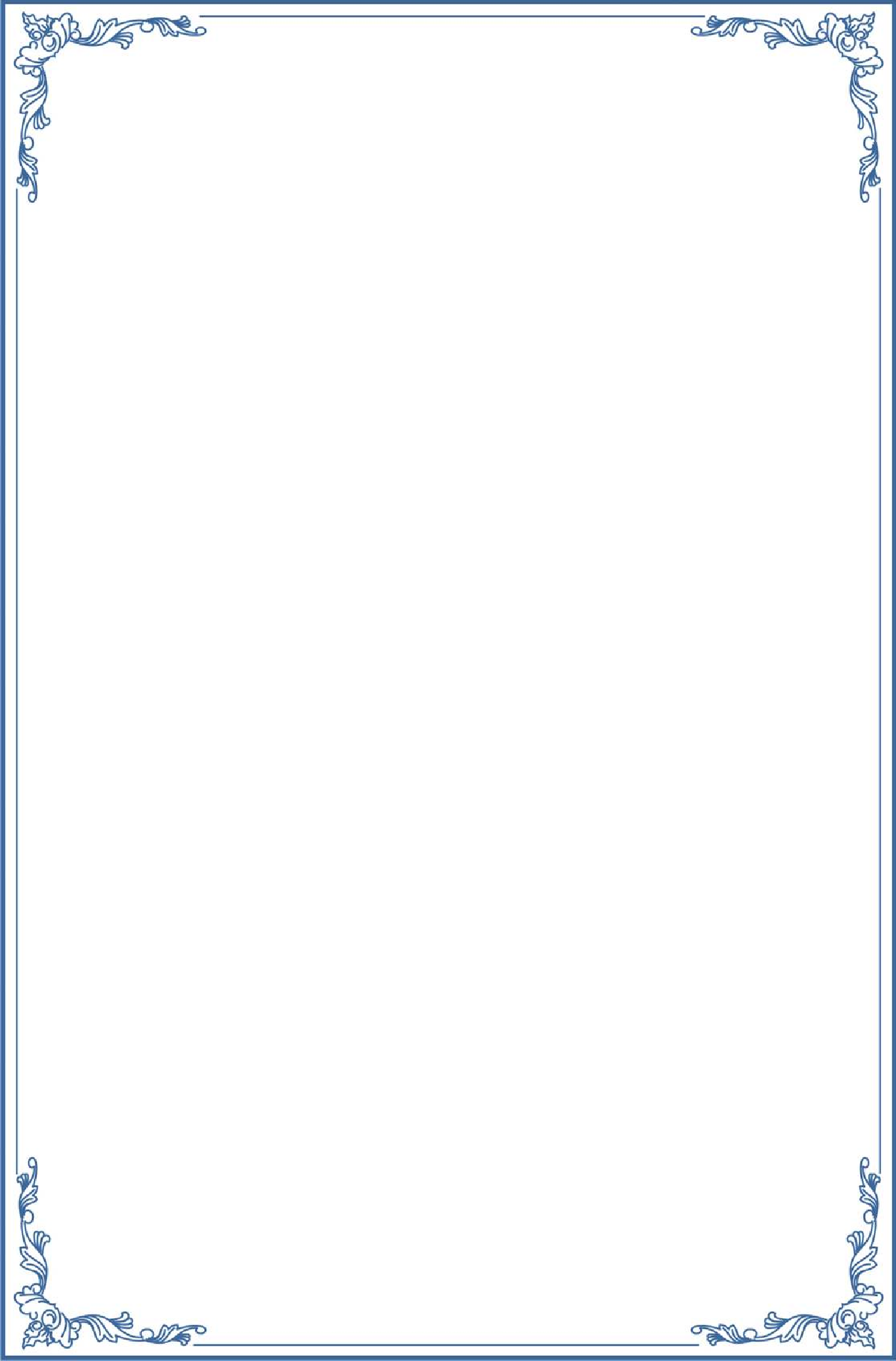
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**



