UNIVERSITATEA TEHNICă "GHEORGHE ASACHI", IAşI

FACULTATEA DE AUTOMATICă şI CALCULATOARE

Proiect la Baze de Date

"Rezervarea locurilor la o pizzerie"

Nume și prenume: Socea Gabriel

Grupa: 1310B

Materie: Baze de date

Profesor coordonator: Avram Sorin

Descrierea aplicației

Proiectul „Rezervarea locurilor la Pizzeria Mille Gusti” urmărește gestionarea datelor unei pizzerii cu scopul de a face rezervare la o masă. Tabelele folosite în această aplicație sunt: Rezervări, Rezervări_info, Clienți, Personal, Mese.

Aplicația desktop realizată dorește o eficientizare a posibilității de a rezerva un loc, bazându-se pe utilizarea unei interfețe cu utilizatorul prin intermediul căreia se realizează comunicarea cu o bază de date care conține toate informațiile aferente acestei acțiuni. Utilizatorul (un membru din echipa pizzeriei) are posibilitatea să realizeze operații diverse, cum ar fi:

1. Inserarea, update-ul, ștergerea, precum și vizualizarea elementelor din tabela Rezervări
2. Inserarea, update-ul, ștergerea, precum și vizualizarea elementelor din tabela Rezervări_info
3. Inserarea, update-ul, ștergerea, precum și vizualizarea elementelor din tabela Clienți
4. Inserarea, update-ul, ștergerea, precum și vizualizarea elementelor din tabela Mese
5. Inserarea, update-ul, ștergerea, precum și vizualizarea elementelor din tabela Personal

Tabela Rezervări conține informațiile aferente unei rezervări a unei mese la pizzerie, tabela Rezervări_info este o extensie a tabelei Rezervări si conține date suplimentare pentru aceasta tabelă, tabela Mese conține informații despre fiecare masă din pizzerie precum număr de locuri și disponibilitate, tabela Personal conține informații despre echipa pizzeriei adică pizzeri, manageri, ospătari, etc. iar tabela Clienți conține informații despre fiecare client care a venit cel puțin odată la pizzerie și a făcut o rezervare.

Aplicația dispune de o interfață de login care permite utilizarea aplicației doar a utilizatorilor care au creat un cont pe aceasta aplicație.

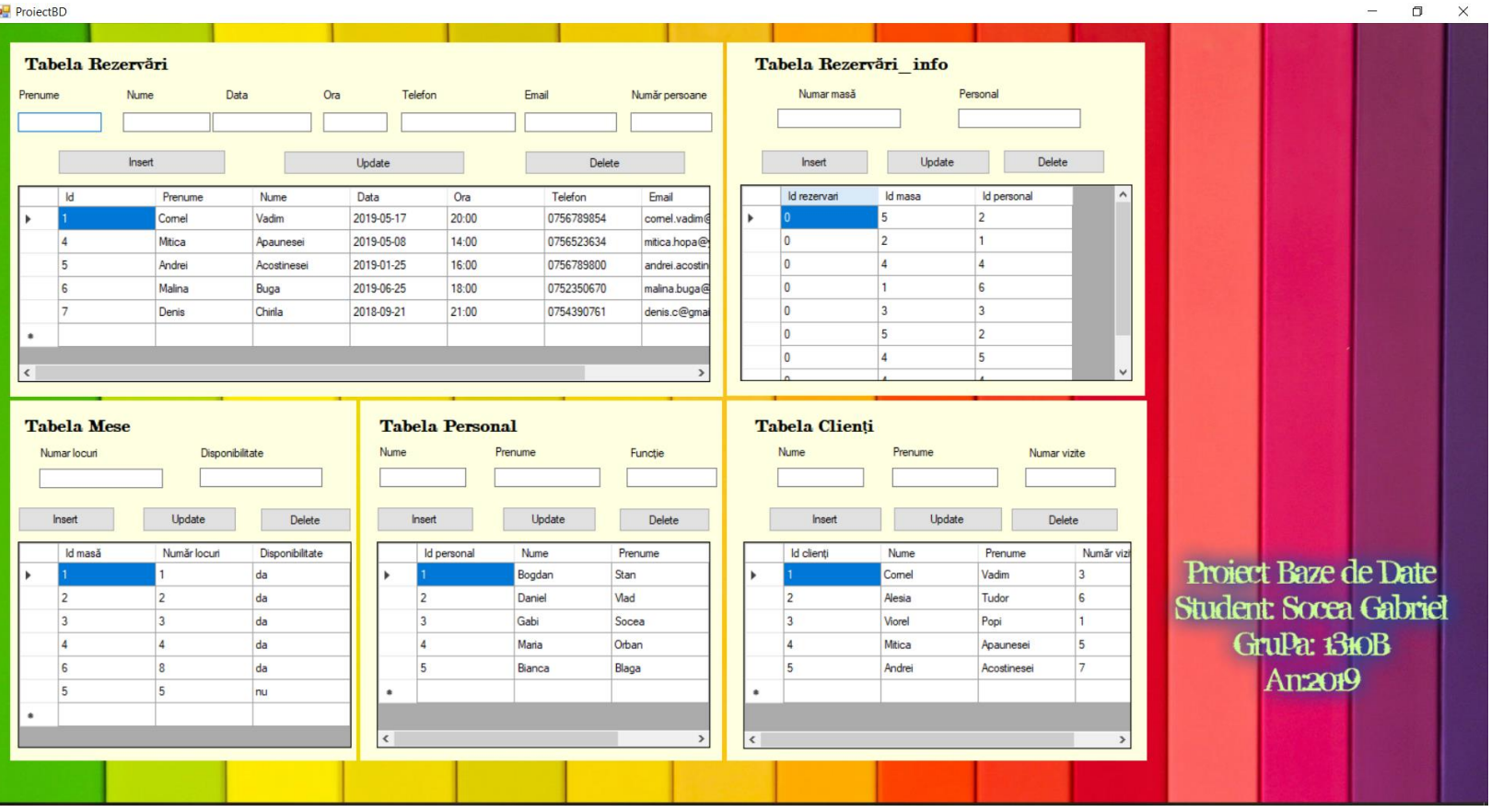
Dacă utilizatorul introduce datele de autentificare corecte, atunci se deschide interfața grafică in care utilizatorul poate efectua insert, update, delete și vizualizarea datelor pentru fiecare tabela in parte.

