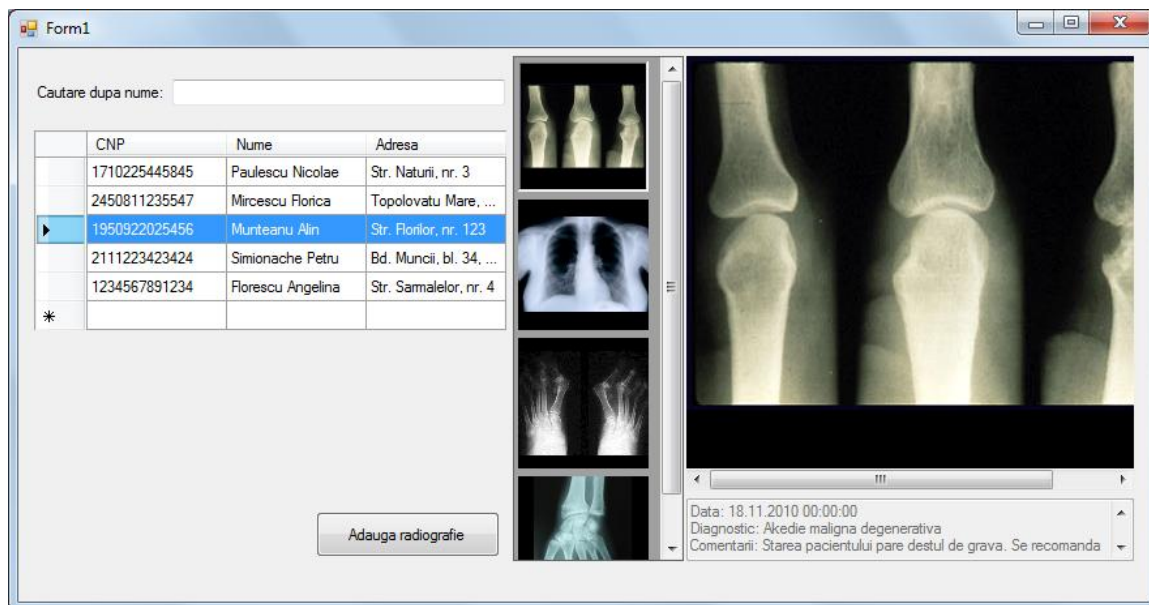


Figura 1. Interfața aplicației pentru cabinete medicale de radiologie

Mai pe larg, vom proceda astfel...

Baza de date va fi de tip MDB și poate fi creată în Microsoft Access. Aceasta va fi relaționată și va avea structura prezentată în Figura 2.

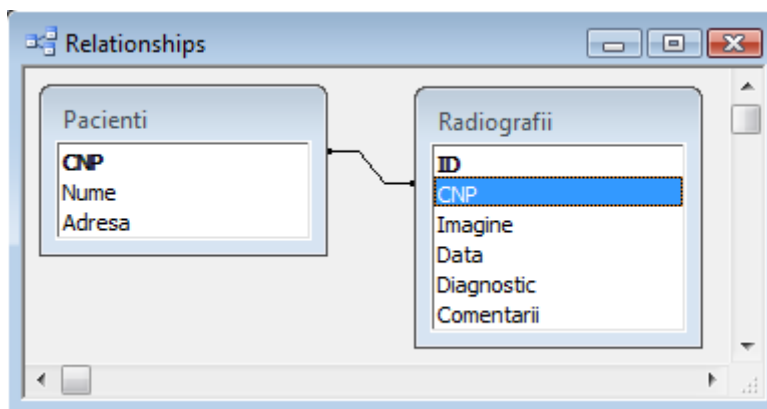
The image shows two screenshots of Microsoft Access table structures. The top screenshot shows the 'Pacienti' table with three fields: 'CNP' (Text, primary key), 'Nume' (Text), and 'Adresa' (Text). The bottom screenshot shows the 'Radiografii' table with six fields: 'ID' (AutoNumber, primary key), 'CNP' (Text), 'Imagine' (Text, description: 'numele fisierului'), 'Data' (Date/Time), 'Diagnostic' (Text), and 'Comentarii' (Text).