**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO CUỐI KỲ**

**ĐỀ TÀI: ỨNG DỤNG QUẢN LÍ PHÒNG TRỌ**

**Lớp Lập trình trên Windows\_WIPR230579\_22\_2\_09**

**GVHD: T.S Lê Văn Vinh**

**SVTH: Lê Hữu Huy 21110471**

**Đào Tuấn Kiệt 21110518**

**Đào Quang Duy 21110398**

**Tp. Hồ Chí Minh, ngày 09 tháng 06 năm 2023**

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

Họ và tên Sinh viên 1: Lê Hữu Huy

Họ và tên Sinh viên 2 : Đào Tuấn Kiệt

Họ và tên Sinh viên 3 : Đào Quang Duy

Ngành:Công nghệ Thông tin

Tên đề tài: **Ứng dụng quản lí phòng trọ**

Họ và tên Giảng viên hướng dẫn: T.S Lê Văn Vinh

**NHẬN XÉT**

**1. Về nội dung đề tài & khối lượng thực hiện:**

**.......................................................................................................................................**

**..............................................................................................................................................................................................................................................................................**

**2. Ưu điểm:**

**.......................................................................................................................................**

**.......................................................................................................................................**

**.......................................................................................................................................**

**3. Khuyết điểm**

**.......................................................................................................................................**

**.......................................................................................................................................**

**.......................................................................................................................................**

**4. Điểm :**

Tp.Hồ Chí Minh, ngày 09 tháng 6 năm 2023

Giáo viên hướng dẫn

(Ký & ghi rõ họ tên)

**Mục lục**

[LỜI CẢM ƠN 4](#_Toc17925)

[PHẦN 1:GIỚI THIỆU 5](#_Toc6413)

[1. Mục đích chính của ứng dụng 5](#_Toc4587)

[2. Các tính năng có trong ứng dụng 5](#_Toc15911)

[3. Cấu trúc 3 tầng 6](#_Toc28385)

[PHẦN 2: NỘI DUNG 9](#_Toc3733)

[1. Cơ sở dữ liệu 9](#_Toc25406)

[1.1 Sơ đồ cơ sở dữ liệu 9](#_Toc11683)

[1.2 Các đối tượng trong cơ sở dữ liệu 9](#_Toc2628)

[2. Tầng giao diện 10](#_Toc16934)

[3. Tầng nghiệp vụ và tầng giao tiếp với cơ sở dữ liệu 10](#_Toc28065)

[3.1 ADO.NET 10](#_Toc31864)

[3.2 ENTITY FRAMEWORK 10](#_Toc15374)

[3.3 LINQ 10](#_Toc6445)

# LỜI CẢM ƠN

Đầu tiên, Nhóm chúng em xin gửi lời cảm ơn chân thành đến Thầy Lê Văn Vinh, giảng viên hướng dẫn của nhóm trong quá trình thực hiện đồ án cuối kì môn Lập trình trên Windows.

Trong suốt quá trình học tập, Thầy đã cung cấp cho nhóm chúng em rất nhiều kiến thức bổ ích và hỗ trợ nhóm trong việc nghiên cứu và phân tích dữ liệu. Thầy luôn tận tình giải đáp các thắc mắc và giúp đỡ chúng em để hoàn thành đồ án tốt nhất.

Thầy đã truyền đạt rất nhiều kỹ năng và kinh nghiệm quý giá trong lĩnh vực này, đó là điều nhóm chúng em sẽ luôn trân trọng và khâm phục.

Bộ môn Lập trình trên Windows là một môn học thú vị và vô cùng bổ ích. Tuy nhiên, những kiến thức và kỹ năng về môn học này của nhóm chúng em vẫn còn nhiều hạn chế. Do đó, đồ án cuối kỳ của nhóm khó tránh khỏi những sai sót. Kính mong Thầy xem xét và góp ý giúp bài tiểu luận của nhóm chúng em được hoàn thiện hơn.