Dacă utilizatorul introduce datele de autentificare eronate, sau cel puțin un câmp din datele de autentificare este greșit atunci se va afișa un mesaj de eroare si nu se va deschide interfața cu utilizatorul pentru manipularea datelor din tabele.

La rularea aplicației, utilizatorul beneficiază de operațiile de mai sus prin intermediul interfețelor:

1. Interfața de Login

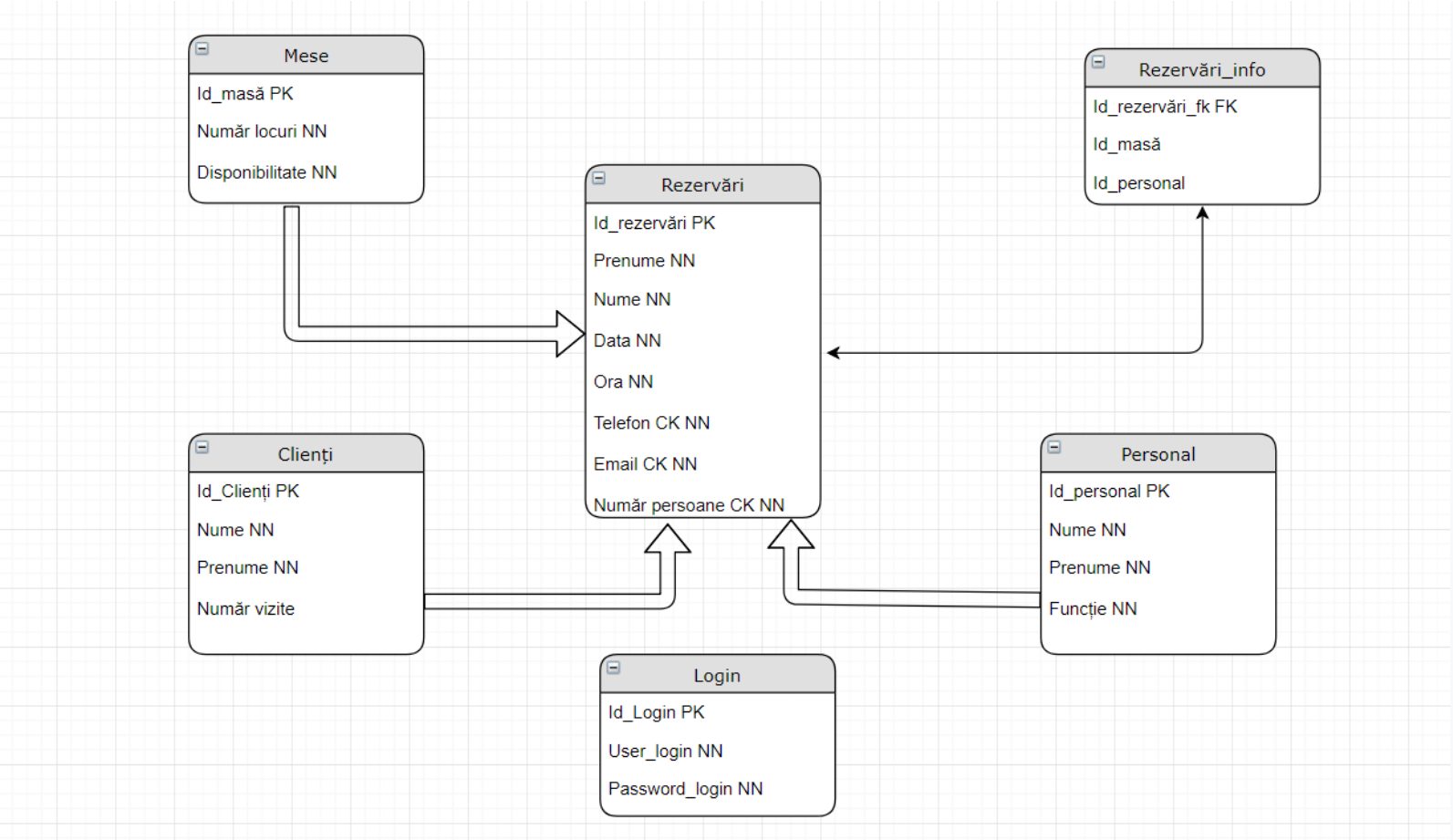
2. Interfața grafică a tabelelor



Tehnologii folosite

Aplicația desktop a fost dezvoltată utilizând C# Windows Forms Application, iar ca sistem de baze de date relaționale a fost folosit PostgreSQL. Am încercat să folosesc baza de date Oracle, dar nu am reușit să fac o conexiune și din acest motiv am ales să folosesc PostgreSQL.

Diagrama ER a bazei de date



Relațiile dintre tabele

Relații **One to Many:**

1. Mese -> Rezervari: o masa poate avea mai multe rezervari, dar nu toate in acelasi timp
2. Clienți -> Rezervari: un client poate face mai multe rezervari
3. Personal -> Rezervari: un mebru din echipa pizzeriei poate face mai multe rezervari

Relații **One to One:**

1. Rezervări <-> Rezervări_info: fiecărei înregistrări din tabela rezervări ii corespunde o înregistrare din tabela rezervări_info

Constrângerile folosite

În acest proiect am folosit constrangeri de tip **NOT NULL** la aproape toate coloanele, pentru a evita ca o tabela sa aiba valori nule.

Am folosit constrangerea **UNIQUE** la tabela Rezervari la coloanele telefon, email.

Am folosit constrangerea **CHECK** la tabela Rezervari la coloana NR_PERSOANE.

Am folosit o singura constrangere de tip **FOREIGN KEY** la tabela Rezervari_info la coloana Id_rezervari care preia datele din coloana Id_rezervari de la tabela Rezervari.

