

TRƯỜNG ĐẠI HỌC THỦ DẦU MỘT
VIỆN KỸ THUẬT CÔNG NGHỆ



ĐỒ ÁN MÔN HỌC
LẬP TRÌNH WINDOWS

Đề tài:

PHẦN MỀM QUẢN LÝ KHÁCH SẠN

GVHD: Hồ Ngọc Trung Kiên

SVTH: 1. Nguyễn Hữu Nghĩa 2124802050013
2. Phạm Tuấn Vũ 2124802050008
3. Nguyễn Minh Tuyền 2124801040049

Tháng 07/2023

MỤC LỤC

LỜI NÓI ĐẦU.....	3
CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI, NHỮNG ỨNG DỤNG TRONG THỰC TẾ CỦA ĐỀ TÀI, HƯỚNG PHÁT TRIỂN,...	4
1.1 Khảo sát hiện tượng thực tế	4
1.2 Ý nghĩa ứng dụng.....	4
1.3 Chức năng phần mềm ứng dụng	5
1.4 Cơ sở dữ liệu	6
CHƯƠNG 2: XÂY DỰNG CHƯƠNG TRÌNH ỨNG DỤNG	8
2.1 Một số đoạn code chính của chương trình	8
2.1.1 Source code connect database	8
2.1.2 Source code đăng nhập	9
2.1.3 Source code form main	12
2.1.4 Source code form danh sách khách thuê phòng	16
2.1.5 Source code form danh sách loại phòng	22
2.1.6 Source code form danh sách nhân viên.....	28
2.1.7 Source code form danh sách phòng	34
2.1.8 Source code form danh sách thuê phòng	45
2.1.9 Source code form gia hạn phòng	59
2.1.10 Source code form hoá đơn	59
2.1.11 Source code form tài khoản	62
2.1.12 Source code form thanh toán	68
2.1.13 Source code form thêm tài khoản.....	70
2.1.14 Source code form thống kê.....	76
2.1.15 Source code form trả phòng.....	79
CHƯƠNG 3: TỔNG KẾT.....	81
3.1. Ưu điểm:	81
3.2. Nhược điểm:	82
TÀI LIỆU THAM KHẢO	84

Trong xu thế phát triển hiện nay trên thế giới khoa học và công nghệ luôn có những thay đổi mạnh mẽ. Một phần trong đó là việc ứng dụng Công Nghệ Thông Tin vào đời sống hàng ngày của con người. Loài người chúng ta đang hướng tới thiết lập một hành tinh thông minh. Ngày nay với sự phát triển mạnh mẽ của CNTT kết hợp với sự phát triển của mạng Internet đã kết nối được toàn thế giới lại với nhau thành một thể thống nhất. Nó đã trở thành công cụ đắc lực cho nhiều ngành nghề : giao thông, quân sự, y học, truyền thông...và đặc biệt là trong công tác quản lý nói chung và quản lý quán Khách sạn nói riêng.

Trước đây khi máy tính chưa được ứng dụng rộng rãi các công việc quản lý đều được thực hiện một cách thủ công nên rất tốn thời gian, nhân lực cũng như tài chính. Ngày nay với sự phát triển mạnh mẽ của công nghệ thông tin đã giúp cho việc quản lý được thực hiện một cách dễ dàng hơn, giảm chi phí, thời gian...

Qua quá trình khảo sát một những khách sạn trong địa phận Bình Dương, nhóm chúng em đã xây dựng lên đề tài quản lý khách sạn với mong muốn giúp cho việc quản lý được thực hiện một cách dễ dàng hơn, thuận tiện và giảm thiểu được các sai sót.

Nhờ sự quan tâm, hướng dẫn của Hồ Ngọc Trung Kiên , chúng em đã từng bước nghiên cứu và vận dụng các kiến thức đã được học để tìm hiểu, phân tích và xây dựng được chương trình quản lý đáp ứng tương đối một số các yêu cầu đặt ra. Tuy nhiên, do kiến thức còn hạn chế nên chương trình vẫn không tránh khỏi những thiếu sót. Vì vậy chúng em rất mong nhận được sự đóng góp ý kiến của tất cả các thầy cô và các bạn để có thể từng bước xây dựng chương trình ngày càng hoàn thiện và hiệu quả hơn.

Em xin chân thành cảm ơn Thầy đã giúp chúng em.

CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI, NHỮNG ỨNG DỤNG TRONG THỰC TẾ CỦA ĐỀ TÀI, HƯỚNG PHÁT TRIỂN,...

1.1 KHẢO SÁT HIỆN TƯỢNG THỰC TẾ

Hiện nay, việc quản lý khách sạn vẫn còn đang được thực hiện bằng cách sử dụng phương pháp quản lý truyền thống, như việc quản lý đơn đặt hàng và việc nhân viên ghi nhận thông tin khách hàng và đơn hàng trên giấy tờ.

Tuy nhiên, để cải thiện hiệu quả quản lý cho một khách sạn, một số doanh nghiệp đã bước đầu áp dụng các phần mềm ứng dụng quản lý khách sạn để quản lý các hoạt động kinh doanh. Các phần mềm này có thể giúp khách sạn quản lý đơn hàng, quản lý bàn, giá cả, quản lý nhân viên và cải thiện chất lượng dịch vụ.

Tuy nhiên, vẫn còn một số khó khăn và hạn chế trong việc sử dụng phần mềm quản lý quán cà phê hiện nay, bao gồm:

- ✓ Chi phí đầu tư ban đầu để triển khai phần mềm và đào tạo nhân viên sử dụng phần mềm.
- ✓ Thời gian và chi phí để cập nhật và bảo trì phần mềm.
- ✓ Yêu cầu kỹ năng sử dụng máy tính và khả năng sử dụng phần mềm của nhân viên.
- ✓ Hạn chế của phần mềm quản lý khách sạn trong việc đáp ứng các yêu cầu cụ thể của khách sạn hoặc tính năng đặc biệt mà quán muốn sử dụng.

Tuy nhiên, với sự phát triển của công nghệ thông tin, các phần mềm ứng dụng quản lý khách sạn đang có xu hướng phát triển và hoàn thiện hơn để đáp ứng được yêu cầu thực tế của các khách sạn.

1.2 Ý NGHĨA ỨNG DỤNG

Các ý nghĩa của ứng dụng quản lý quán cà phê bao gồm:

1. Quản lý khách hàng sử dụng phòng: Ứng dụng giúp quản lý từng khách hàng thuê phòng một cách hiệu quả và dễ dàng.
2. Quản lý thời gian thuê phòng của từng khách: Ứng dụng giúp quản lý thời gian sử dụng phòng của từng cá nhân khách hàng, giúp nhân viên cũng như khách hàng gia hạn thêm thời gian sử dụng phòng dễ dàng hơn.

3. Quản lý phòng hiệu quả: Quản lý số lượng phòng còn trống hoặc phòng đã cho thuê giúp cho nhân viên có thể cung cấp cho khách hàng phòng hiện tại khách có thể sử dụng được.
4. Quản lý nhân viên: Ứng dụng hỗ trợ phân công công việc và quản lý giờ làm của từng nhân viên, vị trí làm việc của từng nhân viên.
5. Báo cáo và thống kê: Ứng dụng giúp báo cáo, tổng kết doanh thu theo từng thời gian cụ thể tình hình hoạt động của khách sạn để có thể phát triển khách sạn tốt hơn.
6. Quản lý khách hàng: Ứng dụng giúp quản lý thông tin của từng khách hàng đồng thời bảo mật thông tin cho họ nhằm phục vụ trong quá trình khách hàng sử dụng tốt cho cả hai bên.

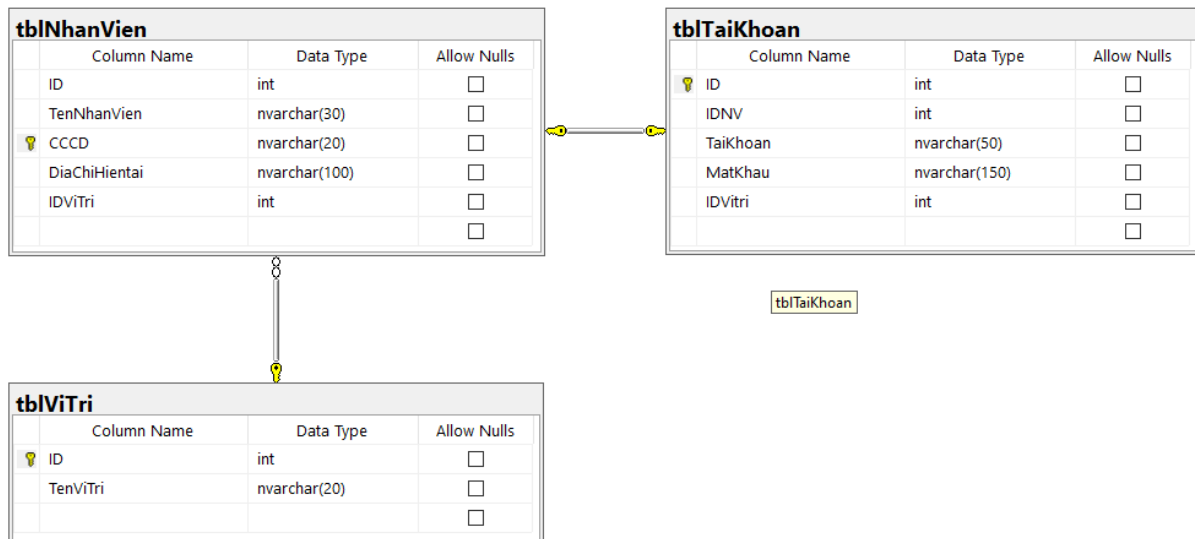
1.3 CHỨC NĂNG PHẦN MỀM ỨNG DỤNG

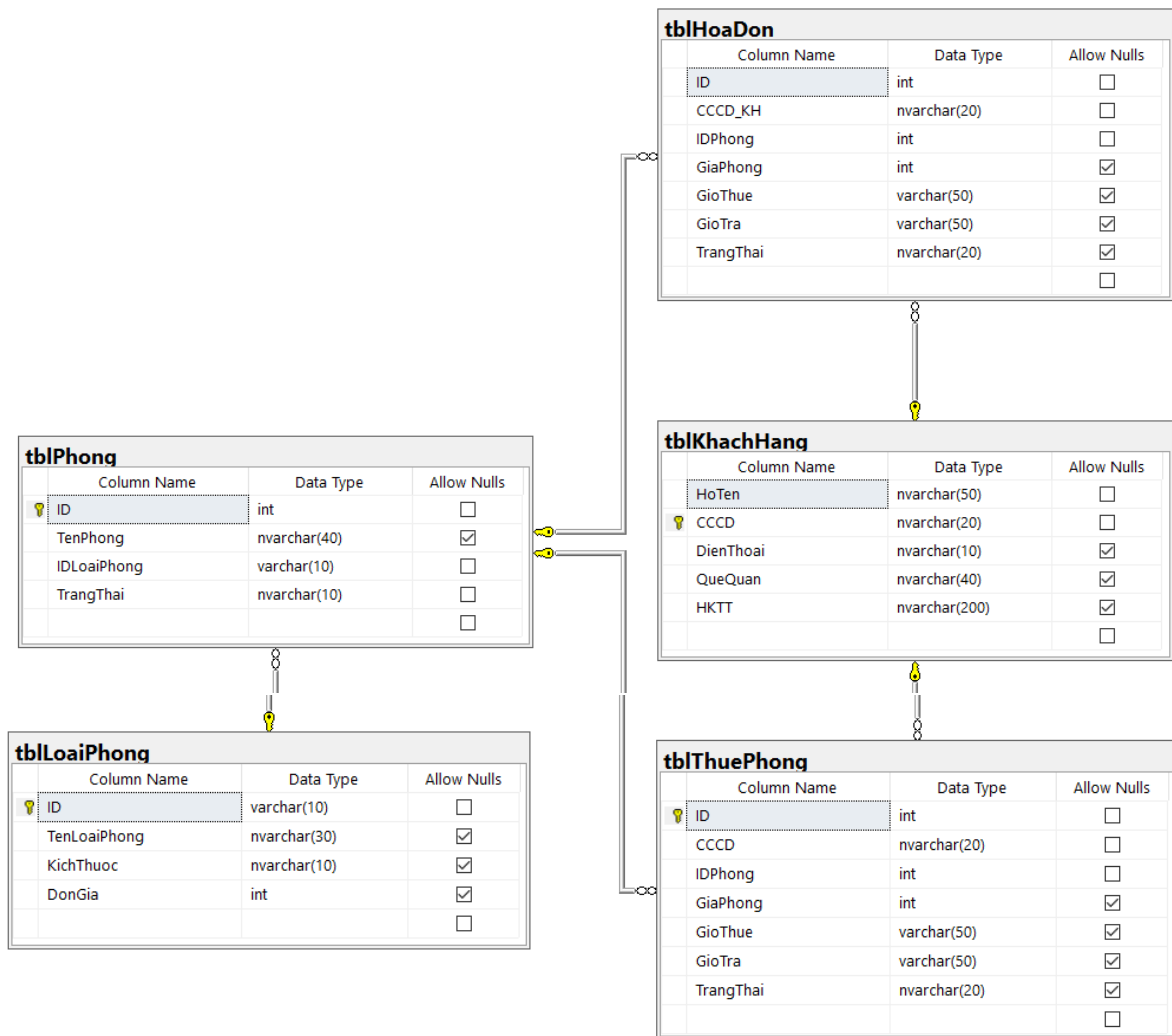
Phần mềm ứng dụng quản lý quán cà phê có các chức năng sau:

1. Quản lý phòng: cho phép nhân viên biết được tình trạng phòng còn trống hay đã cho thuê giúp nhân viên dễ dàng thông báo cung để khách hàng.
2. Quản lý thời gian sử dụng phòng của khách hàng: Quản lý cụ thể từng khách hàng sử dụng phòng trong bao lâu.
3. Quản lý nhân viên: cho phép quản lý thông tin của các nhân viên trong quán, bao gồm thông tin cá nhân, giờ làm việc, lương và các quyền hạn khác.
4. Báo cáo và thống kê: cho phép quản lý xem các báo cáo và thống kê về doanh thu, số lượng đơn hàng, số lượng khách hàng, và các hoạt động kinh doanh khác của khách sạn.
5. Quản lý khách hàng: cho phép quản lý thông tin khách hàng, bao gồm tên, địa chỉ, số điện thoại, cccd và thời điểm cụ thể khách hàng đó sử dụng dịch vụ của khách sạn, giúp khách sạn nâng cao chất lượng dịch vụ và tạo động lực cho khách hàng thường xuyên quay lại sử dụng dịch vụ.

1.4 CƠ SỞ DỮ LIỆU

- tblNhanVien
- tblTaiKhoan
- tblViTri
- tblHoaDon
- tblPhong
- tblLoaiPhong
- tblKhachHang
- tblThuePhong





2.1 Một số đoạn code chính của chương trình

2.1.1 Source code connect database

```
namespace Hotel_manager
{
    public class ConnectDatabase
    {
        private SqlConnection con;
        public SqlConnection connect() //Hàm kết nối
        {
            string strcon = @"Server = DESKTOP-IC4OH16\MSSQLSERVER01; Database
= Hotel_Management1; Integrated Security = True;";
            con = new SqlConnection(strcon);
            try
            {
                con.Open();
            }
            catch (SqlException ex)
            {
                MessageBox.Show("Cannot connect to database! Lỗi: " +
ex, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
                Application.Exit();
            }
            return con;
        }
        public void disconnect() //Hàm ngắt kết nối
        {
            con.Close(); //Ngat ket noi
            con.Dispose(); //Giai phong doi tuong
        }
    }
}
```

- Đoạn mã trên định nghĩa một lớp ConnectDatabase trong namespace Hotel_manager để kết nối và ngắt kết nối với cơ sở dữ liệu. Dưới đây là giải thích chi tiết từng phần trong đoạn mã:
- private SqlConnection con;: Đây là một biến private kiểu SqlConnection được sử dụng để lưu trữ kết nối đến cơ sở dữ liệu.
- public SqlConnection connect(): Đây là một phương thức public trả về một đối tượng SqlConnection để kết nối đến cơ sở dữ liệu. Trong phương thức này:
- Chuỗi kết nối strcon chứa thông tin về máy chủ SQL (Server), tên cơ sở dữ liệu (Database) và phương thức xác thực (Integrated Security).