Một lần nữa, nhóm chúng em xin chân thành cảm ơn Thầy Lê Văn Vinh và hy vọng có cơ hội được học hỏi thêm từ Thầy trong tương lai.

Trân trọng.

PHẦN 1:GIỚI THIỆU

## Mục đích chính của ứng dụng

Ứng dụng cho phép người dùng có thể thuê phòng trọ thích hợp đối với người dùng là người thuê trọ và quản lí tốt được các phòng trọ và người thuê trọ của mình đối với người dùng là chủ trọ.

## Các tính năng có trong ứng dụng

- Chức năng chung:

+ Đăng nhập/Tạo tài khoản

+ Đổi thông tin cá nhân

+ Xem thông tin phòng

+ Xem thông tin hợp đồng

- Đối vời người dùng là chủ trọ:

+ Thêm phòng mới

+ Xóa phòng

+ Thêm người thuê vào phòng trọ bất kì

+ Cho người thuê ra khỏi phòng trọ

+ Xem thống kê các phòng trọ chủ trọ sở hữu

+ Ghi hóa đơn (điện, nước, rác,...)

+ Cập nhật thông tin/tình trạng phòng

+ Tìm kiếm người thuê dựa trên mã người thuê trọ

- Đối với người dùng là người thuê:

+ Tìm phòng

+ Thuê phòng

+ Trả phòng

+ Thanh toán tiền thuê

+ Thanh hóa hóa đơn (điện, nước, rác,...)

+ Lấy tiền cọc

+ Đánh giá phòng

## Cấu trúc 3 tầng

Kiến trúc 3 tầng là một kiến trúc kiểu client/server mà trong đó giao diện người dùng (UI-user interface), các quy tắc xử lý (BR-business rule hay BL-business logic), và việc lưu trữ dữ liệu được phát triển như những module độc lập, hầu hết là được duy trì trên các nền tảng độc lập, và mô hình 3 tầng được coi là một kiến trúc phần mềm và là một mẫu thiết kế.

Kiến trúc 3 tầng gồm có 3 phần chính:

-Presentation Layer (GUI) : Tầng giao diện, tầng này có nhiệm vụ chính giao tiếp với người dùng. Nó gồm các thành phần giao diện (win form, web form,…) và thực hiện các công việc như nhập liệu, hiển thị dữ liêu, kiểm tra tính đúng đắn dữ liệu trước khi gọi tầng Business Layer (BL).

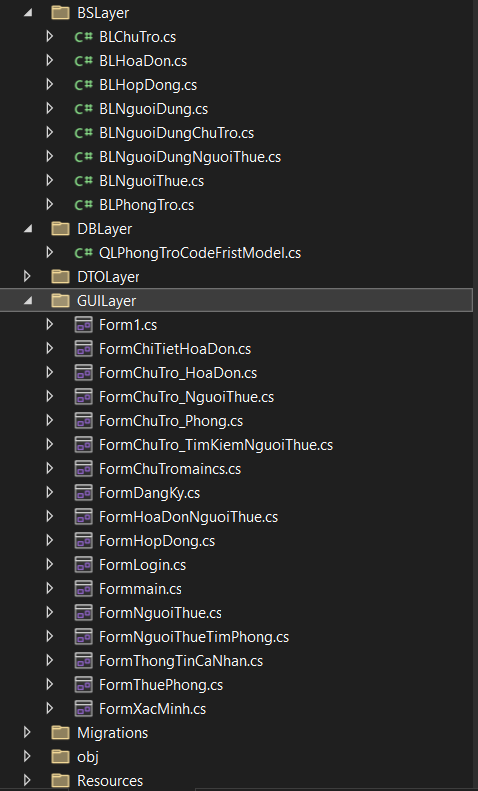
-Business Layer (BL) : Tầng nghiệp vụ, tầng này phân ra 2 thành nhiệm vụ :

+Đây là nơi đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Layer và lưu xuống hệ quản trị CSDL.

