

Capitolul 4

Prezentarea aplicației

4.2 Introducerea de date

Funcționalitățile aplicației sunt organizate într-o structură de tip tree care este permanent vizibilă în partea stângă a aplicației.

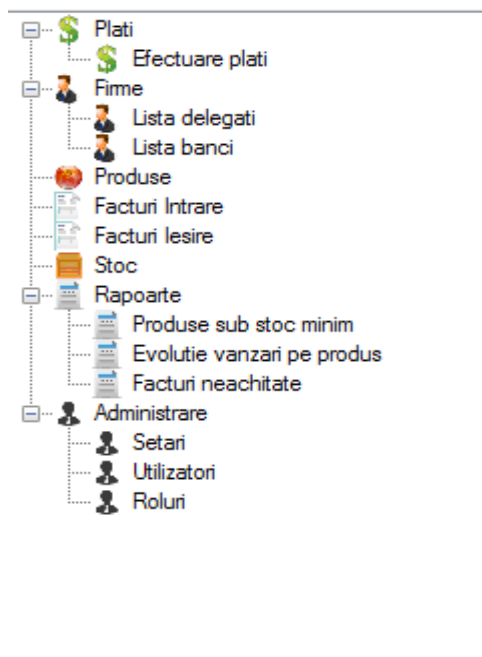


Figura 4.3. Tree-ul aplicației

Pentru folosirea aplicației trebuie introduse datele de bază, acestea sunt: băncile, delegații, firmele și produsele. Toate funcționalitățile aplicației se bazează pe aceste patru tipuri de dată. Fiecare clasă care are un echivalent în baza de date extinde clasa `GestiuneObject`, o clasă abstractă care conține metode pentru verificarea erorilor și o listă care reține numele fiecărei proprietăți. Această listă nu este implementată, este abstractă deoarece fiecare clasă ce derivă din ea trebuie să își enumere proprietățile.

Capitolul 4 - Prezentarea aplicației

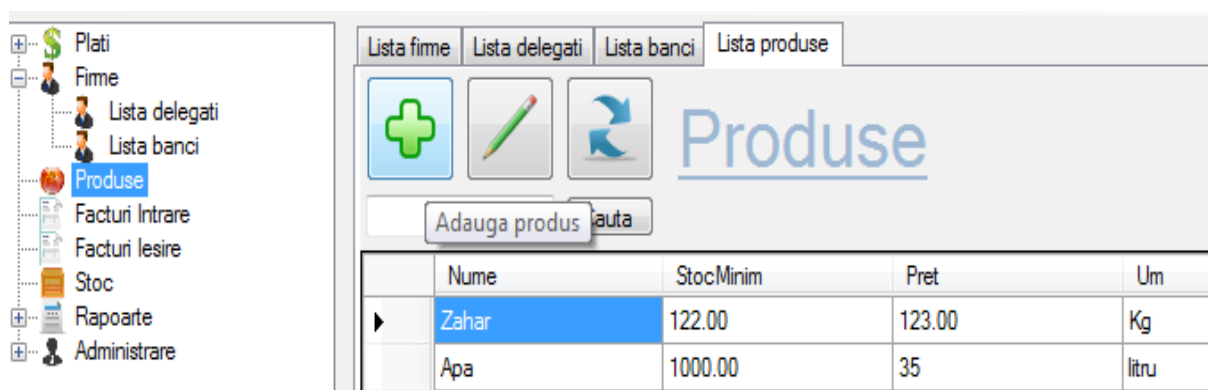


Figura 4.4. Deschiderea listei de produse

Pentru adăugarea unui produs trebuie să deschidem tabul cu lista produselor: din tree-ul din partea dreaptă (**Figura 4.3**) deschidem Produse și toate produsele sunt enumerate într-un grid, după care facem click pe butonul cu plus verde din **Figura 4.4**, în general în toată aplicația semnificația acestui buton este de adăugare, completăm datele necesare pentru un produs: numele, stocul minim, prețul și unitatea de măsură a produsului și apăsăm butonul de salvare din **Figura 4.5**.