Descrierea modalității de conectare la baza de date din aplicație

În cadrul proiectului există proiectul App.Service în care se regăsește clasa DBConnection care face conexiunea cu baza de date (folosește Npgsql v 4.0.7). Aceasta conține datele: nume server, nume baza de date, port, user și parola care realizează conexiunea propriu-zisă. În același timp, există metodele Open și Close, (înainte de orice query din db se apelează Open, după terminare se apelează Close).

Cod de realizare a unei conexiuni cu baza de date:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Npgsql;

namespace App.Controller
{
    public class DBConnection
    {
        private string _server = "localhost";
        private string _userName = "postgres";
        private string _password = "admin";
        private string _database = "bd_project";
        private string _port = "5432";

        private NpgsqlConnection _npgSqlConnnection;

        public DBConnection()
        {
```

```
            string connectionString =
$"Username={_userName};Password={_password};Host={_server};Port={_port};Database={_databa
se};";

            _npgSqlConnnection = new NpgsqlConnection
            {
                ConnectionString = connectionString
            };
        }

        public NpgsqlConnection NpgsqlConnection
        {
            get { return _npgSqlConnnection; }
        }

        public void OpenConnection()
        {
            if (_npgSqlConnnection.State == System.Data.ConnectionState.Closed)
            {
                _npgSqlConnnection.Open();
            }
        }

        public void CloseConnection()
        {
            if (_npgSqlConnnection.State == System.Data.ConnectionState.Open)
            {
                _npgSqlConnnection.Close();
            }
        }
    }
}
```