Field Name	Data Type
CNP	Text
Nume	Text
Adresa	Text

Field Name	Data Type	Description
ID	AutoNumber	
CNP	Text	
Imagine	Text	numele fisierului
Data	Date/Time	
Diagnostic	Text	
Comentarii	Text	

Figura 2. Structura tabelor bazei de date Pacienti

Relația dintre cele două tabele se va crea tot din Microsoft Access (Database Tools -> Relationships). Relația se va defini între câmpul CNP din tabelul Pacienti (câmp care este de tip primary key) și câmpul CNP din tabelul Radiografii prin drag-and-drop (Figura 3).



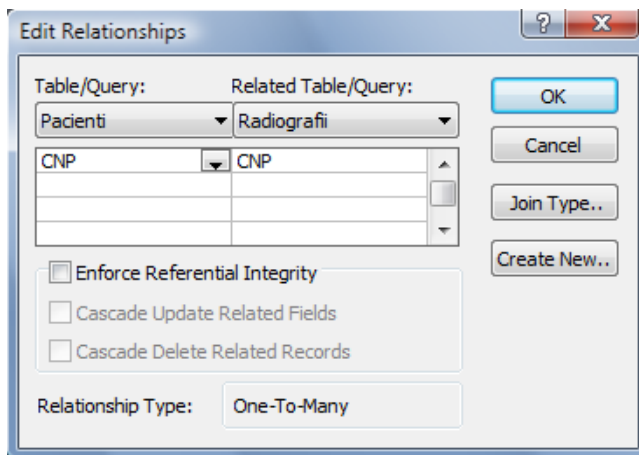


Figura 3. Relația dintre tabelele Pacienti și Radiografii

Relația dintre tabelele Pacienti și Radiografiile va fi de tip one-to-many, iar ca o consecință imediată a acesteia datele relaționate pot fi vizualizate mai ușor (Figura 4).

CNP	Nume	Adresa	ID	Imagine	Data	Diagnostic	Comentarii
1234567891234	Florescu Angeli	Str. Sarmalelor,	7	13.jpg	1.2010 15:06:09	Anevrism pulmc	cauzat de fuma
			10	02.jpg	1.2010 16:04:20	Fractura la oas	Se recomanda
			11	05.jpg	1.2010 16:07:11	Artrita reumatoi	A umblat descu
			*	(AutoNumber)			
1710225445845	Paulescu Nicol	Str. Naturii, nr.	1	02.jpg	22.11.2010	Fibrom glandula	Radiografie obit
			2	10.jpg	23.11.2010	Anomalie fibrila	Pacientul s-a pi
			*	(AutoNumber)			
1950922025456	Munteanu Alin	Str. Florilor, nr.					
2111223423424	Simionache Pe	Bd. Muncii, bl.					
2450811235547	Mircescu Floric	Topolovatu Mar					
			*				

Figura 4. Vizualizarea datelor din baza de date relaționată

Legarea aplicației la baza de date

Putem asigura legarea aplicației la baza de date fără a scrie nicio linie de cod urmând pașii descriși în continuare.

În Visual Studio se selectează din meniul Data comanda „Add New Data Source”. Se configurează o nouă conexiune la o bază de date. Se alege baza de date Microsoft Access creată anterior (Figura 5).