- Một đối tượng SqlConnection mới được tạo với chuỗi kết nối strcon.
- Trong khối try, kết nối con được mở bằng cách gọi phương thức Open().
- Nếu xảy ra lỗi kết nối (được ném ra dưới dạng SQLException), một hộp thoại thông báo lỗi được hiển thị và ứng dụng sẽ thoát bằng cách gọi Application.Exit().
- Nếu kết nối thành công, đối tượng kết nối con được trả về.
- public void disconnect(): Đây là một phương thức public không trả về giá trị, được sử dụng để ngắt kết nối với cơ sở dữ liệu. Trong phương thức này:
- Gọi phương thức Close() trên đối tượng kết nối con để đóng kết nối với cơ sở dữ liệu.
- Gọi phương thức Dispose() trên đối tượng kết nối con để giải phóng tài nguyên.

2.1.2 Source code đăng nhập

```
public partial class Form_Login1 : Form
{
    #region GUI
    public const int WM_NCLBUTTONDOWN = 0xA1;
    public const int HT_CAPTION = 0x2;

    [DllImportAttribute("user32.dll")]
    public static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);
    [DllImportAttribute("user32.dll")]
    public static extern bool ReleaseCapture();

    private void ptbclose_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Bạn muốn thoát chương trình?", "Cảnh Báo",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            Application.Exit();
    }

    private void ptbminimize_Click(object sender, EventArgs e)
    {
        this.WindowState = FormWindowState.Minimized;
    }

    private void ptbstate_Click(object sender, EventArgs e)
    {
        if (this.WindowState == FormWindowState.Normal)
        {
            this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;
        }
    }
}
```

```

        this.WindowState = FormWindowState.Maximized;
    }
    else
    {
        this.WindowState = FormWindowState.Normal;
    }
}

private void pntop_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        Cursor.Current = Cursors.SizeAll;
        ReleaseCapture();
        SendMessage(Handle, WM_NCLBUTTONDOWN, HT_CAPTION, 0);
    }
}

private void pntop_MouseDoubleClick(object sender, MouseEventArgs e)
{
    ptbstate_Click(null, null);
}

#endregion
public Form_Login1()
{
    InitializeComponent();
}

private ConnectDatabase conn = new ConnectDatabase();

public bool Login()
{
    string query = "SELECT * FROM tblTaiKhoan WHERE TaiKhoan = " +
txtTendangnhap.Text + " AND MatKhau = " + txtMatkhau.Text + """;
    SqlCommand cmd = new SqlCommand(query, conn.connect());
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        dr.Close();
        return true;
    }
    dr.Close();
    return false;
}

```

```

private void btnDangnhap_Click(object sender, EventArgs e)
{
    if (Login())
    {
        DataSet ds = new DataSet();
        SqlCommand cmd1 = new SqlCommand();
        cmd1.CommandText = "Select TenViTri from tblTaiKhoan, tblViTri where
tblTaiKhoan.IDViTri = tblViTri.ID and TaiKhoan = '" + txtTendangnhap.Text + "' and
MatKhau = '" + txtMatkhau.Text + "'";
        cmd1.Connection = conn.connect();
        SqlDataAdapter da = new SqlDataAdapter();
        da.SelectCommand = cmd1;
        da.Fill(ds, "ViTri");
        Const.loaitaikhoan = ds.Tables["ViTri"].Rows[0][0].ToString();
        var f = new Form_Main();
        f.Show();
        this.Hide();
        f.logout += F_logout;
    }
    else
    {
        MessageBox.Show("Đăng nhập thất bại. Tên đăng nhập hoặc mật khẩu không
đúng!!!", "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

private void F_logout(object? sender, EventArgs e)
{
    (sender as Form_Main).isExit = false;
    (sender as Form_Main).Close();
    this.Show();
}

private void Form_Login1_Load(object sender, EventArgs e)
{
    txtTendangnhap.Focus();
    menuStrip1.Enabled = false;
}

```

- Đoạn mã trên là một phần của lớp Form_Login1 và có các chức năng sau:
- Xử lý sự kiện giao diện cho các nút ptbclose, ptbminimize, ptbstate, pntop_MouseDown, và pntop_MouseDoubleClick. Đây là các chức năng liên quan đến việc di chuyển và tương tác với cửa sổ ứng dụng.
- Kết nối và ngắt kết nối với cơ sở dữ liệu thông qua lớp ConnectDatabase.

- Kiểm tra thông tin đăng nhập nhập vào với dữ liệu trong cơ sở dữ liệu trong hàm Login().
- Xử lý sự kiện click vào nút "Đăng nhập" (btnDangnhap_Click), kiểm tra thông tin đăng nhập và hiển thị form chính (Form_Main) nếu đăng nhập thành công.
- Xử lý sự kiện đăng xuất (F_logout) khi form chính đóng lại, hiển thị lại form đăng nhập và khôi phục trạng thái ban đầu.
- Thiết lập trạng thái khởi tạo của form, tập trung vào ô nhập tên đăng nhập và vô hiệu hóa menuStrip1 trong hàm Form_Login1_Load.

2.1.3 Source code form main

```
public partial class Form_Main : Form
{
    #region GUI
    public const int WM_NCLBUTTONDOWN = 0xA1;
    public const int HT_CAPTION = 0x2;
    public bool isExit = true;
    public event EventHandler logout;
    [DllImportAttribute("user32.dll")]
    public static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);
    [DllImportAttribute("user32.dll")]
    public static extern bool ReleaseCapture();
    public Form_Main()
    {
        InitializeComponent();
    }

    private void ptbclose_Click(object sender, EventArgs e)
    {
        if (isExit)
        {
            if (MessageBox.Show("Bạn muốn thoát chương trình?", "Cảnh Báo",
                MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
                Application.Exit();
        }
    }

    private void ptbminimize_Click(object sender, EventArgs e)
    {
        this.WindowState = FormWindowState.Minimized;
    }

    private void ptbstate_Click(object sender, EventArgs e)
    {
        if (this.WindowState == FormWindowState.Normal)
        {
            this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;
        }
    }
}
```

```

        this.WindowState = FormWindowState.Maximized;
    }
    else
    {
        this.WindowState = FormWindowState.Normal;
    }
}

private void pntop_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        Cursor.Current = Cursors.SizeAll;
        ReleaseCapture();
        SendMessage(Handle, WM_NCLBUTTONDOWN, HT_CAPTION, 0);
    }
}

private void pntop_MouseDoubleClick(object sender, MouseEventArgs e)
{
    ptbstate_Click(null, null);
}

#endregion

private void Form_Main_Load(object sender, EventArgs e)
{
    /*var f1 = new Form_Login1();
    //AddForm(f1);
    f1.ShowDialog();*/
    Phanquyen();
}

private void Phanquyen()
{
    if (Const.loaitaikhoan == "Lễ Tân")
    {
        doanhThuToolStripMenuItem.Enabled = false;
        danhMucToolStripMenuItem.Enabled = false;
        phòngĐăngThuêToolStripMenuItem_Click(null, null);
    }
    else if (Const.loaitaikhoan == "Quản Lý")
    {
        tàiKhoảnToolStripMenuItem.Enabled = false;
        vịTríViệcLàmToolStripMenuItem.Enabled = false;
    }
    else if (Const.loaitaikhoan == "CSKH")
    {

```

```

        nhânViênToolStripMenuItem.Enabled = false;
        thôngKêToolStripMenuItem.Enabled = false;
        phòngToolStripMenuItem.Enabled = false;
        tàiKhoảnToolStripMenuItem.Enabled = false;
        vịTríViệcLàmToolStripMenuItem.Enabled = false;
        quản LÝKháchHàngToolStripMenuItem_Click(null, null);
    }
}

public void AddForm(Form f)
{
    this.grbcontent.Controls.Clear();//xoa cac control hien co tren groupbox
    f.TopLevel = false;
    f.AutoScroll = true;
    f.FormBorderStyle = FormBorderStyle.None;
    f.Dock = DockStyle.Fill;
    this.Text = f.Text;
    this.grbcontent.Controls.Add(f);
    f.Show();
}

private void ThoátToolStripMenuItem1_Click(object sender, EventArgs e)
{
    Application.Exit();
}

private void loạiPhòngToolStripMenuItem_Click(object sender, EventArgs e)
{
    var f = new Form_DanhSachLoaiPhong();
    AddForm(f);
}

public void trangChủToolStripMenuItem_Click(object sender, EventArgs e)
{
    var f = new Form_Wellcome();
    AddForm(f);
}

private void khoáMànHìnhToolStripMenuItem_Click(object sender, EventArgs e)
{
    /*var f = new Form_Login1();
    //AddForm(f);
    f.ShowDialog();*/
    logout(this, new EventArgs());
}

private void danhSáchPhòngToolStripMenuItem_Click(object sender, EventArgs e)

```

```

{
    var f = new Form_DanhSachPhong();
    AddForm(f);
}

private void grbcontent_Enter(object sender, EventArgs e)
{
}

private void dịchVụToolStripMenuItem_Click(object sender, EventArgs e)
{
    var f = new Form_DanhSachDichVu();
    AddForm(f);
}

private void menuStrip1_ItemClicked(object sender,
ToolStripItemClickedEventArgs e)
{
}

private void quản LÝ Khách Hàng ToolStripMenuItem_Click(object sender, EventArgs
e)
{
    var f = new Form_DanhSachKhachThuePhong();
    AddForm(f);
}

private void thuê Phòng ToolStripMenuItem_Click(object sender, EventArgs e)
{
    var f = new Form_DanhSachThuePhong();
    AddForm(f);
}

private void trả Phòng ToolStripMenuItem_Click(object sender, EventArgs e)
{
    var f = new Form_TraPhong1();
    AddForm(f);
}

private void nhân Viên ToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    var f = new Form_DanhSachNhanVien();
    AddForm(f);
}

```

```

private void tàiKhoảnToolStripMenuItem_Click_1(object sender, EventArgs e)
{
    var f = new Form_TaiKhoan();
    AddForm(f);
}

private void vịTríViệcLàmToolStripMenuItem_Click(object sender, EventArgs e)
{
    var f = new Form_ViTri();
    AddForm(f);
}

private void phòngĐangThuêToolStripMenuItem_Click(object sender, EventArgs e)
{
    var f = new Form_ThongKe();
    f.ShowDialog();
}

private void doanhThuToolStripMenuItem_Click(object sender, EventArgs e)
{
    var f = new Form_HoaDon();
    AddForm(f);
}
}
}

```

- Đoạn mã trên là một phần của lớp Form_Main và có các chức năng sau:
- Xử lý sự kiện giao diện cho các nút ptbclose, ptbminimize, ptbstate, pntop_MouseDown, và pntop_MouseDoubleClick. Đây là các chức năng liên quan đến việc di chuyển và tương tác với cửa sổ ứng dụng.
- Thiết lập trạng thái khởi tạo của form và gọi hàm Phanquyen() trong hàm Form_Main_Load.
- Phân quyền người dùng dựa trên giá trị của biến Const.loaitaikhoan trong hàm Phanquyen(). Các menu và chức năng sẽ được vô hiệu hóa hoặc hiển thị dựa trên vai trò của tài khoản đăng nhập.
- Hàm AddForm() để thay thế nội dung hiện tại trong grbcontent bằng form mới được chọn.
- Xử lý các sự kiện click trên menu và gọi hàm AddForm() để thay đổi nội dung hiển thị theo menu đã chọn.
- Xử lý sự kiện click nút "Khoá Màn hình" để đăng xuất khỏi tài khoản hiện tại và hiển thị lại form đăng nhập.
- Xử lý sự kiện click nút "Thoát" để đóng ứng dụng (nếu biến isExit là true).
- Các hàm và sự kiện còn lại không có mô tả trong mã ngắn gọn.
-

2.1.4 Source code form danh sách khách thuê phòng

```
public partial class Form_DanhSachKhachThuePhong : Form
```



```

{
    private ConnectDatabase conn = new ConnectDatabase();
    private DataSet ds = new DataSet();
    private int kt;
    private void back()
    {
        txtHovaten.Text = string.Empty;
        txtSDT.Text = string.Empty;
        txtCCCD.Text = string.Empty;
        txtQueQuan.Text = string.Empty;
        txtHKTT.Text = string.Empty;
        txtHovaten.Focus();
    }

    private void enablecommand(bool kt)
    {
        btnLuu.Enabled = !kt;
        btnCapnhat.Enabled = kt;
        btnThem.Enabled = kt;
        btnXoa.Enabled = kt;
        grvKhachthuephong.ReadOnly = kt;
        //Form_LoaiPhong.ac
    }

    private void loaddata()
    {
        var con = conn.connect();
        string sql = "Select HoTen as 'Họ và Tên', CCCD as 'CCCD/CMND', DienThoai
as 'Số Điện Thoại' +
        ", QueQuan as 'Quê Quán', HKTT as 'Hộ Khẩu Thường Trú' +
        "FROM tblKhachHang";
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.CommandType = CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(dt);
        con.Close();
        grvKhachthuephong.DataSource = dt;
    }

    private void binding()
    {
        txtHovaten.DataBindings.Clear();
        txtHovaten.DataBindings.Add("Text", grvKhachthuephong.DataSource, "Họ và
Tên");
        txtCCCD.DataBindings.Clear();

```

```

txtCCCD.DataBindings.Add("Text", grvKhachthuephong.DataSource,
"CCCD/CMND");
txtSDT.DataBindings.Clear();
txtSDT.DataBindings.Add("Text", grvKhachthuephong.DataSource, "Số Điện
Thoại");
txtQueQuan.DataBindings.Clear();
txtQueQuan.DataBindings.Add("Text", grvKhachthuephong.DataSource, "Quê
Quán");
txtHKTT.DataBindings.Clear();
txtHKTT.DataBindings.Add("Text", grvKhachthuephong.DataSource, "Hộ Khẩu
Thường Trú");
}

```

```

private bool themkhachhang()
{
    if (string.IsNullOrEmpty(txtHovaten.Text))
    {
        MessageBox.Show("Vui lòng nhập họ khách hàng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return false;
    }
    var con = conn.connect();
    //MessageBox.Show(cbbLoaiphong.SelectedValue.ToString());
    string Check_exist_query = "SELECT CCCD FROM tblKhachHang WHERE
CCCD = " + txtCCCD.Text;
    string Insert_query = "insert into tblKhachHang values" +
        "(N'" + txtHovaten.Text + "', '" + txtCCCD.Text + "', '" + txtSDT.Text + "', N'"
+ txtQueQuan.Text + "', N'" + txtHKTT.Text + "')";
    SqlCommand cmd = new SqlCommand(Check_exist_query, con);
    SqlDataReader dr = cmd.ExecuteReader();
    if (!dr.HasRows)
    {
        dr.Close();
        cmd = new SqlCommand(Insert_query, con);
        if (cmd.ExecuteNonQuery() > 0)
        {
            con.Close();
            return true;
        }
    }
    con.Close();
    MessageBox.Show("CCCD/CMND bị trùng!", "Ràng buộc dữ liệu",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}

```

```

private void btnThem_Click(object sender, EventArgs e)

```

```

{
    back();
    enablecommand(false);
    kt = 1;
}

private bool xoakhachhang()
{
    if (string.IsNullOrEmpty(txtCCCD.Text))
    {
        MessageBox.Show("Vui lòng nhập CCCD/CMND khách hàng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return false;
    }
    var con = conn.connect();
    string Check_exist_query = "SELECT CCCD FROM tblKhachHang WHERE
CCCD = '" + txtCCCD.Text + "'";
    string Delete_query = "Delete from tblKhachHang where CCCD = '" +
txtCCCD.Text + "'";
    SqlCommand cmd = new SqlCommand(Check_exist_query, con);
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        dr.Close();
        cmd = new SqlCommand(Delete_query, con);
        if (cmd.ExecuteNonQuery() > 0)
        {
            con.Close();
            return true;
        }
    }
    con.Close();
    MessageBox.Show("CCCD/CMND không tồn tại!", "Ràng buộc dữ liệu",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}

private void btnXoa_Click(object sender, EventArgs e)
{
    binding();
    if (MessageBox.Show("Bạn thực sự muốn xóa khách hàng có CCCD/CMND là '"
+
        txtCCCD.Text + "' không???", "Warning", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
        if (xoakhachhang())
        {