+Đây còn là nơi kiểm tra các ràng buộc, tính toàn vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer.

-Data Layer (DL) : Tầng giao tiếp với CSDL, tầng này có chức năng thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa,…).

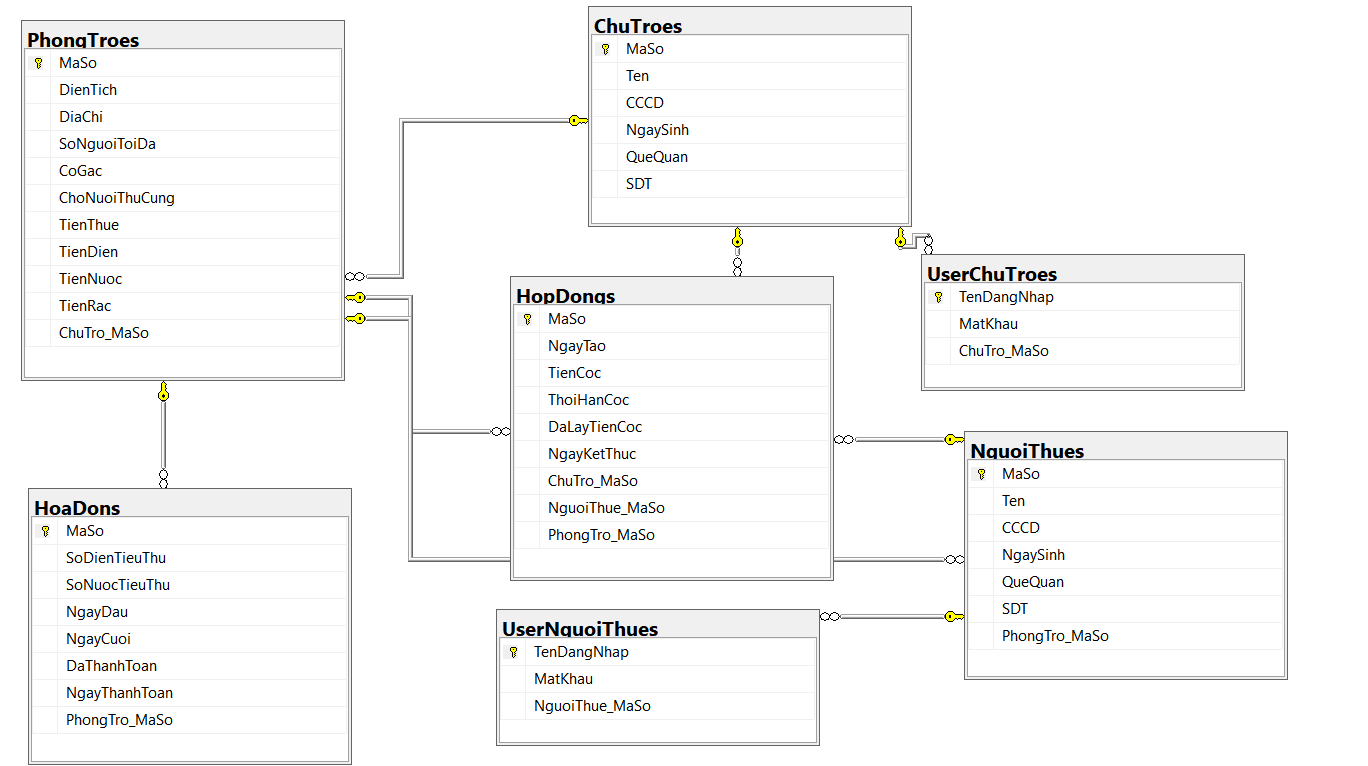
Cấu trúc các file code của ứng dụng theo mô hình 3 tầng:



# PHẦN 2: NỘI DUNG

## Cơ sở dữ liệu

**1.1 Sơ đồ cơ sở dữ liệu**



### **1.2 Các đối tượng trong cơ sở dữ liệu**

**- PhongTro:** đối tượng phòng trọ

- **NguoiThue**: đối tượng người thuê trọ

- **ChuTro**: đối tượng chủ trọ

- **HoaDon**: đối tượng hóa đơn

-**HopDong**: đối tượng hợp đồng

-**UserNguoiThue**: đối tượng người dùng ứng dụng là người thuê

-**UserChuTro**: đối tượng người dùng ứng dụng là chủ trọ

## Tầng giao diện

Tầng giao diện có tất cả 12 form và gồm 3 form chính: Login, Người dùng chủ trọ, Người dùng người thuê. Các Form này sẽ dẫn đến các form khác.

1. Form login: Gồm chức năng đăng nhập, quên mật khẩu và đăng kí người dùng.

A screenshot of a computer

Description automatically generated with medium confidence

1. Form chủ trọ: Gồm các chức năng của người dùng và một ReportVeiwer để xem thống kê về hóa đơn, phòng trọ và hợp đồng của mình.

A screenshot of a computer

Description automatically generated

1. Form người thuê: Tìm nhà trọ, xem hợp đồng, xem hóa đơn và thông tin cá nhân

A screenshot of a computer

Description automatically generated with medium confidence

1. Một vài Form khác.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## Tầng nghiệp vụ và tầng giao tiếp với cơ sở dữ liệu

### **ENTITY FRAMEWORK**

Viết ứng dụng sử dụng Entity Framework theo kịch bản Code first.

1. Tạo cơ sở dữ liệu:

Code tạo cơ sở dữ liệu:

A picture containing text, screenshot, software

Description automatically generated

Ví dụ với CSDL có table Nguoi: Dưới đây là class Người

A screen shot of a computer

Description automatically generated with medium confidence

Tương tự với class người ta sẽ có các class sau được lưu trong DTOLayer:

A screenshot of a computer program

Description automatically generated with medium confidence

1. Tâng nghiệp vụ:

Giao tiếp với là tầng trung gian giữa GUILayer và DBLayer

Dưới đây code là BLNguoiDungChuTro:

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.Entity;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLyPhongTro.BSLayer

{

public class BLNguoiDungChuTro: BLNguoiDung

{

public BLNguoiDungChuTro() : base()

{

db = QLPhongTroCodeFristModel.Instance;

}

public override DataTable LayNguoiDung()

{

var query = from userChuTro in db.NguoiDungChuTroes

select userChuTro;

return query.ToDataTable();

}

public override NguoiDung DangNhap(string mk, string tenDN)

{

var query = (from userChuTro in db.NguoiDungChuTroes

where tenDN == userChuTro.TenDangNhap &&

mk == userChuTro.MatKhau

select userChuTro).FirstOrDefault();

return query;

}

public override bool DangKi(string hVTen, string cCCD, string sDT, string qQuan, string tenDn, string mK, DateTime nSinh)

{

string maSo = (Convert.ToInt16(db.NguoiDungChuTroes.Max(x => x.ChuTro.MaSo) +1)).ToString();

ChuTro newChuTro = new ChuTro(maSo, hVTen, cCCD, nSinh, qQuan, sDT, null);

NguoiDungChuTro newUser = new NguoiDungChuTro(tenDn, mK, newChuTro);

db.NguoiDungChuTroes.Add(newUser);

db.SaveChanges();

return false;

}

public override NguoiDung CheckTrungTenDangNhap(string tenDN)

{

var query = (from userChuTro in db.NguoiDungChuTroes

where tenDN == userChuTro.TenDangNhap

select userChuTro).FirstOrDefault();

return query;

}

public NguoiDungChuTro TimTheoMaSo(string maSo)