Adauga produs

Nume

Stoc minim Pret

Unitate de masura

Iesire Salvare

Figura 4.5. Completarea datelor pentru adăugarea unui produs

La salvare se apelează codul funcției Create, care se ocupă de persistența bazei de date, mai exact operația de inserare:

```
public int Create(List<DBObject> dbObjects)
{
```

Capitolul 4 - Prezentarea aplicației

```
try
{
    using (SqlConnection con = ConnectionHelper.Connection)
    {
        try
        {
            using (SqlCommand cmd =
                new SqlCommand(insertStoredProcedureName, con))
            {
                const string ReferenceId = "@ReferenceID";
                con.Open();
                cmd.CommandType = System.Data.CommandType.StoredProcedure;
                cmd.Parameters.Add(ReferenceId, System.Data.SqlDbType.Int).Direction =
System.Data.ParameterDirection.ReturnValue;

                foreach (var dboObject in dbObjects)
                {
                    cmd.Parameters.AddWithValue(dboObject.Name, dboObject.Value);
                }

                cmd.ExecuteReader();

                var id = cmd.Parameters["@ReferenceID"].Value;
                return (int)id;
            }
        }
        catch (Exception ex)
        {
            throw new Exception(StringProcedureFail + insertStoredProcedureName, ex);
        }
    }
}
```

Capitolul 4 - Prezentarea aplicației

```
}  
catch (Exception ex)  
{  
    throw new Exception(StringDatabaseFail, ex);  
}  
}
```

Această funcție deschide conexiunea cu baza de date și apelează procedura stocată din baza de date care se ocupă cu inserarea de produse. Funcția primește ca parametru o listă de dbObjects care este o colecție de informații despre proprietățile unui produs. Se încearcă salvarea produsului, dacă operația se efectuează cu succes atunci se returnează id-ul înregistrării din baza de date, dacă nu, atunci se aruncă o excepție cu un mesaj corespunzător. Procedura stocată ce adaugă produse în baza de date este următoarea:

```
CREATE Procedure [dbo].[sp_Produse_Insert]  
    @Nume nvarchar(50),  
    @StocMinim decimal(18,2),  
    @Pret decimal(18,2),  
    @Um nvarchar(50)  
  
As  
Begin  
    Insert Into Produse  
        ([Nume],[StocMinim],[Pret],[Um])  
    Values  
        (@Nume,@StocMinim,@Pret,@Um)  
  
    Declare @ReferenceID int  
    Select @ReferenceID = @@IDENTITY  
  
    Return @ReferenceID  
  
End
```

Capitolul 4 - Prezentarea aplicației

Adăugarea unei bănci se face deschizând din tree *Firmele* după care deschidem *Listă Bănci* ce se expandează din *Firme*, ca în **Figura 4.6**. Se deschide tabul cu lista băncilor de unde putem adăuga o nouă bancă sau să modificăm o bancă existentă.



Figura 4.6. Deschiderea listei cu bănci

Pentru adăugarea unei bănci trebuie să completăm informațiile necesare despre o bancă: nume, adresă, CUI (cod unic de identificare), RC (registru comerț) și capitalul social într-o formă ilustrată în **Figura 4.7**.

Adauga banca

Nume

Adresa

CUI RC

Capital social

Iesire Salvare

Figura 4.7. Completarea datelor pentru salvarea unei bănci

La salvarea unei bănci se apelează aceeași funcție Create, dar în acest caz lista dbObjects va conține informații despre proprietățile băncii. Iar procedura stocată care se ocupa cu salvarea entității în baza de date este:

```
CREATE Procedure [dbo].[sp_Banci_Insert]
    @Nume varchar(50),
    @Adresa varchar(50),
    @CUI varchar(50),
```

Capitolul 4 - Prezentarea aplicației

@RC varchar(50),

@CapitalSocial decimal(18,2)

As

Begin

Insert Into Banci

([Nume],[Adresa],[CUI],[RC],[CapitalSocial])

Values

(@Nume,@Adresa,@CUI,@RC,@CapitalSocial)

Declare @ReferenceID int

Select @ReferenceID = @@IDENTITY

Return @ReferenceID

End

Adăugarea unui delegat presupune deschiderea listei de delegați din **Figura 4.8** și apelarea formei de adăugare a unui delegat.