```

```

        MessageBox.Show("Xóa khách hàng công", "Ràng buộc dữ liệu",
        MessageBoxButtons.OK, MessageBoxIcon.Information);
    }
    loaddata();
    back();
}

private bool capnhat()
{
    txtHovaten.Enabled = false;
    var con = conn.connect();
    string Update_query = "UPDATE tblKhachHang SET DienThoai = N'" +
txtSDT.Text + "', QueQuan = N'" + txtQueQuan.Text + "',HKTT = N'" + txtHKTT.Text
+ "' WHERE CCCD = '" + txtCCCD.Text + "'";
    SqlCommand cmd = new SqlCommand(Update_query, con);
    if (cmd.ExecuteNonQuery() > 0)
    {
        con.Close();
        return true;
    }
    con.Close();
    return false;
}

private void btnCapnhat_Click(object sender, EventArgs e)
{
    binding();
    grvKhachthuephong.Enabled = false;
    txtCCCD.ReadOnly = true;
    enablecommand(false);
    kt = 2;
}

private void btnLuu_Click(object sender, EventArgs e)
{
    if (kt == 1)
    {
        if (themkhachhang())
        {
            MessageBox.Show("Thêm khách hàng thành công", "Ràng buộc dữ liệu",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            enablecommand(true);
            loaddata();
            back();
        }
        else if (MessageBox.Show("Bạn muốn dừng thêm???", "Warning",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            enablecommand(true);
    }
}

```

```

    }
    else if (kt == 2)
    {
        if (capnhat())
        {
            MessageBox.Show("Cập nhật khách hàng thành công", "Ràng buộc dữ liệu",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            enablecommand(true);
            txtCCCD.ReadOnly = false;
            loaddata();
            grvKhachthuephong.Enabled = true;
            back();
        }
    }
}

private void grvPhong_CellClick(object sender, DataGridViewCellEventArgs e)
{
    back();
}

private void btnTim_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtCCCD.Text))
    {
        MessageBox.Show("Vui lòng nhập CCCD/CMND khách hàng!",
        "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
    var con = conn.connect();
    string sql = "Select HoTen as 'Họ và Tên', CCCD as 'CCCD/CMND', DienThoai
as 'Số Điện Thoại', QueQuan as 'Quê Quán', HKTT as 'Hộ Khẩu Thường Trú' FROM
tblKhachHang where CCCD = '" + txtCCCD.Text + "'";
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.CommandType = CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    con.Close();
    grvKhachthuephong.DataSource = dt;
}

public Form_DanhSachKhachThuePhong()
{
    InitializeComponent();
}

private void Form_KhachThuePhong_Load(object sender, EventArgs e)
{

```

```

        loaddata();
        back();
    }
}

```

- Đoạn mã trên là một phần của lớp `Form_DanhSachKhachThuePhong` và có các chức năng sau:
- Các biến và đối tượng được khai báo: `conn` là một đối tượng `ConnectDatabase` để kết nối với cơ sở dữ liệu, `ds` là một đối tượng `DataSet` để lưu dữ liệu từ cơ sở dữ liệu, `kt` là một biến kiểu `int` để xác định trạng thái (1: thêm khách hàng, 2: cập nhật khách hàng).
- Các phương thức `back()`, `enablecommand(bool kt)`, `loaddata()`, `binding()`, `themkhachhang()`, `xoakhachhang()`, `capnhat()`, `btnThem_Click()`, `btnXoa_Click()`, `btnCapnhat_Click()`, `btnLuu_Click()`, `grvPhong_CellClick()`, `btnTim_Click()` có nhiệm vụ thực hiện các tác vụ như xóa trường dữ liệu, kích hoạt hoặc vô hiệu hóa các nút và trường nhập liệu, tải dữ liệu từ cơ sở dữ liệu, thực hiện các thao tác thêm, xóa, cập nhật dữ liệu và tìm kiếm khách hàng.
- Hàm `Form_DanhSachKhachThuePhong_Load()` được gọi khi form được tải lên, nó thực hiện việc tải dữ liệu và đặt lại trạng thái ban đầu cho form.

2.1.5 Source code form danh sách loại phòng

```

public partial class Form_DanhSachLoaiPhong : Form
{
    private ConnectDatabase conn = new ConnectDatabase();
    private int kt;

    public Form_DanhSachLoaiPhong()
    {
        InitializeComponent();
    }

    private void back()
    {
        txtTenloaiphong.Text = string.Empty;
        txtKichthuoc.Text = string.Empty;
        txtMaloaiphong.Text = string.Empty;
        txtDongia.Text = "0";
        txtMaloaiphong.Focus();
    }

    private void enablecommand(bool kt)
    {
        btnLuu.Enabled = !kt;
        btnCapnhat.Enabled = kt;
        btnThem.Enabled = kt;
        btnXoa.Enabled = kt;
        grvLoaiPhong.ReadOnly = kt;
        //Form_LoaiPhong.ac
    }
}

```

```

private void load_LoaiPhong()
{
    var con = conn.connect();
    string sql = "Select ID as 'Mã Loại Phòng', TenLoaiPhong as 'Tên Loại Phòng',
KichThuoc as" +
        "'Kích Thước (người)', DonGia as 'Đơn Giá' from tblLoaiPhong";
    SqlCommand com = new SqlCommand(sql, con);
    com.CommandType = CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(com);
    da.Fill(dt);
    con.Close();
    grvLoaiPhong.DataSource = dt;
}

private void Form_LoaiPhong_Load(object sender, EventArgs e)
{
    load_LoaiPhong();
    grvLoaiPhong.Columns["Đơn Giá"].DefaultCellStyle.Format = "N0";
}

private bool themphong()
{
    if (string.IsNullOrEmpty(txtMaloaiphong.Text))
    {
        MessageBox.Show("Vui lòng nhập mã loại phòng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return false;
    }
    if (int.Parse(txtDongia.Text) < 500000)
    {
        MessageBox.Show("Đơn giá thấp nhất là 500,000",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return false;
    }
    var con = conn.connect();
    string Check_exist_query = "SELECT ID FROM tblLoaiPhong WHERE ID = "
+ txtMaloaiphong.Text + """;
    string Insert_query = "insert into tblLoaiPhong values" +
        "(N" + txtMaloaiphong.Text + ",N" + txtTenloaiphong.Text + ",N" +
txtKichthuoc.Text + ", N" + txtDongia.Text + ")";
    SqlCommand cmd = new SqlCommand(Check_exist_query, con);
    SqlDataReader dr = cmd.ExecuteReader();
    if (!dr.HasRows)
    {
        dr.Close();
    }
}

```

```

        cmd = new SqlCommand(Insert_query, con);
        if (cmd.ExecuteNonQuery() > 0)
        {
            con.Close();
            return true;
        }

    }
    con.Close();
    MessageBox.Show("ID đã tồn tại", "Ràng buộc dữ liệu",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    return false;
}

private void btnThem_Click(object sender, EventArgs e)
{
    back();
    enablecommand(false);
    kt = 1;
}

private bool xoaloaiphong()
{
    var con = conn.connect();
    string Check_exist_query = "SELECT ID FROM tblLoaiPhong WHERE ID = " +
+ txtMaloaiphong.Text + "";
    string Delete_query = "DELETE FROM tblLoaiPhong WHERE ID = " +
txtMaloaiphong.Text + "";
    SqlCommand cmd = new SqlCommand(Check_exist_query, con);
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        dr.Close();
        cmd = new SqlCommand(Delete_query, con);
        if (cmd.ExecuteNonQuery() > 0)
        {
            con.Close();
            return true;
        }
    }
    MessageBox.Show("Mã loại phòng không tồn tại", "Ràng buộc dữ liệu",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
    con.Close();
    return false;
}

```



```

private void btnXoa_Click(object sender, EventArgs e)
{
    binding();
    if (MessageBox.Show("Bạn thực sự muốn xóa loại phòng có mã " +
        txtMaloaiphong.Text + " không???", "Warning", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
        if (xoaloaiphong())
            MessageBox.Show("Xóa loại phòng thành công", "Ràng buộc dữ liệu",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
    load_LoiPhong();
    back();
}

private void binding()
{
    txtMaloaiphong.DataBindings.Clear();
    txtMaloaiphong.DataBindings.Add("Text", grvLoaiPhong.DataSource, "Mã Loại
Phòng");
    txtTenloaiphong.DataBindings.Clear();
    txtTenloaiphong.DataBindings.Add("Text", grvLoaiPhong.DataSource, "Tên Loại
Phòng");
    txtKichthuoc.DataBindings.Clear();
    txtKichthuoc.DataBindings.Add("Text", grvLoaiPhong.DataSource, "Kích Thước
(người)");
    txtDongia.DataBindings.Clear();
    txtDongia.DataBindings.Add("Text", grvLoaiPhong.DataSource, "Đơn Giá");
    txtMaloaiphong.Focus();
}

private bool capnhat()
{
    var con = conn.connect();
    string Update_query = "UPDATE tblLoaiPhong SET ID = " +
        txtMaloaiphong.Text +
        ", TenLoaiPhong = N" + txtTenloaiphong.Text + ", KichThuoc = " +
        txtKichthuoc.Text + ", DonGia = " + txtDongia.Text + "" +
        "WHERE ID = " + txtMaloaiphong.Text + "";
    SqlCommand cmd = new SqlCommand(Update_query, con);
    if (cmd.ExecuteNonQuery() > 0)
    {
        con.Close();
        return true;
    }
    con.Close();
    return false;
}

```

```

private void btnCapnhat_Click(object sender, EventArgs e)
{
    binding();
    grvLoaiPhong.Enabled = false;
    txtMaloaiphong.ReadOnly = true;
    enablecommand(false);
    kt = 2;
}

private void btnLuu_Click(object sender, EventArgs e)
{
    if (kt == 1)
    {
        if (themphong())
        {
            MessageBox.Show("Thêm phòng thành công", "Ràng buộc dữ liệu",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            enablecommand(true);
            load_LoaiPhong();
            back();
        }
        else if (MessageBox.Show("Bạn muốn dừng thêm???", "Warning",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            enablecommand(true);
    }
    else if (kt == 2)
    {
        if (capnhat())
        {
            MessageBox.Show("Cập nhật phòng thành công", "Ràng buộc dữ liệu",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            enablecommand(true);
            txtMaloaiphong.ReadOnly = false;
            load_LoaiPhong();
            grvLoaiPhong.Enabled = true;
            back();
        }
    }
}

private void grvLoaiPhong_CellClick(object sender, DataGridViewCellEventArgs
e)
{
    back();
}

private void btnTim_Click(object sender, EventArgs e)

```

```

{
    if (string.IsNullOrEmpty(txtMaloaiphong.Text) &&
string.IsNullOrEmpty(txtTenloaiphong.Text))
    {
        MessageBox.Show("Vui lòng nhập mã loại phòng hoặc tên loại phòng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    var con = conn.connect();
    string sql = "Select ID as 'Mã Loại Phòng', TenLoaiPhong as 'Tên Loại Phòng' +
        "FROM tblLoaiPhong where ID = '" + txtMaloaiphong.Text + "' or
TenLoaiPhong = N'" + txtTenloaiphong.Text + "'";
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.CommandType = CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    con.Close();
    grvLoaiPhong.DataSource = dt;
}
}

```

- Lớp Form_DanhSachLoaiPhong quản lý danh sách các loại phòng trong ứng dụng quản lý khách sạn. Dưới đây là các chức năng và phương thức chính trong lớp này:
- Các biến và đối tượng được khai báo: conn là một đối tượng ConnectDatabase để kết nối với cơ sở dữ liệu, kt là một biến kiểu int để xác định trạng thái (1: thêm loại phòng, 2: cập nhật loại phòng).
- Phương thức back(), enablecommand(bool kt), load_LoaiPhong(), Form_LoaiPhong_Load(), themphong(), xoaloaiphong(), btnXoa_Click(), binding(), capnhat(), btnCapnhat_Click(), btnLuu_Click(), grvLoaiPhong_CellClick(), btnTim_Click() có nhiệm vụ thực hiện các tác vụ như xóa trường dữ liệu, kích hoạt hoặc vô hiệu hóa các nút và trường nhập liệu, tải dữ liệu từ cơ sở dữ liệu, thực hiện các thao tác thêm, xóa, cập nhật dữ liệu và tìm kiếm loại phòng.
- Hàm Form_DanhSachLoaiPhong() là hàm khởi tạo của lớp.
- Hàm Form_LoaiPhong_Load() được gọi khi form được tải lên, nó thực hiện việc tải dữ liệu và đặt lại trạng thái ban đầu cho form.
- Hàm themphong() kiểm tra và thêm một loại phòng mới vào cơ sở dữ liệu.
- Hàm xoaloaiphong() kiểm tra và xóa một loại phòng khỏi cơ sở dữ liệu.
- Hàm capnhat() cập nhật thông tin của một loại phòng trong cơ sở dữ liệu.

- Các sự kiện btnThem_Click(), btnXoa_Click(), btnCapnhat_Click(), btnLuu_Click(), grvLoaiPhong_CellClick(), btnTim_Click() được gắn kết với các nút và điều khiển trên form để thực hiện các tác vụ tương ứng khi người dùng tương tác với chúng.

2.1.6 Source code form danh sách nhân viên

```
public partial class Form_DanhSachNhanVien : Form
{
    private ConnectDatabase conn = new ConnectDatabase();
    private int kt;
    public Form_DanhSachNhanVien()
    {
        InitializeComponent();
    }

    private void back()
    {
        txtIDnhanvien.Text = string.Empty;
        txtTennhanvien.Text = string.Empty;
        cbbVitri.Text = string.Empty;
        txtDiachihientai.Text = string.Empty;
        txtCCCD.Text = string.Empty;
        txtTennhanvien.Focus();
    }

    private void enablecommand(bool kt)
    {
        btnXacnhan.Enabled = !kt;
        btnThem.Enabled = kt;
        btnCapnhat.Enabled = kt;
        btnXoa.Enabled = kt;
    }

    private void binding()
    {
        Int32 i = Convert.ToInt32(grvNhanvien.CurrentRow.Index);
        txtIDnhanvien.DataBindings.Clear();
        txtIDnhanvien.DataBindings.Add("Text", grvNhanvien.DataSource, "Mã Nhân
        Viên");

        cbbVitri.Text = grvNhanvien.Rows[i].Cells["Vị Trí"].Value.ToString();
        txtCCCD.DataBindings.Clear();
        txtCCCD.DataBindings.Add("Text", grvNhanvien.DataSource,
        "CCCD/CMND");
        txtTennhanvien.DataBindings.Clear();
        txtTennhanvien.DataBindings.Add("Text", grvNhanvien.DataSource, "Tên Nhân
        Viên");
    }
}
```

```

txtDiachihientai.DataBindings.Clear();
txtDiachihientai.DataBindings.Add("Text", grvNhanvien.DataSource, "Địa Chỉ
Hiện Tại");
}

private void addcbbVitri()
{
    DataSet dataSet = new DataSet();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "Select * from tblViTri";
    cmd.Connection = conn.connect();
    SqlDataAdapter da = new SqlDataAdapter();
    da.SelectCommand = cmd;
    da.Fill(dataSet, "ViTri");
    cbbVitri.DataSource = dataSet.Tables["ViTri"];
    cbbVitri.DisplayMember = "TenViTri";
    cbbVitri.ValueMember = "ID";
}

private void loaddata()
{
    var con = conn.connect();
    string sql = "select tblNhanVien.ID as 'Mã Nhân Viên', TenNhanVien as 'Tên
Nhân Viên', CCCD as 'CCCD/CMND', DiaChiHienTai as 'Địa Chỉ Hiện Tại', TenViTri
as 'Vị Trí' from tblNhanVien, tblViTri where tblViTri.ID = tblNhanVien.IDViTri";
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.CommandType = CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    con.Close();
    grvNhanvien.DataSource = dt;
}

private bool themnhanvien()
{
    if (string.IsNullOrEmpty(txtTennhanvien.Text) ||
string.IsNullOrEmpty(txtCCCD.Text) || string.IsNullOrEmpty(txtDiachihientai.Text))
    {
        MessageBox.Show("Vui lòng nhập đầy đủ thông tin nhân viên!", "Lưu Ý",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    var con = conn.connect();
    string Check_exist_query = "SELECT CCCD FROM tblNhanVien WHERE
CCCD = '" + txtIDnhanvien.Text + "'";