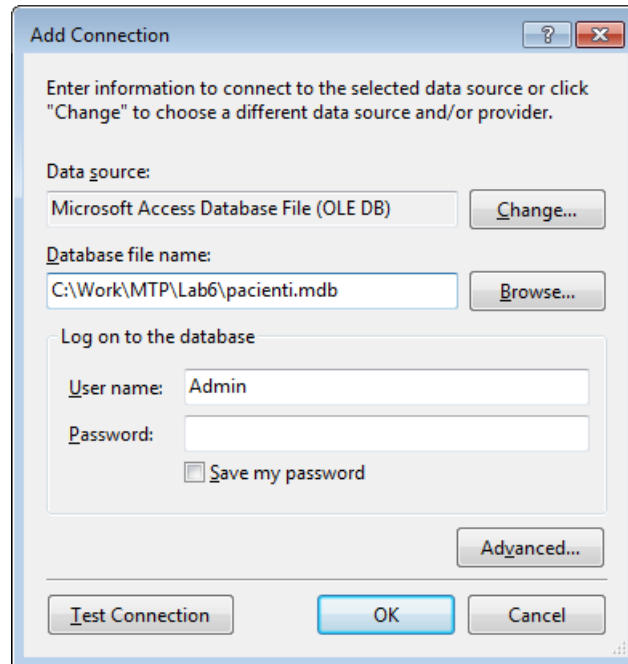


Figura 5. Alegerea bazei de date

La pasul următor vom fi informați că baza noastră de date este locală și vom fi întrebați dacă vrem să o adăugăm la proiectul nostru (Figura 6). Dacă selectăm „Yes”, baza de date va fi inclusă în proiect și va apărea în fereastra *Project Explorer*, iar la fiecare compilare a proiectului baza de date va fi copiată în directorul care conține fișierul executabil. Dacă selectăm „No”, atunci aplicația va lucra cu baza de date aflată într-o locație fixă.

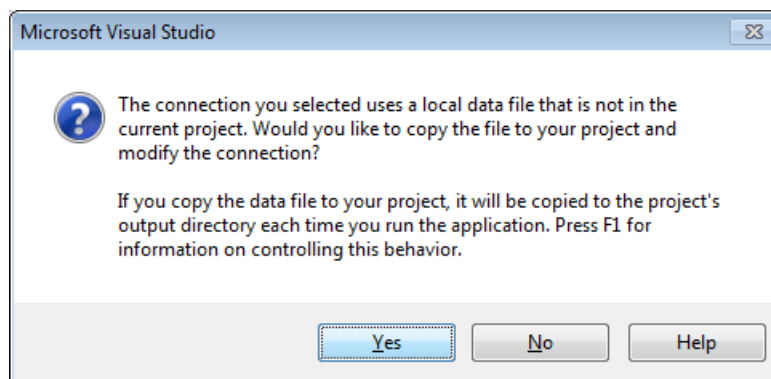


Figura 6. Dialogul pentru adăugarea bazei de date la proiectul curent

În continuare, se parcurge wizardul până la dialogul „Choose your database objects”, în care vom selecta ambele tabele ale bazei de date (Figura 7).