{

var query = (from user in db.NguoiDungChuTroes

where maSo == user.ChuTro.MaSo

select user).FirstOrDefault();

return query;

}

public override bool DoiMK(string CCCD, string TenDn, string mK)

{

var query = (from userChuTro in db.NguoiDungChuTroes

where TenDn == userChuTro.TenDangNhap && CCCD == userChuTro.ChuTro.CCCD

select userChuTro).FirstOrDefault();

if (query != null)

{

query.MatKhau = mK;

db.SaveChanges();

return true;

}

return false;

}

public override bool CapNhatThongTin(string id, string hVTen, string cCCD, string sDT, string qQuan, string tenDn, string mK, DateTime nSinh)

{

var nguoi = (from nguoithue in db.NguoiDungNguoiThues

where nguoithue.NguoiThue.MaSo == id

select nguoithue).FirstOrDefault();

if (nguoi != null)

{

nguoi.TenDangNhap = tenDn;

nguoi.MatKhau = mK;

nguoi.NguoiThue.CCCD = cCCD;

nguoi.NguoiThue.NgaySinh = nSinh;

nguoi.NguoiThue.QueQuan = qQuan;

nguoi.NguoiThue.SDT = sDT;

nguoi.NguoiThue.Ten = hVTen;

db.SaveChanges();

return true;

}

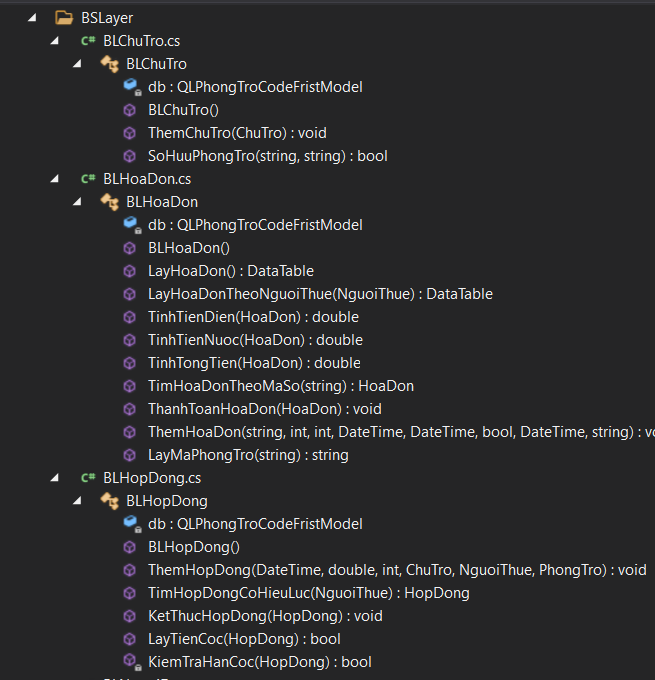
return false;

}

}

}

Và dưới đây là toàn bộ các hàm của các bộ phận trong tầng nghiệp vụ:



A screenshot of a computer program

Description automatically generated with medium confidence

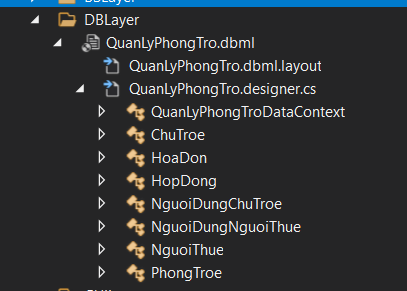
A screen shot of a computer program

Description automatically generated with low confidence

### **LinQ**

1. DBLayer

Dựa trên CSDL đã tạo ở project sử dụng entity framework ta tạo file QuanLyPhongTro.dbml như sau:



A picture containing text, screenshot, diagram, rectangle

Description automatically generated

1. BSLayer

Ở tầng nghiệp vụ có các chức năng tương tự như tầng nghiệp vụ ở entity framework và dưới đây là code của BLNguoiDungChoThue.