```

```

        string Insert_query = "insert into tblNhanVien values(" +
int.Parse(txtIDnhanvien.Text) + ", " + txtTennhanvien.Text + ", " + txtCCCD.Text + ",
" + txtDiachihientai.Text + ", " + int.Parse(cbbVitri.SelectedValue.ToString()) + ")";
        SqlCommand cmd = new SqlCommand(Check_exist_query, con);
        SqlDataReader dr = cmd.ExecuteReader();
        if (!dr.HasRows)
        {
            dr.Close();
            cmd = new SqlCommand(Insert_query, con);
            if (cmd.ExecuteNonQuery() > 0)
            {
                con.Close();
                return true;
            }
        }
        con.Close();
        MessageBox.Show("CCCD của bạn bị trùng!", "Lỗi", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        return false;
    }

```

```

private void btnThem_Click(object sender, EventArgs e)
{
    back();
    if (grvNhanvien.RowCount == 1)
    {
        txtIDnhanvien.Text = "1";
    }
    else
    {
        txtIDnhanvien.Text =
(int.Parse(grvNhanvien.Rows[0].Cells[0].Value.ToString()) + 1).ToString();
    }
    addcbbVitri();
    enablecommand(false);
    kt = 1;
}

```

```

private bool xoanhanvien()
{
    var con = conn.connect();
    string Delete_query = "Delete FROM tblNhanVien WHERE ID = " +
txtIDnhanvien.Text;
    SqlCommand cmd = new SqlCommand();
    cmd = new SqlCommand(Delete_query, con);
    if (cmd.ExecuteNonQuery() > 0)
    {

```

```

        con.Close();
        return true;
    }
    con.Close();
    return false;
}

private void btnXoa_Click(object sender, EventArgs e)
{
    binding();
    if (MessageBox.Show("Bạn thực sự muốn xóa nhân viên có mã " +
        txtIDnhanvien.Text + " không???", "Warning", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
        if (xoanhanvien())
        {
            MessageBox.Show("Xóa nhân viên có mã " + txtIDnhanvien.Text + " thành
công", "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        loaddata();
        back();
    }

private void btnXacnhan_Click(object sender, EventArgs e)
{
    if (kt == 1)
    {
        if (themnhanvien())
        {
            MessageBox.Show("Thêm nhân viên thành công", "Ràng buộc dữ liệu",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            enablecommand(true);
            txtIDnhanvien.ReadOnly = false;
            loaddata();
            back();
        }
        else if (MessageBox.Show("Bạn muốn dừng thêm???", "Warning",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            enablecommand(true);
    }
    else if (kt == 2)
    {
        if (capnhat())
        {
            MessageBox.Show("Cập nhật thông tin nhân viên thành công", "Ràng buộc
dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Information);
            enablecommand(true);
            back();
        }
    }
}

```

```

        txtCCCD.Enabled = txtIDnhanvien.Enabled = txtTennhanvien.Enabled =
true;
    }
}

private bool capnhat()
{
    var con = conn.connect();
    string update_query = "update tblNhanVien set IDViTri = " +
int.Parse(cbbVitri.SelectedValue.ToString()) + ", DiaChiHienTai = N" +
txtDiachihientai.Text + "" +
    "where ID = " + int.Parse(txtIDnhanvien.Text);
    SqlCommand cmd = new SqlCommand(update_query, con);
    if (cmd.ExecuteNonQuery() > 0)
    {
        con.Close();
        Int32 i = Convert.ToInt32(grvNhanvien.CurrentRow.Index);
        loaddata();
        grvNhanvien.Rows[i].Selected = true;
        return true;
    }
    return false;
}

private void btnCapnhat_Click(object sender, EventArgs e)
{
    txtCCCD.Enabled = txtIDnhanvien.Enabled = txtTennhanvien.Enabled = false;
    addcbbVitri();
    binding();
    enablecommand(false);
    kt = 2;
}

private void Form_DanhSachNhanVien_Load(object sender, EventArgs e)
{
    loaddata();
}

private void btnTim_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtIDnhanvien.Text) &&
string.IsNullOrEmpty(txtCCCD.Text) && string.IsNullOrEmpty(txtTennhanvien.Text))
    {
        MessageBox.Show("Vui lòng nhập Thông tin nhân viên!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }
}

```



```

    }
    var con = conn.connect();
    string sql = "select tblNhanVien.ID as 'Mã Nhân Viên', TenNhanVien as 'Tên Nhân Viên', CCCD as 'CCCD/CMND', " +
        "DiaChiHienTai as 'Địa Chỉ Hiện Tại', TenViTri as 'Vị Trí' " +
        "from tblNhanVien, tblViTri " +
        "where tblViTri.ID = tblNhanVien.IDViTri and(tblNhanVien.ID = " +
        int.Parse(txtIDnhanvien.Text) + " or CCCD = N'" + txtCCCD.Text + "' or TenNhanVien = N'" + txtTennhanvien.Text + "'";
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.CommandType = CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    con.Close();
    grvNhanvien.DataSource = dt;
    }
}

```

- Lớp Form_DanhSachNhanVien quản lý danh sách nhân viên trong ứng dụng quản lý khách sạn. Dưới đây là các chức năng và phương thức chính trong lớp này:
- Các biến và đối tượng được khai báo: conn là một đối tượng ConnectDatabase để kết nối với cơ sở dữ liệu, kt là một biến kiểu int để xác định trạng thái (1: thêm nhân viên, 2: cập nhật thông tin nhân viên).
- Phương thức back(), enablecommand(bool kt), binding(), addcbbVitri(), loaddata(), themnhanvien(), xoanhanvien(), capnhat() có nhiệm vụ thực hiện các tác vụ như xóa trường dữ liệu, kích hoạt hoặc vô hiệu hóa các nút và trường nhập liệu, tải dữ liệu từ cơ sở dữ liệu, thực hiện các thao tác thêm, xóa, cập nhật dữ liệu và tìm kiếm nhân viên.
- Hàm Form_DanhSachNhanVien() là hàm khởi tạo của lớp.
- Hàm back() được sử dụng để xóa dữ liệu đã nhập và đặt trỏ chuột vào trường nhập liệu txtTennhanvien.
- Hàm enablecommand(bool kt) được sử dụng để kích hoạt hoặc vô hiệu hóa các nút điều khiển trên form.
- Hàm binding() được sử dụng để gắn kết dữ liệu từ DataGridView lên các trường nhập liệu tương ứng.
- Hàm addcbbVitri() được sử dụng để tải danh sách các vị trí nhân viên từ cơ sở dữ liệu và gắn kết lên combobox cbbVitri.
- Hàm loaddata() được sử dụng để tải danh sách nhân viên từ cơ sở dữ liệu và hiển thị lên DataGridView grvNhanvien.
- Hàm themnhanvien() kiểm tra và thêm một nhân viên mới vào cơ sở dữ liệu.

- Hàm xoanhanvien() xóa một nhân viên khỏi cơ sở dữ liệu.
- Hàm capnhat() cập nhật thông tin của một nhân viên trong cơ sở dữ liệu.
- Các sự kiện btnThem_Click(), btnXoa_Click(), btnXacnhan_Click(), btnCapnhat_Click(), Form_DanhSachNhanVien_Load(), btnTim_Click() được gắn kết với các nút và điều khiển trên form để thực hiện các tác vụ tương ứng khi người dùng tương tác với chúng.

2.1.7 Source code form danh sách phòng

public partial class Form_DanhSachPhong : Form

```
{
    private ConnectDatabase conn = new ConnectDatabase();
    private DataSet ds = new DataSet();
    private int kt;
    public Form_DanhSachPhong()
    {
        InitializeComponent();
    }
    private void back()
    {
        txtTenphong.Text = string.Empty;
        txtMaphong.Text = string.Empty;
        ckbDangthue.Checked = false;
        txtMaphong.Focus();
    }

    private void enablecommand(bool kt)
    {
        btnLuu.Enabled = !kt;
        btnCapnhat.Enabled = kt;
        btnThem.Enabled = kt;
        btnXoa.Enabled = kt;
        grvPhong.ReadOnly = kt;
    }
}
```

```

        //Form_LoaiPhong.ac
    }
    private void addcbbLoaiphong()
    {
        DataSet ds = new DataSet();
        SqlCommand cmd = new SqlCommand();
        cmd.CommandText = "Select ID, TenLoaiPhong from tblLoaiPhong";
        cmd.Connection = conn.connect();
        SqlDataAdapter da = new SqlDataAdapter();
        da.SelectCommand = cmd;
        da.Fill(ds, "LoaiPhong");
        cbbLoaiphong.DataSource = ds.Tables["LoaiPhong"];
        cbbLoaiphong.DisplayMember = "TenLoaiPhong";
        cbbLoaiphong.ValueMember = "ID";
    }

    private void loaddata()
    {
        var con = conn.connect();

        string sql = "Select tblPhong.ID as 'Mã Phòng', TenLoaiPhong as 'Tên Loại  
Phòng', TenPhong as 'Tên Phòng', " +
            " TrangThai as 'Trạng Thái'" +
            "FROM tblPhong, tblLoaiPhong where IDLoaiPhong = '" +
        cbbLoaiphong.SelectedValue.ToString() + "' and tblPhong.IDLoaiPhong =  
tblLoaiPhong.ID";

        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.CommandType = CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(dt);
    }

```

```

        con.Close();

        grvPhong.DataSource = dt;
    }

    private void loadalldata()
    {
        var con = conn.connect();

        string sql = "Select tblPhong.ID as 'Mã Phòng', TenLoaiPhong as 'Tên Loại Phòng', TenPhong as 'Tên Phòng', TrangThai as 'Trạng Thái' FROM tblPhong, tblLoaiPhong Where tblPhong.IDLoaiPhong = tblLoaiPhong.ID";

        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.CommandType = CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(dt);
        con.Close();
        grvPhong.DataSource = dt;

    }

    private void Form_DanhSachPhong_Load(object sender, EventArgs e)
    {
        addcbbLoaiphong();
        loadalldata();

    }

    private void cbbLoaiphong_SelectedIndexChanged(object sender, EventArgs e)
    {
        loaddata();
    }

```

```

private void binding()
{
    Int32 i = Convert.ToInt32(grvPhong.CurrentRow.Index);
    txtTenphong.DataBindings.Clear();
    txtTenphong.DataBindings.Add("Text", grvPhong.DataSource, "Tên Phòng");
    txtMaphong.DataBindings.Clear();
    txtMaphong.DataBindings.Add("Text", grvPhong.DataSource, "Mã Phòng");
    if (grvPhong.Rows[i].Cells["Trạng Thái"].Value.ToString() == "Đang thuê")
    {
        ckbDangthue.Checked = true;
    }
    else
    {
        ckbDangthue.Checked = false;
    }
}

private bool themphong()
{
    if (string.IsNullOrEmpty(txtTenphong.Text))
    {
        MessageBox.Show("Vui lòng nhập tên phòng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return false;
    }
    if (cbbLoaiphong.SelectedIndex < 0)
    {

```

```

        MessageBox.Show("Vui lòng chọn loại phòng!", "Chú ý!",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);

        return false;
    }

    var con = conn.connect();

    string check;

    if (ckbDangthue.Checked)
    {
        check = "Đang thuê";
    }

    else check = "Còn trống";

    //MessageBox.Show(cbbLoaiphong.SelectedValue.ToString());

    string Check_exist_query = "SELECT ID FROM tblPhong WHERE ID = " +
    txtMaphong.Text;

    string Insert_query = "insert into tblPhong values" +

        "(" + int.Parse(txtMaphong.Text) + ", N" + txtTenphong.Text + ", " +
    cbbLoaiphong.SelectedValue.ToString() + ", N" + check + ")";

    SqlCommand cmd = new SqlCommand(Check_exist_query, con);

    SqlDataReader dr = cmd.ExecuteReader();

    if (!dr.HasRows)
    {
        dr.Close();

        cmd = new SqlCommand(Insert_query, con);

        if (cmd.ExecuteNonQuery() > 0)
        {
            con.Close();

            return true;
        }
    }

    con.Close();

```

```
        MessageBox.Show("ID đã tồn tại", "Ràng buộc dữ liệu",  
        MessageBoxButtons.OK, MessageBoxIcon.Error);  
        return false;  
    }
```

```
private void btnThem_Click(object sender, EventArgs e)  
{  
    txtMaphong.ReadOnly = true;  
    back();  
    Int32 i;  
    if (grvPhong.Rows.Count >= 1)  
    {  
        i = 1;  
    }  
    else  
    {  
        i = Convert.ToInt32(grvPhong.CurrentRow.Index);  
    }  
    int id = int.Parse(grvPhong.RowCount.ToString());  
    txtMaphong.Text = id.ToString();  
    txtTenphong.Focus();  
    enablecommand(false);  
    kt = 1;  
}
```

```
private bool xoaphong()  
{  
    if (string.IsNullOrEmpty(txtMaphong.Text))  
    {
```

```

        MessageBox.Show("Vui lòng nhập mã phòng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return false;
    }

    var con = conn.connect();

    string Check_exist_query = "SELECT * FROM tblPhong WHERE ID = " +
txtMaphong.Text;

    string Delete_query = "Delete from tblPhong where ID = " + txtMaphong.Text +
" and TrangThai = N'Còn trống'";

    SqlCommand cmd = new SqlCommand(Check_exist_query, con);
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        dr.Close();

        cmd = new SqlCommand(Delete_query, con);
        if (cmd.ExecuteNonQuery() > 0)
        {
            con.Close();

            return true;
        }
    }

    con.Close();

    MessageBox.Show("ID Không tồn tại hoặc phòng đang được thuê", "Ràng buộc
dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Error);

    return false;
}

```

```

private void btnXoa_Click(object sender, EventArgs e)
{
    binding();
}

```



```

        if (MessageBox.Show("Bạn thực sự muốn xóa phòng có mã '" +
            txtMaphong.Text + "' không???", "Warning", MessageBoxButtons.YesNo,
            MessageBoxIcon.Question) == DialogResult.Yes)
        {
            if (xoaphong())
            {
                MessageBox.Show("Xóa phòng thành công", "Ràng buộc dữ liệu",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
            }
            loaddata();
            back();
        }

```

```

private bool capnhat()
{
    string check;
    if (ckbDangthue.Checked)
    {
        check = "Đang thuê";
    }
    else
    {
        check = "Còn trống";
    }
    var con = conn.connect();
    string Update_query = "UPDATE tblPhong SET ID = '" + txtMaphong.Text +
        "', TenPhong = N'" + txtTenphong.Text +
        "', TrangThai = N'" + check + "'" +
        "WHERE ID = " + txtMaphong.Text;
    SqlCommand cmd = new SqlCommand(Update_query, con);
    if (cmd.ExecuteNonQuery() > 0)

```

```

    {
        con.Close();
        return true;
    }
    con.Close();
    return false;
}

private void btnCapnhat_Click(object sender, EventArgs e)
{
    binding();
    cbbLoaiphong.Enabled = false;
    grvPhong.Enabled = false;
    txtMaphong.ReadOnly = true;
    enablecommand(false);
    kt = 2;
}

private void btnLuu_Click(object sender, EventArgs e)
{
    if (kt == 1)
    {
        if (themphong())
        {
            MessageBox.Show("Thêm phòng thành công", "Ràng buộc dữ liệu",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            enablecommand(true);
            txtMaphong.ReadOnly = false;
            loadalldata();
            back();
        }
    }
}

```

```

    }

    else if (MessageBox.Show("Bạn muốn dừng thêm???", "Warning",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

        enablecommand(true);

    }

    else if (kt == 2)

    {

        if (capnhat())

        {

            MessageBox.Show("Cập nhật phòng thành công", "Ràng buộc dữ liệu",
                MessageBoxButtons.OK, MessageBoxIcon.Information);

            enablecommand(true);

            txtMaphong.ReadOnly = false;

            loadalldata();

            cbbLoaiphong.Enabled = true;

            grvPhong.Enabled = true;

            back();

        }

    }

}

private void grvPhong_CellClick(object sender, DataGridViewCellEventArgs e)

{

    back();

}

private void btnTim_Click(object sender, EventArgs e)

{

    if (string.IsNullOrEmpty(txtMaphong.Text) &&
        string.IsNullOrEmpty(txtTenphong.Text))

    {