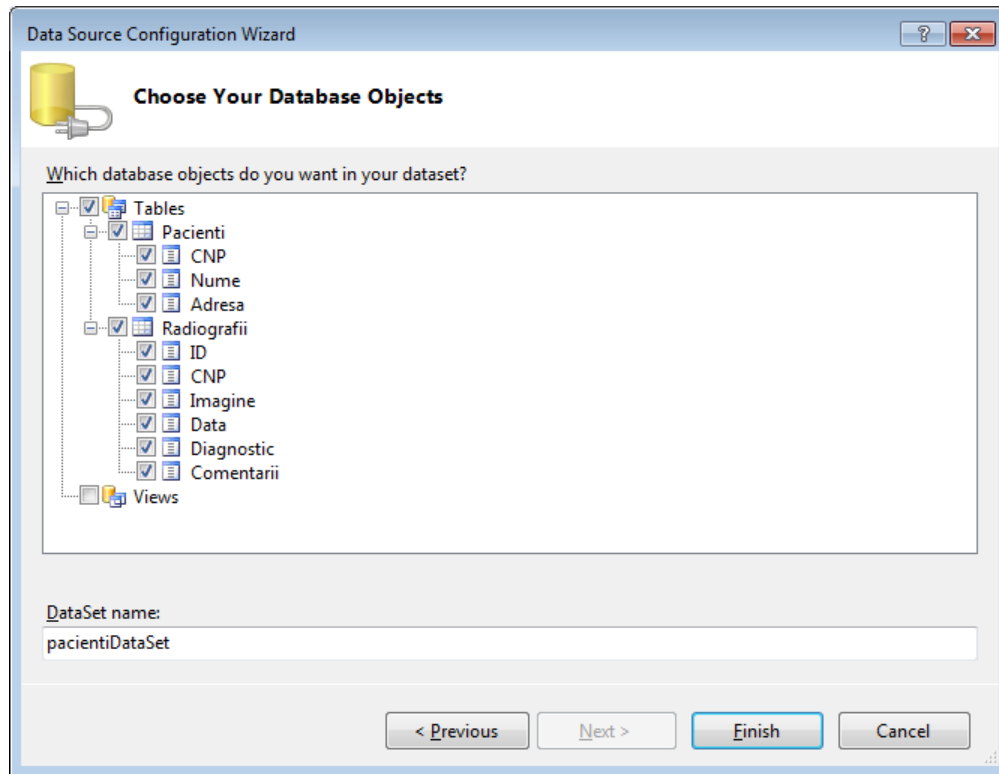


Figura 7. Selectarea obiectelor pe care dorim să le includem în DataSource.

După finalizarea wizardului, se poate observa în fereastra Data Sources că tabelul Radiografii este listat atât ca nod independent, cât și ca nod fiu al tabelului Pacienti, deoarece există o relație “părinte-copil” între cele două (Figura 8).

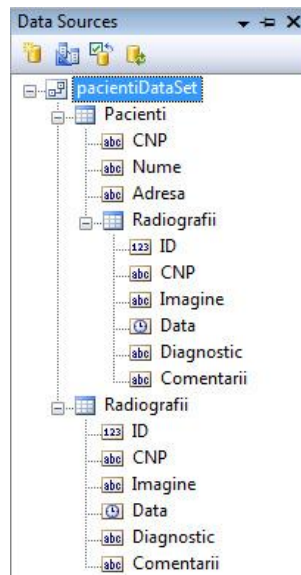


Figura 8. Fereastra Data Sources.

După ce am creat sursa de date putem construi extrem de ușor aplicații pentru lucrul cu baza de date. Trebuie doar să efectuăm operații drag-and-drop cu obiectele din fereastra Data Sources pe un formular.

Putem face click pe un tabel din Data Sources apoi drag-and-drop pe un formular iar Visual Basic va crea automat controalele BindingNavigator, DataGridView și alte controale și componente pentru afișarea datelor din tabel. Datele din tabel pot fi afișate în mod tabelar (control DataGridView) sau detaliat (datele din fiecare câmp sunt afișate în controale separate). Modul de afișare poate fi ales din meniul ce va fi afișat la click pe un tabel din fereastra Data Sources (Figura 9).

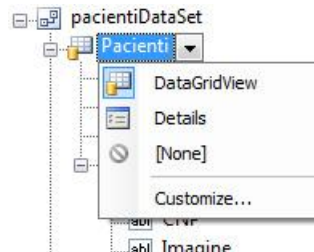


Figura 9. Setarea modului de afișare a datelor dintr-un tabel

În fereastra DataSource vom alege tabelul Pacienti (vizualizare tabelară) și îl vom trage peste formular (drag-and-drop), apoi vom face același lucru pentru tabelul Radiografii (vizualizare tabelară), cel afișat ca nod fiu al tabelului Pacienti. Împreună cu cele două controale de tip DataGridView va fi adăugat automat pe formular un control de tip BindingNavigator și o serie de alte componente utile pentru lucrul cu datele din cele două tabele.

În acest moment se poate rula aplicația, putându-se observa legătura dintre cele două tabele ale bazei de date: la selectarea unui pacient din controlul PacientiDataGridView în controlul RadiografiiDataGridView vor fi afișate numai radiografiile asociate pacientului selectat.

După aceasta vom șterge de pe formular controlul PacientiBindingNavigator (nu avem nevoie de el) și controlul RadiografiiDataGridView (avem nevoie doar de componentele create odată cu acesta: RadiografiiBindingSource și RadiografiiTableAdapter). În final, pentru interacțiunea cu baza de date, vor rămâne pe formular următoarele controale și componente:

- PacientiDataGridView
- PacientiDataSet
- PacientiBindingSource
- PacientiTableAdapter
- TableAdapterManager
- RadiografiiBindingSource
- RadiografiiTableAdapter.