using QuanLyPhongTro.BSLayer;

using QuanLyPhongTro.DBLayer;

using System;

using System.Collections.Generic;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLyPhongTro.BSLayer

{

public class BLNguoiDungNguoiThue

{

QuanLyPhongTroDataContext db;

public BLNguoiDungNguoiThue()

{

db = QuanLyPhongTroDataContext.Instance;

}

public DataTable LayNguoiDung()

{

var query = from userNguoiThue in db.NguoiDungNguoiThues

select userNguoiThue;

return query.ToDataTable();

}

public NguoiDungNguoiThue DangNhap(string mk, string tenDN)

{

var query = (from userNgThue in db.NguoiDungNguoiThues

where tenDN == userNgThue.TenDangNhap &&

mk == userNgThue.MatKhau

select userNgThue).FirstOrDefault();

return query;

}

public bool DangKi(string hVTen, string cCCD, string sDT, string qQuan, string tenDn, string mK, DateTime nSinh)

{

string maSo = (Convert.ToInt16(db.NguoiDungNguoiThues.Max(x => x.NguoiThue.MaSo)) + 1).ToString("D4");

NguoiThue newNguoiThue = new NguoiThue

{

MaSo = maSo,

Ten = hVTen,

CCCD = cCCD,

NgaySinh = nSinh,

QueQuan = qQuan,

SDT = sDT,

};

NguoiDungNguoiThue newUser = new NguoiDungNguoiThue

{

TenDangNhap = tenDn,

MatKhau = mK,

NguoiThue = newNguoiThue

};

db.NguoiDungNguoiThues.InsertOnSubmit(newUser);

db.SubmitChanges();

return false;

}

public NguoiDungNguoiThue CheckTrungTenDangNhap(string tenDN)

{

var query = (from userNthue in db.NguoiDungNguoiThues

where tenDN == userNthue.TenDangNhap

select userNthue).FirstOrDefault();

return query;

}

public NguoiDungNguoiThue TimTheoMaSo(string maSo)

{

var query = (from user in db.NguoiDungNguoiThues

where maSo == user.NguoiThue.MaSo

select user).FirstOrDefault();

return query;

}

public bool DoiMK(string CCCD, string TenDn, string mK)

{

var query = (from userNguoiThue in db.NguoiDungNguoiThues

where TenDn == userNguoiThue.TenDangNhap && CCCD == userNguoiThue.NguoiThue.CCCD

select userNguoiThue).FirstOrDefault();

if (query != null)

{

query.MatKhau = mK;

db.SubmitChanges();

return true;

}

return false;

}

public bool CapNhatThongTin(string id, string hVTen, string cCCD, string sDT, string qQuan, string tenDn, string mK, DateTime nSinh)

{

var nguoi = (from nguoithue in db.NguoiDungNguoiThues

where nguoithue.NguoiThue.MaSo == id

select nguoithue).FirstOrDefault();

if (nguoi != null)

{

nguoi.TenDangNhap = tenDn;

nguoi.MatKhau = mK;

nguoi.NguoiThue.CCCD = cCCD;

nguoi.NguoiThue.NgaySinh = nSinh;

nguoi.NguoiThue.QueQuan = qQuan;

nguoi.NguoiThue.SDT = sDT;

nguoi.NguoiThue.Ten = hVTen;

db.SubmitChanges();

return true;

}

return false;

}

}

}

### **ADO.NET**

1. DBLayer

Sử dụng để kết nối trực tiếp cơ sở dữ liệu sử dụng các querystring(câu truy vấn dưới dạng string)

Dưới đây gồm 3 hàm:

+ ExecuteQueryDataSet: Truy vấn trả về 1 một bảng dưới dạng DataSet

+ MyExecuteNonQuery: Thực hiện câu truy vấn và không trả về một đối tượng được truy vấn

+ GetSingleRecord: Truy vấn trả về 1 record của 1 table trong database dưới dạng DataRow