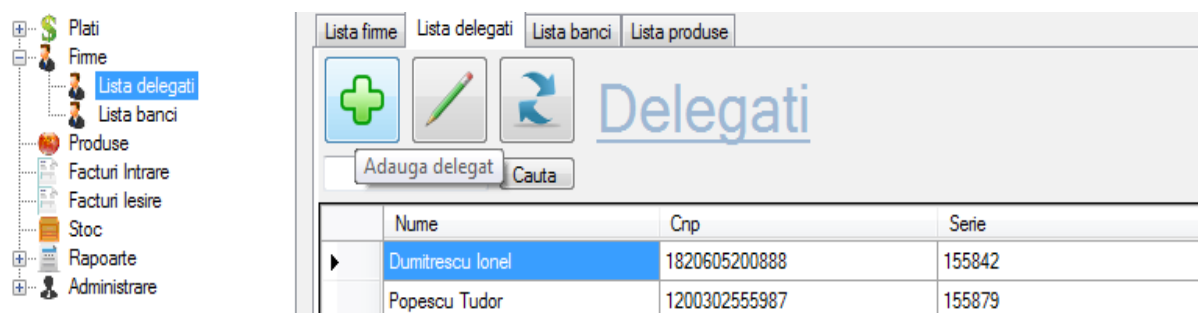


Figura 4.8. Deschiderea listei cu delegați

Butonul cu plus deschide o formă similară cu cea pentru produs și bancă ca în **Figura 4.9**. Datele necesare pentru adăugarea unui delegat fiind: numele, CNP (codul numeric personal) și seria.

Capitolul 4 - Prezentarea aplicației

Figura 4.9. Completarea datelor pentru adăugarea unui delegat

Salvarea delegatului se face prin apelul funcției Create, care la rândul ei apelează procedura stocată de inserare în tabela delegaților:

```
CREATE Procedure [dbo].[sp_Delegati_Insert]
```

```
    @Nume varchar(50),
```

```
    @CNP varchar(50),
```

```
    @Serie varchar(50)
```

```
As
```

```
Begin
```

```
    Insert Into Delegati
```

```
        ([Nume],[CNP],[Serie])
```

```
    Values
```

```
        (@Nume,@CNP,@Serie)
```

```
    Declare @ReferenceID int
```

```
    Select @ReferenceID = @@IDENTITY
```

```
    Return @ReferenceID
```

```
End
```

Firma este o relație ce are două chei străine: delegatul și banca. Pentru a putea adăuga

Capitolul 4 - Prezentarea aplicației

o firmă trebuie mai întâi să adăugăm cel puțin o bancă și un delegat. Din tree se deschide lista de firme (**Figura 4.10**) și se selectează adăugarea unei firme.

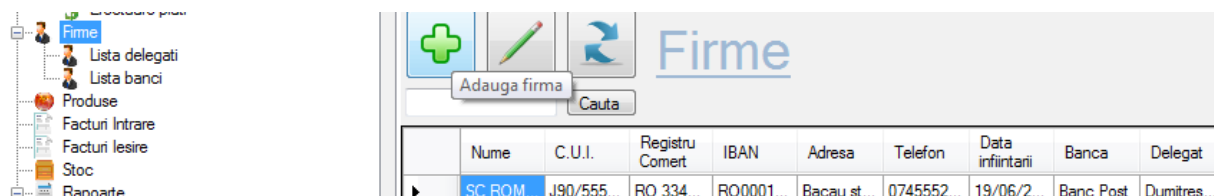


Figura 4.10. Deschiderea listei cu firme

Pentru adăugarea unei firme trebuie să completăm numele, CUI, IBAN, RC, adresa firmei, numărul de telefon și data înființării. Delegații și băncile sunt încărcate în controale de tip ComboBox, astfel se poate selecta fiecare, dar firma nu va reține decât id-ul lor (**Figura 4.11**).

Figura 4.11. Completarea datelor pentru adăugarea unei firme

La salvarea unei firme se apelează aceeași funcție Create, dar în acest caz lista dbObjects va conține informațiile completate despre firmă. Salvarea unei firme este posibilă doar dacă se selectează un delegat și o bancă, acestea fiind chei străine în tabela firmelor. Procedura stocată nu poate să salveze o firmă dacă aceasta nu are asociată niciun delegat sau nicio bancă. Procedura de inserare a unei firme conține următoarea secvență de

Capitolul 4 - Prezentarea aplicației

cod SQL:

```
CREATE Procedure [dbo].[sp_Firme_Insert]
    @Nume nvarchar(50),
    @CUI nvarchar(50),
    @RC nvarchar(50),
    @IBAN nvarchar(50),
    @Adresa nvarchar(50),
    @Telefon nvarchar(50),
    @DataInfiintarii datetime,
    @IdBanca int,
    @IdDelegat int
As
Begin
    Insert Into Firme
        ([Nume],[CUI],[RC],[IBAN],[Adresa],[Telefon],
        [DataInfiintarii],[IdBanca],[IdDelegat])
    Values
        (@Nume,@CUI,@RC,@IBAN,@Adresa,@Telefon,
        @DataInfiintarii,@IdBanca,@IdDelegat)

    Declare @ReferenceID int
    Select @ReferenceID = @@IDENTITY

    Return @ReferenceID
End
```