Reamintim faptul că atât operația de adăugare a unui pacient cât și cea de modificare a datelor acestuia se vor efectua direct în controlul DataGridView.

Iată o scurtă descriere a rolului pe care îl are fiecare tip de control/componentă utilizat pentru operarea cu datele din baza de date:

- Un obiect *DataSet* reprezintă o întreagă bază de date. Acesta conține obiecte de tip *DataTable* care reprezintă tabelele din baza de date. Fiecare obiect *DataTable* conține obiecte *DataRow* care reprezintă înregistrările din baza de date, iar fiecare obiect *DataRow* reprezintă câte o coloană a unei înregistrări.
- Obiectele de tip *TableAdapter* sunt utilizate pentru comunicația dintre aplicație (*DataSet*) și o bază de date. Acestea oferă metode pentru efectuarea operațiilor asupra

- bazei de date. Obiectele *TableAdapter* mai sunt utilizate pentru a trimite date actualizate de la aplicația curentă înapoi la baza de date.
- *TableAdapterManager* reprezintă o componentă nouă în Visual Studio 2008 și oferă funcții de salvare a datelor în tabele relaționate.
 - Obiectul *BindingSource* încapsulează toate datele din *DataSet* și oferă funcții pentru controlul acestora din cadrul programului.
 - *BindingNavigator* oferă o interfață grafică pentru ca utilizatorul să poată controla *BindingSource*.

Afișarea imaginilor și crearea dinamică a controalelor

Pentru a putea afișa sub formă miniaturală imaginile radiologice asociate unui pacient va trebui să utilizăm un control container de tip *FlowLayoutPanel* și o modalitate de creare dinamică a unor controale de tip *PictureBox*, câte unul pentru fiecare imagine radiologică.

Iată un exemplu de creare dinamică a unui control și de adăugare a unui handler de eveniment (procedură de tratare a unui eveniment).

```
Dim p As PictureBox
p = New PictureBox

'Setarea proprietatilor controlului:
p.Name = "nume_control"
p.SetBounds(x, y, width, height)
p.BackColor = Color.Black
p.SizeMode = PictureBoxSizeMode.Zoom
p.Image = Bitmap.FromFile(nume_fisier)

'Adaugarea controlului nou creat la colectia de controale a containerului:
FlowLayoutPanel1.Controls.Add(p)

'Adaugarea unui handler pentru evenimentul Click:
AddHandler p.Click, AddressOf Radiografie_Click
```

Procedura *Radiografie_Click* va fi creată manual și va avea următoarea formă:

```
Private Sub Radiografie_Click(ByVal sender As System.Object, ByVal e As EventArgs)
End Sub
```

Parametrul *sender* poate oferi informații despre controlul *PictureBox* pentru care a fost apelat:

```
Dim myPic As PictureBox = CType(sender, PictureBox)
```

Accesul la datele din tabelul Radiografiei se face astfel:

```
Dim dvw As DataView = RadiografiiBindingSource.List
Dim r As DataRowView = dvw.Item(0) 'obținerea primei înregistrari din tabel
data=r("Data") 'obținerea valorii din câmpul 'Data'
```

Controlul container *FlowLayoutPanel* permite adăugarea automată a unor bare de derulare în cazul în care conținutul acestuia depășește marginile controlului container (proprietatea *AutoScroll* setată pe *True*). Același mecanism va fi utilizat și în cazul afișării imaginii radiologice la dimensiune completă (control *PictureBox* în interiorul unui control container *Panel*).