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace QuanLyPhongTro.DBLayer

{

public class QuanLyPhongTroContext

{

string ConnStr = "Data Source=DESKTOP-S1O44U1\\SQLEXPRESS;Initial Catalog=QuanLyPhongTro.QLPhongTroCodeFristModel;Integrated Security=True";

SqlConnection conn = null; SqlCommand comm = null; SqlDataAdapter da = null;

public QuanLyPhongTroContext()

{

conn = new SqlConnection(ConnStr);

comm = conn.CreateCommand();

}

public DataSet ExecuteQueryDataSet(string strSQL, CommandType ct)

{

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

comm.CommandText = strSQL;

comm.CommandType = ct;

da = new SqlDataAdapter(comm);

DataSet ds = new DataSet();

da.Fill(ds);

return ds;

}

public bool MyExecuteNonQuery(string strSQL, CommandType ct)

{

bool f = false;

if (conn.State == ConnectionState.Open)

conn.Close();

conn.Open();

comm.CommandText = strSQL;

comm.CommandType = ct;

try

{

comm.ExecuteNonQuery();

f = true;

}

finally

{

conn.Close();

}

return f;

}

public DataRow GetSingleRecord(string strSQL, CommandType ct)

{

DataSet ds = ExecuteQueryDataSet(strSQL, ct);

if (ds != null && ds.Tables.Count > 0 && ds.Tables[0].Rows.Count > 0)

return ds.Tables[0].Rows[0];

else return null;

}

}

}

1. BSLayer

Về tầng nghiệp vụ thì cũng có các hàm và chức năng tương tự như 2 project trên nhưng khác là sử dụng các câu truy vấn dạng string để kết nối với CSDL.

using QuanLyPhongTro.DBLayer;

using System;

using System.Collections.Generic;

using System.Data;

using System.Text;

using System.Threading.Tasks;

namespace QuanLyPhongTro.BSLayer

{

public class BLHoaDon

{

QuanLyPhongTroContext db;

public BLHoaDon()

{

db = new QuanLyPhongTroContext();

}

public DataTable LayHoaDonTheoNguoiThue(NguoiThue ngThue)

{

string query = "select \* from HoaDons" +

"where PhongTro\_MaSo = " +

"'" + ngThue.PhongTro.MaSo + "'";

return db.ExecuteQueryDataSet(query, CommandType.Text).Tables[0];

}

public double TinhTienDien(HoaDon hoaDon)

{

return hoaDon.SoDienTieuThu \* hoaDon.PhongTro.TienDien;

}

public double TinhTienNuoc(HoaDon hoaDon)

{

return hoaDon.SoNuocTieuThu \* hoaDon.PhongTro.TienNuoc;

}

public double TinhTongTien(HoaDon hoaDon)

{

return TinhTienDien(hoaDon) + TinhTienNuoc(hoaDon) + hoaDon.PhongTro.TienRac + hoaDon.PhongTro.TienThue;

}

public HoaDon TimHoaDonTheoMaSo(string maSo)

{

string query = "select \* from HoaDons" +

"where MaSo = " +

"'" + maSo + "'";

DataRow res = db.GetSingleRecord(query, CommandType.Text);

if (res != null)

{

var fields = res.ItemArray;

return new HoaDon(fields[0].ToString(), (int)fields[1], (int)fields[2], (DateTime)fields[3], (DateTime)fields[4], (bool)fields[5], (DateTime)fields[6], (string)fields[7]);

}

return null;

}

public void ThanhToanHoaDon(HoaDon hoaDon)

{

string query = "update HoaDons set DaThanhToan = 1, NgayThanhToan = '" + DateTime.Today + "' " +

"where MaSo = " + "'" + hoaDon.MaSo + "'";

hoaDon.DaThanhToan = true;

hoaDon.NgayThanhToan = DateTime.Today;

db.MyExecuteNonQuery(query, CommandType.Text);

}

}

}