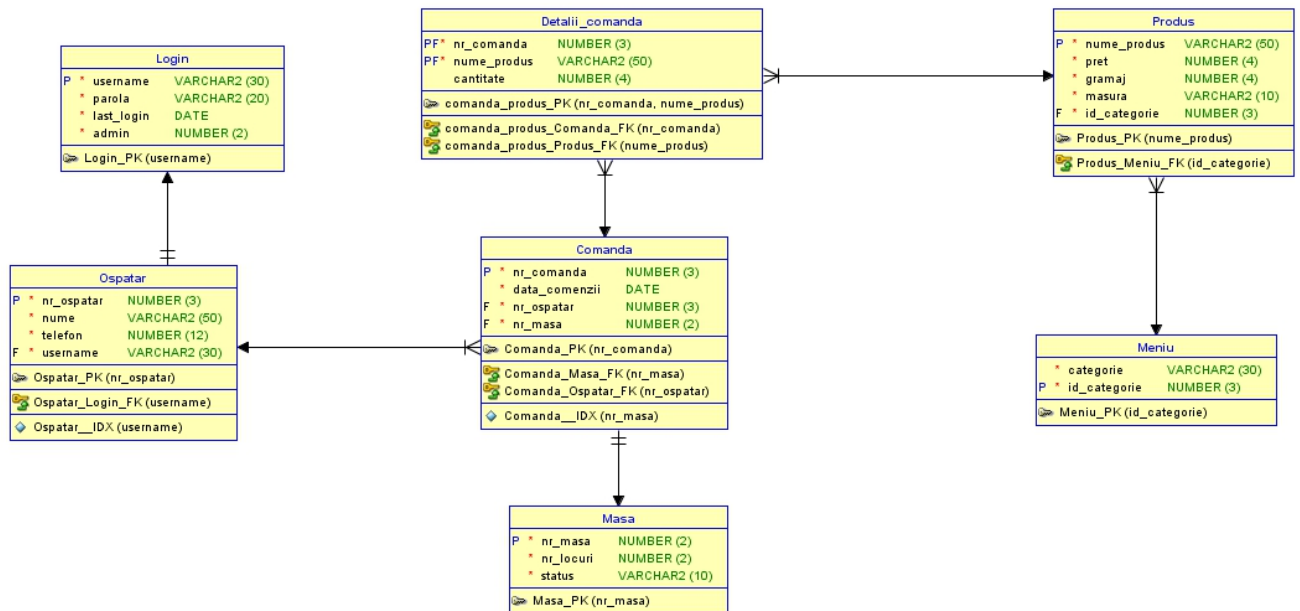


Anexă

1. Baza de date

1.1 Structura BD



1.2 Interogări

Cele mai relevante interogări folosite în cadrul interacțiunii cu BD sunt:

```
cmd.CommandText = "SELECT * FROM Login where username='" + username + "'";
OracleDataReader odr = cmd.ExecuteReader();
```

Selectarea utilizatorilor din tabela Login in functie de username

```
cmd.CommandText = "INSERT INTO Comanda(NR_COMANDA,DATA_COMENZII,NR_OSPATAR,NR_MASA) VALUES('" + comanda.NumarComanda + "','" + comanda.DataComenzii.Day + "/" + comanda.DataComenzii.Month + "/" + comanda.DataComenzii.Year + "','" + comanda.NumarOspatar + "','" + comanda.NumarMasa + "')";
Console.WriteLine(cmd.CommandText);
cmd.ExecuteNonQuery();

if (comanda.DetaliiComanda.Count > 0)
{
    foreach (string produs in comanda.DetaliiComanda.Keys)
    {
        cmd.CommandText = "INSERT INTO DETALII_COMANDA(NR_COMANDA,NUME_PRODUS,CANTITATE) VALUES('" + comanda.NumarComanda + "','" + produs + "','" + comanda.DetaliiComanda[produs] + "')";
        Console.WriteLine(cmd.CommandText);
        cmd.ExecuteNonQuery();
    }
}
```