Afișarea imaginilor sub formă de miniatură se face prin dimensionarea controlului *PictureBox* la lățimea și înălțimea dorite, apoi prin setarea proprietății *SizeMode* la valoarea *Zoom*. Astfel, imaginea încărcată în controlul *PictureBox* va fi afișată astfel încât să fie cuprinsă în întregime pe suprafața controlului, indiferent de dimensiunile acestuia (dacă imaginea are dimensiuni mai mari decât controlul, atunci aceasta va fi micșorată la dimensiunile controlului, iar dacă

imaginea are dimensiuni mai mici decât controlul, atunci aceasta va fi mărită la dimensiunile controlului).

Afișarea imaginii radiologice la dimensiune completă (împreună cu celelalte detalii, vezi Figura 1) se va face la click pe una dintre imaginile miniaturale. Imaginea miniaturală selectată își va schimba aspectul prin setarea proprietății *BorderStyle* la valorile *None* sau *Fixed3d*. Pentru ca o imagine să fie afișată la dimensiunile ei normale, vom seta proprietatea *SizeMode* a controlului *PictureBox* la valoarea *AutoSize*.

După cum am amintit la început, aplicația va prevedea posibilitatea adăugării unei imagini radiologice noi pentru un pacient. La apăsarea butonului corespunzător se va deschide o fereastră dialog în care utilizatorul va putea indica toate detaliile asociate imaginii radiologice (Figura 10). Câmpul CNP-ul va fi precompletat, valoarea acestuia putând fi obținută astfel:

```
PacientiDataSet.Tables("Pacienti").Rows(PacientiBindingSource.Position)("CNP")
```

Presupunem că toate imaginile radiologice sunt salvate automat de către echipamentul medical într-un anumit director.

The image shows a Windows-style dialog box titled "dlgAduagare". It contains several input fields: "CNP" with the value "1950922025456", "Data" with a date picker set to "29.11.2010", "Imagine" with a text box containing "06.jpg" and a folder icon, "Diagnostic" with a text box containing "Artrita reumatoida", and "Comentarii" with a text area containing "Se recomanda tratament urgent". At the bottom right, there are "OK" and "Cancel" buttons.

Figura 10. Adăugarea detaliilor unei radiografii

Salvarea datelor

La apăsarea butonul OK va trebui ca în tabelul Radiografii să fie adăugată o nouă înregistrare. În lucrarea de laborator precedentă am învățat modul de adăugare a unei înregistrări în trei pași: inițierea comenzii de adăugare, adăugarea datelor prin intermediul controalelor de pe formular legate la baza de date și salvarea în baza de date a acestora. De această dată, vom efectua operația de adăugare într-un mod mai simplu, într-un singur pas. Ne vom folosi de componenta *RadiografiiTableAdapter*.

```
'Adaugarea datelor
RadiografiiTableAdapter.Insert(cnp, imagine, data, diagnostic, comentarii)
'Salvarea datelor
TableAdapterManager.UpdateAll(PacientiDataSet)
'Reincarcarea datelor
RadiografiiTableAdapter.Fill(PacientiDataSet.Radiografii)
```


Metoda *Insert* a obiectului de tip *TableAdapter* acceptă ca parametri valorile care să fie introduse în câmpurile noii înregistrări. Acești parametri trebuie indicați în ordinea în care apar câmpurile din tabel și trebuie să aibă același tip de date cu acestea.

Sfaturi utile



Cea mai simplă modalitate de efectuare a căutării unui pacient din baza de date este setarea corespunzătoare a proprietății *Filter* a unei componente de tip *BindingSource*. Pentru anularea filtrului de căutare se va apela metoda *RemoveFilter()* a componentei de tip *BindingSource*.

Accesul restricționat la datele aplicației

Considerăm că datele din aplicația de radiografie sunt date confidențiale, astfel că accesul la acestea ar trebui să fie protejat prin parolă.

Putem seta o parolă pentru baza de date din cadrul aplicației Microsoft Access (secțiunea *Database Tools* -> *Set Database Password*). În acest caz va trebui indicată parola de acces în wizard-ul pentru conectarea la baza de date, astfel încât stringul de conectare la baza de date să conțină parola de acces. **Pentru lucrarea de față NU se cere setarea unei parole pentru baza de date.**

În mod obligatoriu, accesul în aplicația de radiografie se va face prin autentificarea utilizatorului. În acest sens se va crea un nou formular de autentificare (precum cel din Figura 11).