```

```

        MessageBox.Show("Vui lòng nhập mã phòng hoặc tên phòng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    var con = conn.connect();

    string sql = "Select tblPhong.ID as 'Mã Phòng', TenLoaiPhong as 'Tên Loại Phòng', TenPhong as 'Tên Phòng', " +
        "IDLoaiPhong as 'Mã Loại Phòng', TrangThai as 'Trạng Thái'" +
        "FROM tblPhong, tblLoaiPhong " +
        "where (tblPhong.ID = '" + txtMaphong.Text + "' or TenPhong = N'" + txtTenphong.Text + "') and IDLoaiPhong = tblLoaiPhong.ID";

    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.CommandType = CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    con.Close();
    grvPhong.DataSource = dt;
}

private void panel1_Paint(object sender, PaintEventArgs e)
{

}

private void btnHientatcaphong_Click(object sender, EventArgs e)
{
    loadalldata();
}

```

```
}
}
```

- Lớp Form_DanhSachPhong quản lý danh sách phòng trong ứng dụng quản lý khách sạn. Dưới đây là các chức năng và phương thức chính trong lớp này:
- Các biến và đối tượng được khai báo: conn là một đối tượng ConnectDatabase để kết nối với cơ sở dữ liệu, ds là một đối tượng DataSet để lưu trữ dữ liệu, kt là một biến kiểu int để xác định trạng thái (1: thêm phòng, 2: cập nhật thông tin phòng).
- Phương thức back(), enablecommand(bool kt), addcbbLoaiphong(), loaddata(), loadalldata(), binding(), themphong(), xoaphong(), capnhat() có nhiệm vụ thực hiện các tác vụ như xóa trường dữ liệu, kích hoạt hoặc vô hiệu hóa các nút và điều khiển, tải dữ liệu từ cơ sở dữ liệu, thực hiện các thao tác thêm, xóa, cập nhật dữ liệu và tìm kiếm phòng.
- Hàm Form_DanhSachPhong() là hàm khởi tạo của lớp.
- Hàm back() được sử dụng để xóa dữ liệu đã nhập và đặt trỏ chuột vào trường nhập liệu txtMaphong.
- Hàm enablecommand(bool kt) được sử dụng để kích hoạt hoặc vô hiệu hóa các nút điều khiển trên form.
- Hàm addcbbLoaiphong() được sử dụng để tải danh sách các loại phòng từ cơ sở dữ liệu và gắn kết lên combobox cbbLoaiphong.
- Hàm loaddata() được sử dụng để tải danh sách phòng từ cơ sở dữ liệu dựa trên loại phòng đã chọn và hiển thị lên DataGridView grvPhong.
- Hàm loadalldata() được sử dụng để tải tất cả danh sách phòng từ cơ sở dữ liệu và hiển thị lên DataGridView grvPhong.
- Hàm binding() được sử dụng để gắn kết dữ liệu từ DataGridView lên các trường nhập liệu tương ứng.
- Hàm themphong() kiểm tra và thêm một phòng mới vào cơ sở dữ liệu.
- Hàm xoaphong() xóa một phòng khỏi cơ sở dữ liệu.
- Hàm capnhat() cập nhật thông tin của một phòng trong cơ sở dữ liệu.
- Các sự kiện Form_DanhSachPhong_Load(), cbbLoaiphong_SelectedIndexChanged(), btnThem_Click(), btnXoa_Click(), btnCapnhat_Click(), btnLuu_Click(), grvPhong_CellClick(), btnTim_Click(), btnHientatcaphong_Click() được gắn kết với các nút và điều khiển trên form để thực hiện các tác vụ tương ứng khi người dùng tương tác với chúng.

2.1.8 Source code form danh sách thuê phòng

```
public partial class Form_DanhSachThuePhong : Form
```

```

{
    private ConnectDatabase conn = new ConnectDatabase();
    private int kt;
    int current_price;
    int single_price;
    public Form_DanhSachThuePhong()
    {
        InitializeComponent();
    }

    private void back()
    {
        txtID.Text = string.Empty;
        cbbKhachhang.Text = string.Empty;
        cbbPhong.Text = string.Empty;
        txtThoigianthue.Text = string.Empty;
        dtpNgaythue.Text = string.Empty;
        txtID.Focus();
    }

    private void enablecommand(bool kt)
    {
        btnLuu.Enabled = !kt;
        btnThem.Enabled = kt;
        grvThuephong.ReadOnly = kt;
        btnGiahan.Enabled = kt;
    }

    private void addcbbKhachhang()

```

```

{
    DataSet dataSet = new DataSet();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "Select HoTen, CCCD from tblKhachHang";
    cmd.Connection = conn.connect();
    SqlDataAdapter da = new SqlDataAdapter();
    da.SelectCommand = cmd;
    da.Fill(dataSet, "KhachHang");
    cbbKhachhang.DataSource = dataSet.Tables["KhachHang"];
    cbbKhachhang.DisplayMember = "HoTen";
    cbbKhachhang.ValueMember = "CCCD";
}

private void addcbbPhong()
{
    DataSet dataSet = new DataSet();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "Select tblPhong.ID, TenPhong, DonGia, IDLoaiPhong,
TrangThai, " +
        "concat_ws(' - ', IDLoaiPhong, TenPhong, FORMAT(DonGia, '#,##.')) as
'Phòng' from tblPhong, tblLoaiPhong where tblLoaiPhong.ID = tblPhong.IDLoaiPhong
and tblPhong.TrangThai = N'Còn trống";
    cmd.Connection = conn.connect();
    SqlDataAdapter da = new SqlDataAdapter();
    da.SelectCommand = cmd;
    da.Fill(dataSet, "Phong");
    cbbPhong.DataSource = dataSet.Tables["Phong"];
    cbbPhong.DisplayMember = "Phòng";
    cbbPhong.ValueMember = "ID";
}

```

```

private void loaddata()
{
    var con = conn.connect();

    string sql = "select tblThuePhong.ID as 'Mã', TenLoaiPhong as 'Tên Loại Phòng',
TenPhong as 'Tên Phòng', HoTen as 'Họ và Tên', " +
        "tblKhachHang.CCCD as 'CCCD/CMND', DienThoai as 'Số Điện Thoại',
GioThue as 'Giờ Thuê', " +
        "GioTra as 'Giờ Trả', DonGia as 'Đơn Giá', GiaPhong as 'Cần Thanh Toán',
tblThuePhong.TrangThai as 'Trạng Thái' " +
        "from tblLoaiPhong, tblPhong, tblKhachHang, tblThuePhong " +
        "where tblLoaiPhong.ID = tblPhong.IDLoaiPhong and tblPhong.ID =
tblThuePhong.IDPhong " +
        "and tblThuePhong.CCCD = tblKhachHang.CCCD " +
        "order by tblThuePhong.ID DESC";
SqlCommand cmd = new SqlCommand(sql, con);
cmd.CommandType = CommandType.Text;
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
con.Close();
grvThuephong.DataSource = dt;
}

```

```

private bool themthuephong()
{
    if (string.IsNullOrEmpty(txtID.Text))
    {
        MessageBox.Show("Vui lòng nhập ID thuê phòng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}

```



```

        return false;
    }
    string check = "";
    if (ckbTrangthai.Checked)
    {
        check = "Đã thanh toán";
    }
    else
    {
        check = "Chưa thanh toán";
    }
    string[] text = cbbPhong.Text.Split(" - ");
    int price = int.Parse(text[text.Length - 1].Replace(",", ""));
    //TimeSpan timeSpan = dtpNgaytra.Value-dtpNgaythue.Value;
    if (txtThoigianthue.Text == "")
    {
        MessageBox.Show("Vui lòng nhập thời gian thuê!", "Cảnh Báo",
        MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return false;
    }
    var con = conn.connect();

    string Check_exist_query = "SELECT tblThuePhong.ID, tblPhong.TrangThai
    FROM tblThuePhong, tblPhong WHERE tblThuePhong.ID = " + txtID.Text + "and
    tblPhong.TrangThai = N'Đang Thuê'";

    string Insert_query = "insert into tblThuePhong values" +
        "(" + int.Parse(txtID.Text) + ", " + cbbKhachhang.SelectedValue.ToString() +
        ", " + int.Parse(cbbPhong.SelectedValue.ToString()) + ", " +
        int.Parse(txtThoigianthue.Text) * price + ", " +
        "" + dtpNgaythue.Value.ToString("MM/dd/yyyy HH:mm") + ", " +
        dtpNgaythue.Value.AddHours(int.Parse(txtThoigianthue.Text)).ToString("MM/dd/yyyy
        HH:mm") + ", N" + check + ")";

```

```

SqlCommand cmd = new SqlCommand(Check_exist_query, con);
SqlDataReader dr = cmd.ExecuteReader();
if (!dr.HasRows)
{
    dr.Close();
    cmd = new SqlCommand(Insert_query, con);
    if (cmd.ExecuteNonQuery() > 0)
    {
        con.Close();
        return true;
    }
}
con.Close();

MessageBox.Show("ID đã tồn tại hoặc phòng đang có người thuê!", "Lỗi",
MessageBoxButtons.OK, MessageBoxIcon.Error);

return false;
}

private void chuyentrangthaiphong()
{
    var con = conn.connect();
    string check = "Còn trống";
    string check_query1 = "select IDPhong from tblThuePhong where IDPhong = " +
cbbPhong.SelectedValue;
    SqlCommand cmd = new SqlCommand(check_query1, con);
    SqlDataReader dr = cmd.ExecuteReader();
    if (dr.HasRows)
    {
        dr.Close();
    }
}

```

```

        check = "Đang thuê";

        string update_query = "update tblPhong set tblPhong.TrangThai = N'" + check
+ "' from tblPhong, tblThuePhong where tblThuePhong.IDPhong = '" +
cbbPhong.SelectedValue + "' and tblPhong.ID = tblThuePhong.IDPhong";

        SqlCommand cmd1 = new SqlCommand(update_query, con);
        cmd1.ExecuteNonQuery();
    }
}

```

```

private void btnThem_Click(object sender, EventArgs e)
{
    back();
    if (grvThuephong.RowCount == 1)
    {
        txtID.Text = "1";
    }
    else
    {
        txtID.Text = (int.Parse(grvThuephong.Rows[0].Cells[0].Value.ToString()) +
1).ToString();
    }
    addcbbKhachhang();
    addcbbPhong();

    enablecommand(false);
    kt = 1;
    txtThoigianthue.Focus();
}

```

```

private void btnTim_Click(object sender, EventArgs e)

```

```

{
    txtID.ReadOnly = false;

    if (string.IsNullOrEmpty(txtID.Text) &&
        string.IsNullOrEmpty(cbbKhachhang.Text))
    {
        MessageBox.Show("Vui lòng nhập mã thuê phòng hoặc tên khách hàng!",
            "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Warning);
        return;
    }

    var con = conn.connect();

    string find_sql = "select tblThuePhong.ID as 'Mã', TenLoaiPhong as 'Tên Loại Phòng', TenPhong as 'Tên Phòng', HoTen as 'Họ và Tên', " +
        "tblKhachHang.CCCD as 'CCCD/CMND', DienThoai as 'Số Điện Thoại', Format(cast(GioThue as datetime), 'M/dd/yyyy HH:mm') as 'Giờ Thuê', " +
        "Format(cast(GioTra as datetime), 'M/dd/yyyy HH:mm') as 'Giờ Trả', GiaPhong as 'Cần Thanh Toán', tblThuePhong.TrangThai as 'Trạng Thái' " +
        "from tblLoaiPhong, tblPhong, tblKhachHang, tblThuePhong " +
        "where tblLoaiPhong.ID = tblPhong.IDLoaiPhong and tblPhong.ID = tblThuePhong.IDPhong " +
        "and tblThuePhong.CCCD = tblKhachHang.CCCD and (tblThuePhong.ID = " +
        + txtID.Text + " or tblKhachHang.HoTen = N'" + cbbKhachhang.Text + "'";

    SqlCommand cmd = new SqlCommand(find_sql, con);
    cmd.CommandType = CommandType.Text;

    DataTable dt = new DataTable();

    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);

    con.Close();

    grvThuephong.DataSource = dt;
}

```

```

private void btnLuu_Click(object sender, EventArgs e)
{
    if (kt == 1)
    {
        if (themthuephong())
        {
            MessageBox.Show("Thêm tác vụ thuê phòng thành công", "Ràng buộc dữ
            liệu", MessageBoxButtons.OK, MessageBoxIcon.Information);

            enablecommand(true);

            chuyentrangthaiphong();

            addcbbPhong();

            loaddata();

            back();
        }

        else if (MessageBox.Show("Bạn muốn dừng thêm???", "Warning",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)

            enablecommand(true);
    }

    else if (kt == 2)
    {
        if (giahanphong())
        {
            MessageBox.Show("Gia hạn tác vụ thuê phòng thành công", "Ràng buộc dữ
            liệu", MessageBoxButtons.OK, MessageBoxIcon.Information);

            enablecommand(true);

            txtID.ReadOnly = false;

            back();
        }
    }
}

```

```

private void Format_Gridview()
{
    grvThuephong.Columns["Mã"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    grvThuephong.Columns["Mã"].Width = 30;

    grvThuephong.Columns["Đơn Giá"].DefaultCellStyle.Format = "N0";

    grvThuephong.Columns["Đơn Giá"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    grvThuephong.Columns["Cần Thanh Toán"].DefaultCellStyle.Format = "N0";

    grvThuephong.Columns["Cần Thanh Toán"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    grvThuephong.Columns["CCCD/CMND"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    grvThuephong.Columns["Giờ Thuê"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    grvThuephong.Columns["Giờ Trả"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    grvThuephong.Columns["Số Điện Thoại"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;

    grvThuephong.Columns["Giờ Thuê"].DefaultCellStyle.Format = "M/dd/yyyy
HH:mm tt";

    grvThuephong.Columns["Giờ Trả"].DefaultCellStyle.Format = "M/dd/yyyy
HH:mm tt";
}

```

```

private void Form_DanhSachThuePhong_Load(object sender, EventArgs e)
{
    grvThuephong.Enabled = true;

    loaddata();

    Format_Gridview();

    dtpNgaythue.Value = DateTime.Now;
}

```

```

    }

    private void button1_Click(object sender, EventArgs e)
    {
        var f = new Form_KhachHang_Dialog();
        f.ShowDialog();
    }

    private void binding()
    {
        Int32 i = Convert.ToInt32(grvThuephong.CurrentRow.Index);
        txtID.DataBindings.Clear();
        txtID.DataBindings.Add("Text", grvThuephong.DataSource, "Mã");
        cbbKhachhang.DataBindings.Clear();
        //cbbKhachhang.DataBindings.Add("DisplayMember",
        grvThuephong.DataSource, "Họ và Tên");

        cbbKhachhang.Text = grvThuephong.Rows[i].Cells["Họ và
        Tên"].Value.ToString();
        cbbPhong.DataBindings.Clear();
        cbbPhong.Text = grvThuephong.Rows[i].Cells["Tên Phòng"].Value.ToString() +
        " - " + grvThuephong.Rows[i].Cells["Tên Loại Phòng"].Value.ToString();
        dtpNgaythue.DataBindings.Clear();
        dtpNgaythue.Value = DateTime.Now;
        if (grvThuephong.Rows[i].Cells["Trạng Thái"].Value.ToString() == "Đã thanh
        toán")
        {
            ckbTrangthai.Checked = true;
        }
        else
    }