Inserarea unei noi comenzi

```
cmd.CommandText = "UPDATE produs SET PRET=" + produs.PretProdus + ", GRAMAJ=" + produs.Gramaj + ", MASURA=";

switch (produs.MasuraProdus)
{
    case Masura.Bucata:
        cmd.CommandText += "'bucata'";
        break;
    case Masura.Gram:
        cmd.CommandText += "'gram'";
        break;
    case Masura.Kilogram:
        cmd.CommandText += "'kilogram'";
        break;
    case Masura.Litru:
        cmd.CommandText += "'litru'";
        break;
}

cmd.CommandText += ", ID_CATEGORIE=" + produs.IdCategorie + " WHERE NUME_PRODUS='" + produs.NumeProdus + "'";
cmd.ExecuteNonQuery();
```

Actualizarea unui produs în funcție de id-ul categoriei din care face parte

```
string usernameOspatar = "";
cmd.CommandText = "SELECT USERNAME FROM OSPATAR WHERE NR_OSPATAR=" + numar;
OracleDataReader odr = cmd.ExecuteReader();
while (odr.Read())
{
    usernameOspatar = Convert.ToString(odr["USERNAME"]);
}

// ștergerea comenzilor asociate(dacă există)
cmd.CommandText = "DELETE FROM DETALII_COMANDA WHERE NR_COMANDA=(SELECT NR_COMANDA FROM COMANDA WHERE NR_OSPATAR=" + numar + ")";
cmd.ExecuteNonQuery();

cmd.CommandText = "DELETE FROM COMANDA WHERE NR_OSPATAR=" + numar;
cmd.ExecuteNonQuery();

cmd.CommandText = "DELETE FROM OSPATAR WHERE NR_OSPATAR=" + numar;
cmd.ExecuteNonQuery();

cmd.CommandText = "DELETE FROM LOGIN WHERE USERNAME='" + usernameOspatar + "'";
cmd.ExecuteNonQuery();
```

Ștergerea unui utilizator în funcție de numărul lui de identificare

2. Bucăți de cod semnificative

```
1 reference
private RestaurantDAO()
{
    //Deschidere conexiune
    try
    {
        this._oracleConnection = new OracleConnection(_oradb);
        this._oracleConnection.Open();
    }
    catch(Exception e)
    {
        Console.WriteLine("Eroare la conexiune: " + e.Message);
    }
}
```

Deschiderea conexiunii cu baza de date

```

32 references | 0/1 passing
public static RestaurantDAO GetInstance()
{
    if (_instance == null)
    {
        _instance = new RestaurantDAO();
    }

    return _instance;
}

```

Ilustrarea șablonului Singleton

```

double total = 0;
foreach (string numeProduce in c.DetaliiComanda.Keys)
{
    string[] row = new string[4];
    row[0] = numeProduce;

    Produe p = _userController.GetProduce(numeProduce);
    row[1] = p.PretProduce.ToString();
    row[2] = c.DetaliiComanda[numeProduce].ToString();
    row[3] = (p.PretProduce * c.DetaliiComanda[numeProduce]).ToString();
    total += p.PretProduce * c.DetaliiComanda[numeProduce];
    dataGridViewProduce.Rows.Add(row);
}

textBoxOrderNumber.Text = c.NumarComanda.ToString();
_comanda = c;
textBoxOrderTable.Text = c.NumarMasa.ToString();
labelTotal.Text = total.ToString();

```

Dacă există deja o comandă la masa selectată, sunt preluate din baza de date detaliile referitoare la comanda în curs

```

_numeCategorie = comboBoxCategory.SelectedItem.ToString();
_idCategorie = _userController.GetIdCategorie(_numeCategorie);

_produce = new List<Produe>();

var produse = _userController.GetProduceByCategorie(_idCategorie);

listBoxItems.Items.Clear();

foreach(Produe p in produse)
{
    _produce.Add(p);
    listBoxItems.Items.Add(p.NumeProduce);
}

```

Funcția care permite căutarea produselor în funcție de o categorie selectată

```

2 references
public void RefreshControl()
{
    _produse = _adminController.GetProdus();
    _menu = _adminController.GetMenu();

    dataGridViewUpdateItem.Rows.Clear();

    foreach (Produs p in _produse)
    {
        string[] row = new string[5];
        row[0] = SearchCategoryName(p, _menu);
        row[1] = p.NumeProdus;
        row[2] = p.PretProdus.ToString();
        row[3] = p.MasuraProdus.ToString();
        row[4] = p.Gramaj.ToString();
        dataGridViewUpdateItem.Rows.Add(row);
    }
}

/// <summary>

```

După prelucrările asupra produselor, funcția este apelată pentru a se afișa datele actualizate în interfață