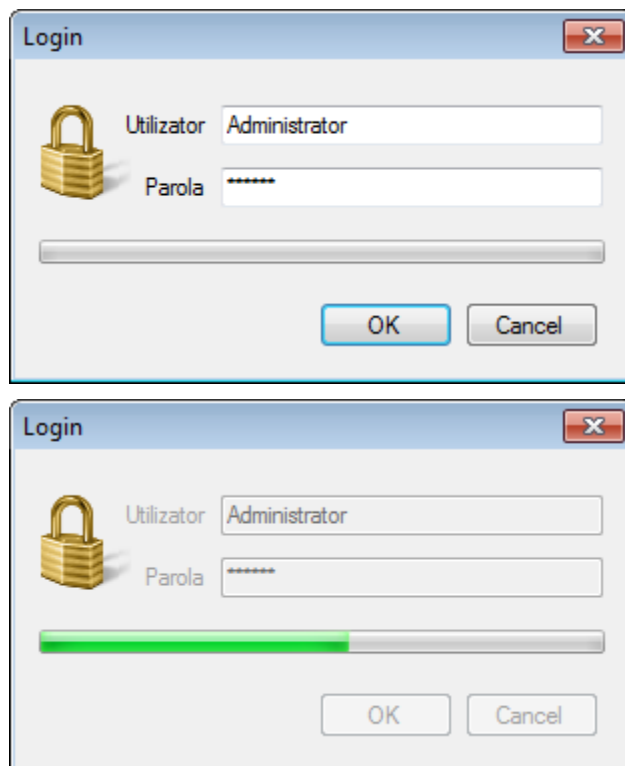


Figura 11. Fereastra de autentificare a utilizatorilor în aplicația de Radiografie, înainte și după autentificare

Pentru crearea și utilizarea acestei ferestre trebuie precizate câteva lucruri.

1. Fereastra trebuie setată ca fereastră de start, astfel încât, la pornirea aplicației aceasta să fie fereastra afișată. Acest lucru poate fi setat în pagina de proprietăți a proiectului. Tot de aici se va seta modul de închidere a aplicației (Shutdown mode: When last form closes).
2. Controlul TextBox asociat parolei va trebui să ascundă textul introdus prin setarea proprietății *PasswordChar* a acestuia.
3. Fereastra va utiliza o componentă Timer și un control de tip ProgressBar. Acesta din urmă permite indicarea progresului unei operații. Se vor utiliza proprietățile *Minimum* (0), *Maximum* (20) și *Value* (inițial având valoarea 0). Se dorește ca la autentificarea cu succes a utilizatorului controlul ProgressBar să se „umple” treptat, indicând posibile operații efectuate în fundal (inițializări diverse, încărcare de date etc.)
4. Pentru componenta Timer va fi setată proprietatea *Interval* la valoarea 100. Aceasta indică numărul de milisecunde scurs între generarea a două evenimente Tick. La autentificarea utilizatorului, dacă numele utilizatorului și parola au fost introduse corect, se va activa componenta Timer (proprietatea *Enabled*), iar din acest moment, la fiecare 100 ms va fi generat câte un eveniment Tick. Handlerul acestui eveniment poate fi utilizat pentru a indica operații care trebuie efectuate periodic, cu o anumită temporizare. În cazul nostru, operația aceasta este cea de incrementare a valorii controlului ProgressBar, până la atingerea valorii maxime, situație în care componenta Timer va fi dezactivată, va fi încărcată fereastra principală a aplicației de radiografie, iar fereastra de autentificare va fi închisă.

Cu ce ne-am ales?



Prin aplicația dezvoltată în cadrul laboratorului de astăzi am reușit să ne familiarizăm cu utilizarea bazelor de date relaționate, să creăm dinamic controale și să le gestionăm cu ajutorul controalelor container.

Bibliografie



[1] <http://msdn.microsoft.com/en-us/vbasic/default.aspx>