```

```

    {
        ckbTrangthai.Checked = false;
    }
}

private bool checkthanhtoan()
{
    current_price = 0;
    single_price = 0;
    Int32 i = Convert.ToInt32(grvThuephong.CurrentRow.Index);
    single_price += int.Parse(grvThuephong.Rows[i].Cells["Đơn
Giá"].Value.ToString());
    if (grvThuephong.Rows[i].Cells["Trạng Thái"].Value.ToString() == "Chưa thanh
toán")
    {
        MessageBox.Show("Chưa thanh toán đợt vừa rồi!!!", "Cảnh Báo",
        MessageBoxButtons.OK, MessageBoxIcon.Error);
        current_price += int.Parse(grvThuephong.Rows[i].Cells["Cần Thanh
Toán"].Value.ToString());
        return true;
    }
    return false;
}

private bool giahaphong()
{
    var con = conn.connect();
    if (checkthanhtoan())
    {

```



```

        int tt = current_price + (int.Parse(txtThoigianthue.Text) * single_price);

        string update_query = "update tblThuePhong set GioThue = " +
dtpNgaythue.Value.ToString() + ", GioTra = " +
dtpNgaythue.Value.AddHours(int.Parse(txtThoigianthue.Text)).ToString() + " +
        ", GiaPhong = " + tt +
        " where ID = " + txtID.Text + """;

SqlCommand cmd = new SqlCommand(update_query, con);
if (cmd.ExecuteNonQuery() > 0)
{
    con.Close();
    Int32 i = Convert.ToInt32(grvThuephong.CurrentRow.Index);
    loaddata();
    grvThuephong.Rows[i].Selected = true;
    return true;
}
}
return false;
}

```

```

private void btnGiahan_Click(object sender, EventArgs e)
{
    txtID.ReadOnly = true;

    enablecommand(false);
    binding();
    kt = 2;
}

```

- Lớp Form_DanhSachThuePhong quản lý danh sách các tác vụ thuê phòng trong ứng dụng quản lý khách sạn. Dưới đây là các chức năng và phương thức chính trong lớp này:

- Các biến và đối tượng được khai báo: conn là một đối tượng ConnectDatabase để kết nối với cơ sở dữ liệu, kt là một biến kiểu int để xác định trạng thái (1: thêm tác vụ thuê phòng, 2: gia hạn tác vụ thuê phòng), current_price và single_price là các biến để lưu giá tiền hiện tại và giá tiền của một đơn vị thời gian thuê phòng.
- Phương thức back(), enablecommand(bool kt), addcbbKhachhang(), addcbbPhong(), loaddata(), themthuephong(), chuyentrangthaiphong(), giahanphong(), checkthanhtoan(), Format_Gridview(), binding() có nhiệm vụ thực hiện các tác vụ như xóa trường dữ liệu, kích hoạt hoặc vô hiệu hóa các nút và điều khiển, tải dữ liệu từ cơ sở dữ liệu, thực hiện các thao tác thêm, gia hạn, kiểm tra thanh toán tác vụ thuê phòng, định dạng hiển thị dữ liệu trên DataGridView và gắn kết dữ liệu lên các trường nhập liệu tương ứng.
- Hàm Form_DanhSachThuePhong() là hàm khởi tạo của lớp.
- Hàm back() được sử dụng để xóa dữ liệu đã nhập và đặt trỏ chuột vào trường nhập liệu txtID.
- Hàm enablecommand(bool kt) được sử dụng để kích hoạt hoặc vô hiệu hóa các nút điều khiển trên form.
- Hàm addcbbKhachhang() và addcbbPhong() được sử dụng để tải danh sách khách hàng và danh sách phòng từ cơ sở dữ liệu và gắn kết lên combobox tương ứng.
- Hàm loaddata() được sử dụng để tải danh sách tác vụ thuê phòng từ cơ sở dữ liệu và hiển thị lên DataGridView grvThuephong.
- Hàm themthuephong() kiểm tra và thêm một tác vụ thuê phòng mới vào cơ sở dữ liệu.
- Hàm chuyentrangthaiphong() chuyển trạng thái của phòng đã được thuê sang "Đang thuê" khi có một tác vụ thuê phòng mới được thêm.
- Hàm giahanphong() gia hạn tác vụ thuê phòng hiện tại.
- Hàm checkthanhtoan() kiểm tra xem tác vụ thuê phòng hiện tại đã được thanh toán hay chưa.
- Hàm Format_Gridview() được sử dụng để định dạng hiển thị dữ liệu trên DataGridView grvThuephong.
- Các sự kiện Form_DanhSachThuePhong_Load(), btnThem_Click(), btnTim_Click(), btnLuu_Click(), button1_Click(), btnGiahan_Click() được gắn kết với các nút và điều khiển trên form để thực hiện các tác vụ tương ứng khi người dùng tương tác với chúng.

2.1.9 Source code form gia hạn phòng

```
public partial class Form_GiaHanPhong : Form
{
    private ConnectDatabase conn = new ConnectDatabase();
    private DataSet ds = new DataSet();
    private int kt;
    public Form_GiaHanPhong()
    {
        InitializeComponent();
    }

    private void Form_GiaHanPhong_Load(object sender, EventArgs e)
    {
    }
}
```

- Lớp Form_GiaHanPhong là một form để thực hiện gia hạn tác vụ thuê phòng trong ứng dụng quản lý khách sạn. Dưới đây là mô tả ngắn gọn về lớp này:
- Các biến và đối tượng được khai báo: conn là một đối tượng ConnectDatabase để kết nối với cơ sở dữ liệu, ds là một đối tượng DataSet để lưu trữ dữ liệu, kt là một biến kiểu int để xác định trạng thái.
- Hàm khởi tạo Form_GiaHanPhong().
- Sự kiện Form_GiaHanPhong_Load().

2.1.10 Source code form hoá đơn

```
public partial class Form_HoaDon : Form
{
    private ConnectDatabase conn = new ConnectDatabase();
    public Form_HoaDon()
    {
        InitializeComponent();
    }

    private void loadalldata()
    {
        var con = conn.connect();
        string sql = "select tblHoaDon.ID as 'Mã', TenLoaiPhong as 'Tên Loại Phòng',
        TenPhong as 'Tên Phòng', HoTen as 'Họ và Tên', " +
        "tblKhachHang.CCCD as 'CCCD/CMND', DienThoai as 'Số Điện Thoại',
        GioThue as 'Giờ Thuê', " +
```

```

        "GioTra as 'Giờ Trả', DonGia as 'Đơn Giá', GiaPhong as 'Cần Thanh Toán',
tblHoaDon.TrangThai as 'Trạng Thái' " +
        "from tblLoaiPhong, tblPhong, tblKhachHang, tblHoaDon " +
        "where tblLoaiPhong.ID = tblPhong.IDLoaiPhong and tblPhong.ID =
tblHoaDon.IDPhong " +
        "and tblHoaDon.CCCD_KH = tblKhachHang.CCCD " +
        "order by tblHoaDon.ID DESC";
SqlCommand cmd = new SqlCommand(sql, con);
cmd.CommandType = CommandType.Text;
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
con.Close();
grvHoaDon.DataSource = dt;
}

private void loadadata()
{
    var con = conn.connect();
    string sql = "select tblHoaDon.ID as 'Mã', TenLoaiPhong as 'Tên Loại Phòng',
TenPhong as 'Tên Phòng', HoTen as 'Họ và Tên', " +
        "tblKhachHang.CCCD as 'CCCD/CMND', DienThoai as 'Số Điện Thoại',
Convert(nvarchar, GioThue, 101) as 'Giờ Thuê', " +
        "Convert(nvarchar, GioTra, 101) as 'Giờ Trả', CONVERT(nvarchar,
tblHoaDon.GioThue, 101) as 'Ngày', DonGia as 'Đơn Giá', GiaPhong as 'Cần Thanh
Toán', tblHoaDon.TrangThai as 'Trạng Thái' " +
        "from tblLoaiPhong, tblPhong, tblKhachHang, tblHoaDon " +
        "where tblLoaiPhong.ID = tblPhong.IDLoaiPhong and tblPhong.ID =
tblHoaDon.IDPhong and tblHoaDon.CCCD_KH = tblKhachHang.CCCD " +
        "and tblHoaDon.CCCD_KH = tblKhachHang.CCCD " +
        "and SUBSTRING(GioThue, 1, 10) >= '" +
dtpTuNgay.Value.ToString("MM/dd/yyyy") + "' " +
        "and SUBSTRING(GioTra, 1, 10) <= '" +
dtpToingay.Value.ToString("MM/dd/yyyy") + "' " +
        "order by tblHoaDon.ID DESC";
SqlCommand cmd = new SqlCommand(sql, con);
cmd.CommandType = CommandType.Text;
DataTable dt = new DataTable();
SqlDataAdapter da = new SqlDataAdapter(cmd);
da.Fill(dt);
con.Close();
grvHoaDon.DataSource = dt;
}

private ulong tinh tongdoanhthu()
{
    ulong s = 0;

```

```

        for (int i = 0; i < grvHoaDon.RowCount - 1; i++)
        {
            //MessageBox.Show(dtpTuNgay.Value.ToString().Split(" ")[0] +
            "\n"+grvHoaDon.Rows[i].Cells["Giờ Thuê"].Value.ToString().Split(" ")[0]);
            s += ulong.Parse(grvHoaDon.Rows[i].Cells["Cần Thanh
Toán"].Value.ToString());
        }
        return s;
    }

    private void Form_HoaDon_Load(object sender, EventArgs e)
    {

        loadalldata();
        lbTongdoanhthu.Text = string.Format("{0:0,00}", tinhtongdoanhthu());
        grvHoaDon.Columns["Cần Thanh Toán"].DefaultCellStyle.Format = "N0";
        grvHoaDon.Columns["Cần Thanh Toán"].DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
    }

    private void btnLoc_Click(object sender, EventArgs e)
    {
        loadadata();
        lbTongdoanhthu.Text = string.Format("{0:0,00}", tinhtongdoanhthu());
    }

    private void Form_HoaDon_Click(object sender, EventArgs e)
    {
        lbTongdoanhthu.Text = string.Format("{0:0,00}", tinhtongdoanhthu());
        loadalldata();
    }
}

```

- Lớp Form_HoaDon là một form trong ứng dụng quản lý khách sạn được tạo để hiển thị thông tin về hóa đơn. Trong lớp này, có các biến và đối tượng sau:
- conn: Đối tượng của lớp ConnectDatabase để kết nối với cơ sở dữ liệu.
- Lớp này bao gồm các phương thức và sự kiện sau:
- loadalldata(): Phương thức để tải tất cả dữ liệu hóa đơn từ cơ sở dữ liệu và hiển thị lên GridView grvHoaDon.
- loadadata(): Phương thức để tải dữ liệu hóa đơn từ cơ sở dữ liệu theo khoảng thời gian đã chọn từ dtpTuNgay đến dtpToingay và hiển thị lên GridView grvHoaDon.
- tinhtongdoanhthu(): Phương thức để tính tổng doanh thu từ các hóa đơn trong GridView grvHoaDon.

- Form_HoaDon_Load(): Sự kiện khi form được tải lên, gọi phương thức loadalldata() để hiển thị tất cả dữ liệu hóa đơn và tính tổng doanh thu.
- btnLoc_Click(): Sự kiện khi nhấn nút "Lọc", gọi phương thức loadadata() để tải dữ liệu hóa đơn theo khoảng thời gian đã chọn và tính tổng doanh thu.
- Form_HoaDon_Click(): Sự kiện khi click vào form, hiển thị lại tổng doanh thu và tải lại tất cả dữ liệu hóa đơn.

2.1.11 Source code form tài khoản

public partial class Form_TaiKhoan : Form

```
{
    private ConnectDatabase conn = new ConnectDatabase();
    private int kt;

    private void back()
    {
        txtTendangnhap.Text = string.Empty;
        txtMatkhau.Text = string.Empty;
        cbbVitri.Text = string.Empty;
        txtTendangnhap.Focus();
    }

    private void enablecommand(bool kt)
    {
        btnXacnhan.Enabled = !kt;
        btnThem.Enabled = kt;
        btnCapnhat.Enabled = kt;
        btnXoa.Enabled = kt;
    }

    private void binding()
    {
        Int32 i = Convert.ToInt32(grvTaikhoan.CurrentRow.Index);
        txtTendangnhap.DataBindings.Clear();
        txtTendangnhap.DataBindings.Add("Text", grvTaikhoan.DataSource, "Tên Đăng
        Nhập");

        cbbVitri.Text = grvTaikhoan.Rows[i].Cells["Vị Trí"].Value.ToString();

        txtMatkhau.DataBindings.Clear();
        txtMatkhau.DataBindings.Add("Text", grvTaikhoan.DataSource, "Mật Khẩu");
        txtIDtaikhoan.DataBindings.Clear();
        txtIDtaikhoan.DataBindings.Add("Text", grvTaikhoan.DataSource, "Mã Tài
        Khoản");
    }
}
```

```

private void addcbbVitri()
{
    DataSet dataSet = new DataSet();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "Select * from tblViTri";
    cmd.Connection = conn.connect();
    SqlDataAdapter da = new SqlDataAdapter();
    da.SelectCommand = cmd;
    da.Fill(dataSet, "ViTri");
    cbbVitri.DataSource = dataSet.Tables["ViTri"];
    cbbVitri.DisplayMember = "TenViTri";
    cbbVitri.ValueMember = "ID";
}

private void loaddata()
{
    var con = conn.connect();
    string sql = "select tblTaiKhoan.ID as 'Mã Tài Khoản',TenNhanVien as 'Tên Nhân  

Viên', TaiKhoan as 'Tên Đăng Nhập', " +
        "CONVERT(VARCHAR(10), HashBytes('MD5', MatKhau)) as 'Mật Khẩu',  

TenViTri as 'Vị Trí' " +
        "from tblTaiKhoan, tblViTri, tblNhanVien " +
        "where tblViTri.ID = tblTaiKhoan.IDViTri and tblNhanVien.ID =  

tblTaiKhoan.IDNV " +
        "order by tblTaiKhoan.ID DESC";
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.CommandType = CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    con.Close();
    grvTaikhoan.DataSource = dt;
}

private bool themtaikhoan()
{
    if (string.IsNullOrEmpty(txtTendangnhap.Text) ||
string.IsNullOrEmpty(txtMatkhau.Text))
    {
        MessageBox.Show("Vui lòng nhập đầy đủ thông tin tài khoản!", "Lưu Ý",
MessageBoxButtons.OK, MessageBoxIcon.Error);
        return false;
    }
    var con = conn.connect();
    string Check_exist_query = "SELECT TaiKhoan FROM tblTaiKhoan WHERE  

TaiKhoan = '" + txtTendangnhap.Text + "'";

```

```

        string Insert_query = "insert into tblTaiKhoan values(" +
int.Parse(txtIDtaikhoan.Text) + "," + txtTendangnhap.Text + "," + txtMatkhau.Text +
", " + int.Parse(cbbVitri.SelectedValue.ToString()) + ")";
        SqlCommand cmd = new SqlCommand(Check_exist_query, con);
        SqlDataReader dr = cmd.ExecuteReader();
        if (!dr.HasRows)
        {
            dr.Close();
            cmd = new SqlCommand(Insert_query, con);
            if (cmd.ExecuteNonQuery() > 0)
            {
                con.Close();
                return true;
            }
        }
        con.Close();
        MessageBox.Show("Tên tài khoản đã tồn tại!", "Lỗi", MessageBoxButtons.OK,
        MessageBoxIcon.Error);
        return false;
    }

    private void btnThem_Click(object sender, EventArgs e)
    {
        addcbbVitri();
        back();
        Int32 i;
        if (grvTaikhoan.RowCount == 1)
        {
            txtIDtaikhoan.Text = "1";
            Const.idtaikhoan = 1;
        }
        else
        {
            txtIDtaikhoan.Text =
(int.Parse(grvTaikhoan.Rows[0].Cells[0].Value.ToString()) + 1).ToString();
            Const.idtaikhoan = (int.Parse(grvTaikhoan.Rows[0].Cells[0].Value.ToString())
+ 1);
        }
        int id = int.Parse(grvTaikhoan.RowCount.ToString());
        txtIDtaikhoan.Text = id.ToString();
        enablecommand(false);
        kt = 1;
        var f = new Form_ThemTaiKhoan();
        f.ShowDialog();
    }

```



```

private bool xoataikhoan()
{
    var con = conn.connect();
    string Delete_query = "Delete FROM tblTaiKhoan WHERE ID = " +
int.Parse(txtIDtaikhoan.Text);
    SqlCommand cmd = new SqlCommand();
    cmd = new SqlCommand(Delete_query, con);
    if (cmd.ExecuteNonQuery() > 0)
    {
        con.Close();
        return true;
    }
    con.Close();
    return false;
}

private void btnXoa_Click(object sender, EventArgs e)
{
    binding();
    if (MessageBox.Show("Bạn thực sự muốn xóa tài khoản có mã " +
        txtIDtaikhoan.Text + " không???", "Warning", MessageBoxButtons.YesNo,
        MessageBoxIcon.Question) == DialogResult.Yes)
        if (xoataikhoan())
        {
            MessageBox.Show("Xóa tài khoản có mã " + txtIDtaikhoan.Text + " thành
công", "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
        loaddata();
        back();
    }

private void btnXacnhan_Click(object sender, EventArgs e)
{
    if (kt == 1)
    {
        if (themtaikhoan())
        {
            MessageBox.Show("Thêm tài khoản thành công", "Ràng buộc dữ liệu",
            MessageBoxButtons.OK, MessageBoxIcon.Information);
            enablecommand(true);
            loaddata();
            back();
        }
        else if (MessageBox.Show("Bạn muốn dừng thêm???", "Warning",
            MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            enablecommand(true);
    }
}

```

```

else if (kt == 2)
{
    if (capnhat())
    {
        MessageBox.Show("Cập nhật thông tin tài khoản thành công", "Ràng buộc dữ liệu", MessageBoxButtons.OK, MessageBoxIcon.Information);
        enablecommand(true);
        back();
        txtIDtaikhoan.Enabled = txtTendangnhap.Enabled = true;
    }
}

private bool capnhat()
{
    var con = conn.connect();
    string update_query = "update tblTaiKhoan set IDViTri = " +
int.Parse(cbbVitri.SelectedValue.ToString()) + ", MatKhau = " + txtMatkhau.Text + ""
+
    "where ID = " + int.Parse(txtIDtaikhoan.Text);
SqlCommand cmd = new SqlCommand(update_query, con);
if (cmd.ExecuteNonQuery() > 0)
{
    con.Close();
    Int32 i = Convert.ToInt32(grvTaikhoan.CurrentRow.Index);
    loaddata();
    grvTaikhoan.Rows[i].Selected = true;
    return true;
}
return false;
}

private void btnCapnhat_Click(object sender, EventArgs e)
{
    txtIDtaikhoan.Enabled = txtTendangnhap.Enabled = false;
    addcbbVitri();
    binding();
    enablecommand(false);
    kt = 2;
}

public Form_TaiKhoan()
{
    InitializeComponent();
}

private void Form_TaiKhoan_Load(object sender, EventArgs e)
{

```

```

        loaddata();
    }
}

```

- Lớp Form_TaiKhoan là một form trong ứng dụng quản lý khách sạn được tạo để hiển thị và quản lý thông tin tài khoản. Trong lớp này, có các biến và đối tượng sau:
- conn: Đối tượng của lớp ConnectDatabase để kết nối với cơ sở dữ liệu.
- Lớp này bao gồm các phương thức và sự kiện sau:
- back(): Phương thức để xóa dữ liệu đã nhập trong các textbox và combobox và đưa con trỏ về textbox txtTendangnhap.
- enablecommand(): Phương thức để kích hoạt hoặc vô hiệu hóa các nút điều khiển.
- binding(): Phương thức để ràng buộc dữ liệu từ GridView grvTaikhoan vào các textbox và combobox.
- addcbbVitri(): Phương thức để thêm dữ liệu vào combobox cbbVitri từ bảng tblViTri.
- loaddata(): Phương thức để tải dữ liệu tài khoản từ cơ sở dữ liệu và hiển thị lên GridView grvTaikhoan.
- themtaikhoan(): Phương thức để thêm tài khoản vào cơ sở dữ liệu.
- btnThem_Click(): Sự kiện khi nhấn nút "Thêm", gọi phương thức addcbbVitri(), back() và hiển thị form Form_ThemTaiKhoan.
- xoataikhoan(): Phương thức để xóa tài khoản khỏi cơ sở dữ liệu.
- btnXoa_Click(): Sự kiện khi nhấn nút "Xóa", gọi phương thức binding() và xác nhận xóa tài khoản đã chọn.
- btnXacnhan_Click(): Sự kiện khi nhấn nút "Xác nhận", kiểm tra nếu là thêm tài khoản thì gọi phương thức themtaikhoan(), nếu là cập nhật tài khoản thì gọi phương thức capnhat().
- capnhat(): Phương thức để cập nhật thông tin tài khoản vào cơ sở dữ liệu.
- btnCapnhat_Click(): Sự kiện khi nhấn nút "Cập nhật", gọi phương thức addcbbVitri(), binding() và vô hiệu hóa các textbox txtIDtaikhoan và txtTendangnhap.
- Form_TaiKhoan_Load(): Sự kiện khi form được tải lên, gọi phương thức loaddata() để hiển thị dữ liệu tài khoản.

2.1.12 Source code form thanh toán

```
public partial class Form_ThanhToan : Form
{
    #region GUI
    public const int WM_NCLBUTTONDOWN = 0xA1;
    public const int HT_CAPTION = 0x2;
    public bool isExit = true;
    public event EventHandler logout;
    [DllImportAttribute("user32.dll")]
    public static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);
    [DllImportAttribute("user32.dll")]
    public static extern bool ReleaseCapture();

    private void ptbclose_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Bạn muốn dừng thanh toán?", "Chú Ý!",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
        {
            Const.tatnutluu = false;
            this.Dispose();
        }
    }

    private void ptbminimize_Click(object sender, EventArgs e)
    {
        this.WindowState = FormWindowState.Minimized;
    }

    private void ptbstate_Click(object sender, EventArgs e)
    {
        if (this.WindowState == FormWindowState.Normal)
        {
            this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;
            this.WindowState = FormWindowState.Maximized;
        }
        else
        {
            this.WindowState = FormWindowState.Normal;
        }
    }

    private void pntop_MouseDown(object sender, MouseEventArgs e)
    {
        if (e.Button == MouseButtons.Left)
        {

```

```

        Cursor.Current = Cursors.SizeAll;
        ReleaseCapture();
        SendMessage(Handle, WM_NCLBUTTONDOWN, HT_CAPTION, 0);
    }
}
#endregion
public Form_ThanhToan()
{
    InitializeComponent();

    private void groupBox1_Enter(object sender, EventArgs e)
    {

    }

    private void Form_ThanhToan_Load(object sender, EventArgs e)
    {
        lbTenkhachhang.Text = Const.tenkhachhang;
        lbTenphong.Text = Const.tenphong;
        lbNgaythanhtoan.Text = Const.ngaythanhtoan;
        lbCanthanhtoan.Text = string.Format("{0:###0}",
int.Parse(Const.sotiencanthanhtoan));
    }
    private void txtTienkhachdua_KeyUp(object sender, KeyEventArgs e)
    {
        lbTienthua.Text = string.Format("{0:###0}", (int.Parse(txtTienkhachdua.Text) -
int.Parse(lbCanthanhtoan.Text.Replace(",", ""))));
    }

    private void btnXacnhan_Click(object sender, EventArgs e)
    {
        if (int.Parse(lbTienthua.Text.Replace(",", "")) >= 0)
        {
            Const.tatnutluu = true;
            Const.check = true;
            this.Dispose();
        }
        else
        {
            MessageBox.Show("Khách đưa chưa đủ tiền cần thanh toán!", "Chú Ý",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
        }
    }
}
}

```

- Lớp Form_ThanhToan là một form trong ứng dụng quản lý khách sạn được tạo để thực hiện thanh toán cho đơn hàng. Trong lớp này, có các thành phần GUI và các sự kiện tương ứng.
- Thành phần GUI:
 - ptbclose, ptbminimize, ptbstate: Các PictureBox đại diện cho nút đóng, thu nhỏ và phóng to.
 - pntop: Panel ở phía trên cùng của form để di chuyển form khi kéo thả.
 - Các nhãn (Label) để hiển thị thông tin khách hàng, phòng, ngày thanh toán và số tiền cần thanh toán.
 - txtTienkhachdua: TextBox để nhập số tiền khách hàng đưa.
 - lbTienthua: Nhãn để hiển thị số tiền thừa hoặc thiếu sau khi thanh toán.
 - btnXacnhan: Nút "Xác nhận" để hoàn tất thanh toán.
- Sự kiện GUI:
 - ptbclose_Click: Sự kiện khi nhấn nút đóng. Hiển thị hộp thoại xác nhận và đóng form nếu người dùng đồng ý.
 - ptbminimize_Click: Sự kiện khi nhấn nút thu nhỏ. Thu nhỏ form xuống thanh taskbar.
 - ptbstate_Click: Sự kiện khi nhấn nút phóng to/thu nhỏ. Phóng to hoặc thu nhỏ form.
 - pntop_MouseDown: Sự kiện khi nhấn chuột trái trên panel pntop. Cho phép di chuyển form bằng cách kéo thả.
 - Phương thức Form_ThanhToan_Load: Sự kiện khi form được tải lên. Hiển thị thông tin khách hàng, phòng, ngày thanh toán và số tiền cần thanh toán.
 - Sự kiện txtTienkhachdua_KeyUp: Sự kiện khi người dùng nhập giá trị trong TextBox txtTienkhachdua. Cập nhật số tiền thừa hoặc thiếu dựa trên số tiền khách hàng đưa.
 - Sự kiện btnXacnhan_Click: Sự kiện khi người dùng nhấn nút "Xác nhận". Kiểm tra số tiền thừa hoặc thiếu và đánh dấu hóa đơn đã được thanh toán. Đóng form sau khi hoàn tất thanh toán.

2.1.13 Source code form thêm tài khoản

```

public partial class Form_ThemTaiKhoan : Form
{
    private ConnectDatabase conn = new ConnectDatabase();
    #region GUI
    public const int WM_NCLBUTTONDOWN = 0xA1;
    public const int HT_CAPTION = 0x2;
    public bool isExit = true;
    public event EventHandler logout;
    [DllImportAttribute("user32.dll")]
    public static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);
    [DllImportAttribute("user32.dll")]
    public static extern bool ReleaseCapture();
    private void ptbclose_Click(object sender, EventArgs e)
    {
        if (MessageBox.Show("Bạn muốn dừng thêm tài khoản?", "Cảnh Báo",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.Yes)
            this.Dispose();
    }

    private void ptbminimize_Click(object sender, EventArgs e)
    {
        this.WindowState = FormWindowState.Minimized;
    }

    private void ptbstate_Click(object sender, EventArgs e)
    {
        if (this.WindowState == FormWindowState.Normal)
        {
            this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;

```

```

        this.WindowState = FormWindowState.Maximized;

    }
    else
    {
        this.WindowState = FormWindowState.Normal;
    }
}

private void pntop_MouseDown(object sender, MouseEventArgs e)
{
    if (e.Button == MouseButtons.Left)
    {
        Cursor.Current = Cursors.SizeAll;
        ReleaseCapture();
        SendMessage(Handle, WM_NCLBUTTONDOWN, HT_CAPTION, 0);
    }
}

#endregion

public Form_ThemTaiKhoan()
{
    InitializeComponent();
}

private void addcbbVitri()
{
    DataSet dataSet = new DataSet();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "Select * from tblViTri";

```



```

cmd.Connection = conn.connect();

SqlDataAdapter da = new SqlDataAdapter();
da.SelectCommand = cmd;
da.Fill(dataSet, "ViTri");
cbbVitri.DataSource = dataSet.Tables["ViTri"];
cbbVitri.DisplayMember = "TenViTri";
cbbVitri.ValueMember = "ID";
}

private void addcbbNhanvien()
{
    DataSet dataSet = new DataSet();
    SqlCommand cmd = new SqlCommand();
    cmd.CommandText = "Select * from tblNhanVien";
    cmd.Connection = conn.connect();
    SqlDataAdapter da = new SqlDataAdapter();
    da.SelectCommand = cmd;
    da.Fill(dataSet, "NhanVien");
    cbbTennhanvien.DataSource = dataSet.Tables["NhanVien"];
    cbbTennhanvien.DisplayMember = "TenNhanVien";
    cbbTennhanvien.ValueMember = "ID";
}

private void btnXacnhan_Click(object sender, EventArgs e)
{
    if (string.IsNullOrEmpty(txtTendangnhap.Text) ||
string.IsNullOrEmpty(txtMatkhau.Text))
    {
        MessageBox.Show("Vui lòng nhập đầy đủ thông tin tài khoản!", "Lưu Ý",
MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
}

```

```

    }

    var con = conn.connect();

    string Check_exist_query = "SELECT TaiKhoan FROM tblTaiKhoan WHERE
TaiKhoan = '" + txtTendangnhap.Text + "'";

    string Insert_query = "insert into tblTaiKhoan values(" +
int.Parse(txtIDtaikhoan.Text) + ", " +

    "" + int.Parse(cbbTennhanvien.SelectedValue.ToString()) + "," +
txtTendangnhap.Text + ", " +

    "" + txtMatkhau.Text + ", " + int.Parse(cbbVitri.SelectedValue.ToString()) + ")";

SqlCommand cmd = new SqlCommand(Check_exist_query, con);
SqlDataReader dr = cmd.ExecuteReader();
if (!dr.HasRows)
{
    dr.Close();

    cmd = new SqlCommand(Insert_query, con);
    if (cmd.ExecuteNonQuery() > 0)
    {
        con.Close();

        MessageBox.Show("Thêm tài khoản thành công", "Ràng buộc dữ liệu",
MessageBoxButtons.OK, MessageBoxIcon.Information);

        Const.loadgridview = true;

        this.Dispose();
    }
}
else
{
    con.Close();

    MessageBox.Show("Tên tài khoản đã tồn tại!", "Lỗi",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}

```

```

    }

    private void Form_ThemTaiKhoan_Load(object sender, EventArgs e)
    {
        txtIDtaikhoan.Enabled = false;
        addcbbVitri();
        addcbbNhanvien();
        txtIDtaikhoan.Text = Const.idtaikhoan.ToString();
    }
}

```

- Lớp Form_ThemTaiKhoan là một form trong ứng dụng quản lý khách sạn được tạo để thêm tài khoản mới. Trong lớp này, có các thành phần GUI và các sự kiện tương ứng.
- Thành phần GUI:
 - ptbclose, ptbminimize, ptbstate: Các PictureBox đại diện cho nút đóng, thu nhỏ và phóng to.
 - pntop: Panel ở phía trên cùng của form để di chuyển form khi kéo thả.
 - txtTendangnhap: TextBox để nhập tên đăng nhập.
 - txtMatkhau: TextBox để nhập mật khẩu.
 - cbbVitri: ComboBox để chọn vị trí của tài khoản.
 - cbbTennhanvien: ComboBox để chọn tên nhân viên liên kết với tài khoản.
 - btnXacnhan: Nút "Xác nhận" để thêm tài khoản.
- Sự kiện GUI:
 - ptbclose_Click: Sự kiện khi nhấn nút đóng. Hiện thị hộp thoại xác nhận và đóng form nếu người dùng đồng ý.
 - ptbminimize_Click: Sự kiện khi nhấn nút thu nhỏ. Thu nhỏ form xuống thanh taskbar.
 - ptbstate_Click: Sự kiện khi nhấn nút phóng to/thu nhỏ. Phóng to hoặc thu nhỏ form.
 - pntop_MouseDown: Sự kiện khi nhấn chuột trái trên panel pntop. Cho phép di chuyển form bằng cách kéo thả.
- Phương thức addcbbVitri: Load dữ liệu cho ComboBox cbbVitri từ bảng tblViTri.

- Phương thức addcbbNhanvien: Load dữ liệu cho ComboBox cbbTennhanvien từ bảng tblNhanVien.
- Sự kiện btnXacnhan_Click: Sự kiện khi người dùng nhấn nút "Xác nhận". Kiểm tra thông tin tài khoản nhập vào, kiểm tra xem tên tài khoản đã tồn tại chưa và thêm tài khoản vào cơ sở dữ liệu nếu hợp lệ.
- Phương thức Form_ThemTaiKhoan_Load: Sự kiện khi form được tải lên. Khởi tạo các giá trị ban đầu cho các thành phần và load dữ liệu vào các ComboBox.

2.1.14 Source code form thống kê

public partial class Form_ThongKe : Form

```
{
    private ConnectDatabase conn = new ConnectDatabase();
    private int kt;
    #region GUI
    public const int WM_NCLBUTTONDOWN = 0xA1;
    public const int HT_CAPTION = 0x2;
    public bool isExit = true;
    public event EventHandler logout;
    [DllImportAttribute("user32.dll")]
    public static extern int SendMessage(IntPtr hWnd, int Msg, int wParam, int lParam);
    [DllImportAttribute("user32.dll")]
    public static extern bool ReleaseCapture();

    private void ptbclose_Click(object sender, EventArgs e)
    {
        this.Dispose();
    }

    private void ptbminimize_Click(object sender, EventArgs e)
    {
        this.WindowState = FormWindowState.Minimized;
    }

    private void ptbstate_Click(object sender, EventArgs e)
    {
        if (this.WindowState == FormWindowState.Normal)
        {
            this.MaximizedBounds = Screen.FromHandle(this.Handle).WorkingArea;
            this.WindowState = FormWindowState.Maximized;
        }
        else
        {
            this.WindowState = FormWindowState.Normal;
        }
    }
}
```

```

    }

    private void pntop_MouseDown(object sender, MouseEventArgs e)
    {
        if (e.Button == MouseButtons.Left)
        {
            Cursor.Current = Cursors.SizeAll;
            ReleaseCapture();
            SendMessage(Handle, WM_NCLBUTTONDOWN, HT_CAPTION, 0);
        }
    }

    #endregion
    public Form_ThongKe()
    {
        InitializeComponent();
    }
    public readonly Color controng = Color.Green;

    private int count()
    {
        if (grvThongkephong.RowCount > 0)
        {
            int c = 0;
            for (int i = 0; i < grvThongkephong.RowCount - 1; i++)
            {
                string s = grvThongkephong.Rows[i].Cells["Trạng Thái"].Value.ToString();
                if (s == "Còn trống")
                {
                    c++;
                }
            }
            return c;
        }
        return -1;
    }

    private void formatgrid()
    {
        grvThongkephong.ColumnHeadersDefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
        grvThongkephong.DefaultCellStyle.Alignment =
DataGridViewContentAlignment.MiddleCenter;
        for (int i = 0; i < grvThongkephong.RowCount - 1; i++)
        {
            string s = grvThongkephong.Rows[i].Cells["Trạng Thái"].Value.ToString();

```

```

        if (s == "Còn trống")
        {
            grvThongkephong.Rows[i].DefaultCellStyle.BackColor = controng;
        }
        else
        {
            grvThongkephong.Rows[i].DefaultCellStyle.BackColor = Color.White;
        }
    }
}

private void loaddata()
{
    var con = conn.connect();
    string sql = "select tblPhong.ID as 'Mã Phòng', TenLoaiPhong as 'Tên Loại Phòng', TenPhong as 'Tên Phòng', TrangThai as 'Trạng Thái' from tblPhong, tblLoaiPhong where tblPhong.IDLoaiPhong = tblLoaiPhong.ID order by tblPhong.TrangThai ASC";
    SqlCommand cmd = new SqlCommand(sql, con);
    cmd.CommandType = CommandType.Text;
    DataTable dt = new DataTable();
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    da.Fill(dt);
    con.Close();
    grvThongkephong.DataSource = dt;
}

private void Form_ThongKe_Load(object sender, EventArgs e)
{
    loaddata();
    if (count() == -1)
    {
        lbSophongtrong.Text = "Không còn Phòng trống";
    }
    else
    {
        lbSophongtrong.Text = count().ToString();
    }
    formatgrid();
}
}

```

- Lớp Form_ThongKe là một form trong ứng dụng quản lý khách sạn được tạo để hiển thị và thống kê thông tin về các phòng trong khách sạn. Trong lớp này, có các thành phần GUI và các sự kiện tương ứng.
- Thành phần GUI:

- ptbclose, ptbminimize, ptbstate: Các PictureBox đại diện cho nút đóng, thu nhỏ và phóng to.
- pntop: Panel ở phía trên cùng của form để di chuyển form khi kéo thả.
- grvThongkephong: DataGridView để hiển thị thông tin về các phòng.
- lbSophongtrong: Label để hiển thị số lượng phòng trống.
- Sự kiện GUI:
 - ptbclose_Click: Sự kiện khi nhấn nút đóng. Đóng form.
 - ptbminimize_Click: Sự kiện khi nhấn nút thu nhỏ. Thu nhỏ form xuống thanh taskbar.
 - ptbstate_Click: Sự kiện khi nhấn nút phóng to/thu nhỏ. Phóng to hoặc thu nhỏ form.
 - pntop_MouseDown: Sự kiện khi nhấn chuột trái trên panel pntop. Cho phép di chuyển form bằng cách kéo thả.
- Phương thức count: Đếm số lượng phòng còn trống trong DataGridView grvThongkephong.
- Phương thức formatgrid: Định dạng hiển thị cho DataGridView grvThongkephong. Các phòng còn trống được làm nổi bật với màu nền xanh lá cây.
- Phương thức loaddata: Load dữ liệu về thông tin phòng từ cơ sở dữ liệu và hiển thị trong DataGridView grvThongkephong.
- Sự kiện Form_ThongKe_Load: Sự kiện khi form được tải lên. Thực hiện load dữ liệu và hiển thị số lượng phòng trống. Sau đó, áp dụng định dạng hiển thị cho DataGridView grvThongkephong.

2.1.15 Source code form trả phòng

```
public partial class Form_TraPhong : Form
{
    private ConnectDatabase conn = new ConnectDatabase();
    private DataSet ds = new DataSet();
    public Form_TraPhong()
    {
        InitializeComponent();
    }
}
```

```

    }
    private void loaddata()
    {
        var con = conn.connect();
        string sql = "select tblThuePhong.ID as 'Mã', TenLoaiPhong as 'Tên Loại Phòng',
        TenPhong as 'Tên Phòng', HoTen as 'Họ và Tên', " +
        "tblKhachHang.CCCD as 'CCCD/CMND', DienThoai as 'Số Điện Thoại',
        Convert(varchar, NgayThue, 103) as 'Ngày Thuê', " +
        "Convert(varchar, NgayTra,103) as 'Ngày Trả', GiaPhong as 'Giá Thuê',
        TienDatCoc as 'Tiền Đặt Cọc', tblThuePhong.TrangThai as 'Trạng Thái' " +
        "from tblLoaiPhong, tblPhong, tblKhachHang, tblThuePhong " +
        "where tblLoaiPhong.ID = tblPhong.IDLoaiPhong and tblPhong.ID =
        tblThuePhong.IDPhong " +
        "and tblThuePhong.CCCD = tblKhachHang.CCCD";
        SqlCommand cmd = new SqlCommand(sql, con);
        cmd.CommandType = CommandType.Text;
        DataTable dt = new DataTable();
        SqlDataAdapter da = new SqlDataAdapter(cmd);
        da.Fill(dt);
        con.Close();
        grvThuephong.DataSource = dt;
    }

    private void Form_TraPhong_Load(object sender, EventArgs e)
    {
        loaddata();
        grvThuephong.Hide();
    }

    private void btnHuy_Click(object sender, EventArgs e)
    {
        this.Dispose();
    }

    private void grvThuephong_CellContentClick(object sender,
    DataGridViewCellEventArgs e)
    {
        if (e.RowIndex >= 0)
        {
            lbHovaten1.Text = grvThuephong.Rows[e.RowIndex].Cells[0].ToString();
        }
    }
}

```

- Lớp Form_TraPhong là một form trong ứng dụng quản lý khách sạn được tạo để hiển thị thông tin và thực hiện việc trả phòng. Trong lớp này, có các thành phần GUI và các sự kiện tương ứng.

- Thành phần GUI:
- grvThuephong: DataGridView để hiển thị thông tin về các phiếu thuê phòng.
- btnHuy: Button để hủy việc trả phòng và đóng form.
- lbHovaten1: Label để hiển thị thông tin về họ và tên của khách hàng.
- Phương thức loaddata: Load dữ liệu về các phiếu thuê phòng từ cơ sở dữ liệu và hiển thị trong DataGridView grvThuephong.
- Sự kiện Form_TraPhong_Load: Sự kiện khi form được tải lên. Thực hiện load dữ liệu và ẩn DataGridView grvThuephong.
- Sự kiện btnHuy_Click: Sự kiện khi nhấn nút Hủy. Đóng form.
- Sự kiện grvThuephong_CellContentClick: Sự kiện khi chọn một ô trong DataGridView grvThuephong. Lấy thông tin về họ và tên của khách hàng và hiển thị trong Label lbHovaten1.

CHƯƠNG 3: TỔNG KẾT

3.1. ƯU ĐIỂM:

- ✓ Quản lý được những khách hàng đặt phòng và trả phòng dễ dàng hơn
- ✓ Thanh toán hóa đơn nhanh chóng
- ✓ Quản lý được thời gian sử dụng phòng của từng khách hàng cụ thể
- ✓ Báo cáo doanh thu theo ngày, tháng

3.2. KHUYẾT ĐIỂM:

Giao diện người dùng không thân thiện: Thiết kế giao diện người dùng không đơn giản và trực quan có thể làm cho người dùng khó sử dụng và hiểu cách thao tác trong ứng dụng.

Hiệu suất yếu: Nếu ứng dụng không được tối ưu hoặc không sử dụng các phương pháp lưu trữ dữ liệu hiệu quả, có thể gây ra trễ trong thời gian phản hồi hiệu suất chậm.

Quản lý dữ liệu không hiệu quả: Nếu không có một cơ sở dữ liệu được thiết kế tốt hoặc không sử dụng các phương pháp quản lý dữ liệu hiệu quả, có thể gặp vấn đề về đồng bộ hoá, xử lý trùng lặp dữ liệu hoặc mất mát dữ liệu trong quá trình xử lý.

Bảo mật yếu: Thiếu các biện pháp bảo mật như xác thực, phân quyền chi tiết, và mã hoá dữ liệu có thể gây ra lỗ hổng bảo mật và tiềm ẩn rủi ro rất lớn cho hệ thống.

Không có tính linh hoạt và mở rộng: Nếu ứng dụng không được thiết kế để dễ dàng mở rộng và thay đổi, thì việc thêm các tính năng mới hoặc thích nghi với nhu cầu kinh doanh có thể trở nên phức tạp hơn rất nhiều và tiêu tốn nhiều thời gian hơn. Từ đó gây ra rất nhiều khó khăn và để lại hệ quả nghiêm trọng.

Thiếu khả năng tương tác với hệ thống khác: Nếu ứng dụng không hỗ trợ tích hợp và tương tác với các hệ thống khác như hệ thống thanh toán, hệ thống đặt phòng trực tuyến, việc quản lý và liên kết dữ liệu có thể trở nên khó khăn hơn rất nhiều.

Thiếu tính năng cần thiết: Nếu ứng dụng không cung cấp đầy đủ các tính năng cần thiết cho quản lý khách sạn như đặt phòng, thanh toán, quản lý phòng, báo cáo thống kê, có thể làm hạn chế khả năng sử dụng và độ hiệu quả của ứng dụng bị giảm sút nghiêm trọng.

3.3. Hướng phát triển:

- Cải thiện giao diện người dùng: Tối ưu hóa giao diện người dùng để đảm bảo đơn giản, trực quan và dễ sử dụng. Tìm hiểu về các nguyên tắc thiết kế giao diện người dùng và sử dụng các thành phần UI hợp lý để cung cấp trải nghiệm tốt cho người dùng.
- Tối ưu hiệu suất: Nghiên cứu và tối ưu hoá việc truy vấn cơ sở dữ liệu, xử lý dữ liệu và tải dữ liệu để cải thiện hiệu suất của ứng dụng. Sử dụng các kỹ thuật caching và tối ưu hóa mã để đạt được thời gian phản hồi nhanh hơn.
- Tăng cường bảo mật: Đảm bảo rằng ứng dụng có các biện pháp bảo mật như xác thực người dùng, quản lý phiên, mã hóa dữ liệu và phân quyền truy cập. Cập nhật các biện pháp bảo mật theo các tiêu chuẩn và quy định hiện tại.

- **Mở rộng tính năng:** Xem xét việc thêm tính năng mới để cung cấp giá trị gia tăng cho khách hàng. Điều này có thể bao gồm tính năng đặt phòng trực tuyến, tích hợp thanh toán trực tuyến, tính năng quản lý nhân viên và quản lý dịch vụ khách sạn khác.
- **Tích hợp với hệ thống khác:** Nếu có yêu cầu, tích hợp ứng dụng với các hệ thống khác như hệ thống đặt phòng trực tuyến, hệ thống thanh toán, hệ thống quản lý khách hàng để cung cấp tính năng mạnh mẽ và tối ưu hóa quy trình làm việc.
- **Phát triển ứng dụng di động:** Xem xét việc phát triển ứng dụng di động để khách hàng có thể truy cập và quản lý thông tin từ xa. Ứng dụng di động cung cấp tính linh hoạt và tiện lợi cho người dùng.
- **Tạo báo cáo và thống kê:** Cải thiện khả năng tạo báo cáo và thống kê để cung cấp thông tin quản lý chi tiết về hoạt động khách sạn. Điều này giúp quản lý dễ dàng theo dõi hiệu suất, doanh thu và xu hướng khách hàng.
- **Tích hợp trí tuệ nhân tạo:** Nghiên cứu và áp dụng trí tuệ nhân tạo trong ứng dụng để cung cấp các dự đoán và khuyến nghị thông minh. Ví dụ, sử dụng học máy để dự đoán nhu cầu đặt phòng và tối ưu hóa giá cả và lợi nhuận.
- **Liên tục cải tiến và tương tác với khách hàng:** Lắng nghe ý kiến và phản hồi từ khách hàng để cải thiện và phát triển ứng dụng theo hướng tốt hơn. Điều này đảm bảo rằng ứng dụng đáp ứng đúng nhu cầu và mong muốn của khách hàng.

1. Microsoft Docs: <https://docs.microsoft.com/en-us/dotnet/csharp/>

Đây là nguồn tài liệu chính thống từ Microsoft, cung cấp hướng dẫn chi tiết, ví dụ và tài liệu về ngôn ngữ lập trình C#.

2. C# Station: <http://www.csharp-station.com/>

Trang web này cung cấp các bài hướng dẫn từ cơ bản đến nâng cao về lập trình C#. Bạn có thể tìm hiểu về cú pháp, kiến thức lập trình và các khía cạnh khác của ngôn ngữ.

3. C# Corner: <https://www.c-sharpcorner.com/>

C# Corner là một cộng đồng lập trình viên nơi bạn có thể tìm thấy bài viết, hướng dẫn và dự án liên quan đến C#. Ngoài tài liệu, bạn còn có thể tham gia thảo luận và chia sẻ kiến thức của mình.

4. TutorialsTeacher: <https://www.tutorialsteacher.com/csharp/csharp-tutorials>

Trang web này cung cấp các bài hướng dẫn chi tiết về C# với ví dụ minh họa dễ hiểu. Các chủ đề từ cơ bản đến nâng cao đều được bao quát.

5. C# Yellow Book: <https://www.csharpcourse.com/>

Cuốn "C# Yellow Book" của Rob Miles là một tài liệu miễn phí dành cho người mới học lập trình với C#. Nó giúp bạn nắm vững cơ bản của ngôn ngữ.

6. Codecademy: <https://www.codecademy.com/learn/learn-c-sharp>

Codecademy cung cấp khóa học trực tuyến về lập trình C# dành cho người mới học. Đây là một nền tảng học tốt để bắt đầu.

7. Stack Overflow C# Tag:

<https://stackoverflow.com/questions/tagged/c%23>

Stack Overflow là diễn đàn quan trọng cho lập trình viên. Bạn có thể tìm kiếm và đặt câu hỏi liên quan đến C# để nhận được sự giúp đỡ từ cộng đồng.

8. Reddit r/csharp: <https://www.reddit.com/r/csharp/>

Reddit cũng có cộng đồng chia sẻ và thảo luận về lập trình C#. Bạn có thể đặt câu hỏi, thảo luận về các vấn đề liên quan và chia sẻ kiến thức của bạn.