

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

**TRƯỜNG ĐẠI HỌC AN GIANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**LẬP TRÌNH QUẢN LÝ**

**Đề tài: QUẢN LÝ KHÁCH SẠN**

| **Sinh viên thực hiện:** | | **Giảng viên hướng dẫn:** |
| --- | --- | --- |
| 1. Nguyễn Trọng Khang 2. Ngô Anh Kiệt | DTH215955  DTH215982 | Nguyễn Ngọc Minh |

**AN GIANG, 10 THÁNG 5 NĂM 2024**

LỜI NÓI ĐẦU 3

NỘI DUNG BÁO CÁO 4

**I.** **Mục đích yêu cầu** 4

***1.*** ***Mục đích*** 4

***2.*** ***Yêu cầu*** 4

**II.** **Giới thiệu đồ án** 4

**1.** **Thông tin tổng quan** 4

**2.** **Cây chức năng** 5

**3.** **Thời gian thực hiện** 5

**4.** **Phân công nhiệm vụ** 6

**III.** **Đối tượng sử dụng** 6

**1.** **Đối tượng người dùng** 6

**2.** **Mô tả hệ thống** 6

**IV.** **Phân tích thiết kế hệ thống** 7

***1.*** ***Phân tích:*** 7

***2.*** ***Cơ sở dữ liệu:*** 9

**V.** **Giao diện chương trình** 10

**1.** **Login** 10

**2.** **Giao diện chính (DashBoard)** 17

**3.** **Quản lý đặt phòng (sơ đồ phòng)** 21

**4.** **Quản lý hóa đơn** 27

**5.** **Quản lý dịch vụ** 31

**6.** **Quản lý nhân viên** 35

**7.** **Quản lý phòng** 39

**8.** **Form trung gian** 46

**9.** **Ngoài ra còn có các UserControl và Form hổ trợ** 49

**VI.** **Kết luận và hướng phát triển:** 78

**VII.** **Tài liệu tham khảo** 79

# LỜI NÓI ĐẦU

Trong thế giới đang phát triển với sự bùng nổ của ngành du lịch, vai trò của quản lý khách sạn trở nên ngày càng quan trọng. Trong lời nói đầu này, chúng ta sẽ khám phá sâu hơn về tầm quan trọng và những thách thức mà quản lý khách sạn đối diện.

Quản lý khách sạn không chỉ đơn thuần là việc cung cấp chỗ ở cho khách du lịch, mà còn là một nghệ thuật kết hợp giữa chất lượng dịch vụ, quản lý tài nguyên, và chiến lược kinh doanh. Chúng ta sẽ đào sâu vào các khía cạnh của việc quản lý khách sạn, từ việc tối ưu hóa quy trình vận hành hàng ngày đến việc xây dựng một thương hiệu mạnh mẽ và đáp ứng nhu cầu đa dạng của khách hàng.

Ngoài ra ngành du lịch và khách sạn đang trở thành một trong những lĩnh vực quan trọng nhất của nền kinh tế, và Việt Nam không phải là ngoại lệ.

Với sự phát triển kinh tế và sự gia tăng về du lịch trong và ngoài nước, ngành khách sạn tại Việt Nam đang trải qua một giai đoạn phát triển đầy tiềm năng. Từ các khu nghỉ dưỡng ven biển đến các khách sạn sang trọng trong các thành phố lớn, cơ hội và thách thức đều hiện diện.

Vì thế, chúng em đã chọn đề tài quản lý khách sạn với mong muốn hiểu rõ hơn về cơ hội và thách thức trong lĩnh vực này, áp dụng kiến thức vào các tình huống thực tế. Kiến thức về quản lý khách sạn có tính toàn cầu, mở ra cơ hội tương tác với các mô hình quản lý khác nhau trên thế giới. Cuối cùng, ngành du lịch và khách sạn là một lĩnh vực tương tác xã hội cao, và thông qua việc nghiên cứu, chúng em mong muốn hiểu rõ hơn về cách tạo ra trải nghiệm tốt nhất cho khách hàng và tương tác trong môi trường làm việc đa văn hóa.

# NỘI DUNG BÁO CÁO

1. **Mục đích yêu cầu**
2. ***Mục đích***

* Hệ thống quản lý khách sạn sẽ là người bạn đồng hành đắc lực của nhà quản lý, giúp họ tổ chức và điều hành mọi hoạt động trong khách sạn một cách hiệu quả.
* Không chỉ đơn thuần là quản lý bằng cách thủ công, việc chuyển sang hình thức quản lý bằng hệ thống sẽ tạo ra một sự chuyên nghiệp và minh bạch
* Không còn những tình trạng sai sót khi tính lương hoặc thiếu sót khi lưu trữ thông tin khách hàng như cách quản lý thủ công
* Dễ dàng cập nhật, điều chỉnh, tìm kiếm, tra cứu thông tin khách hàng đều được thực hiện nhanh chóng và chính xác..
* Tăng hiệu suất công việc.

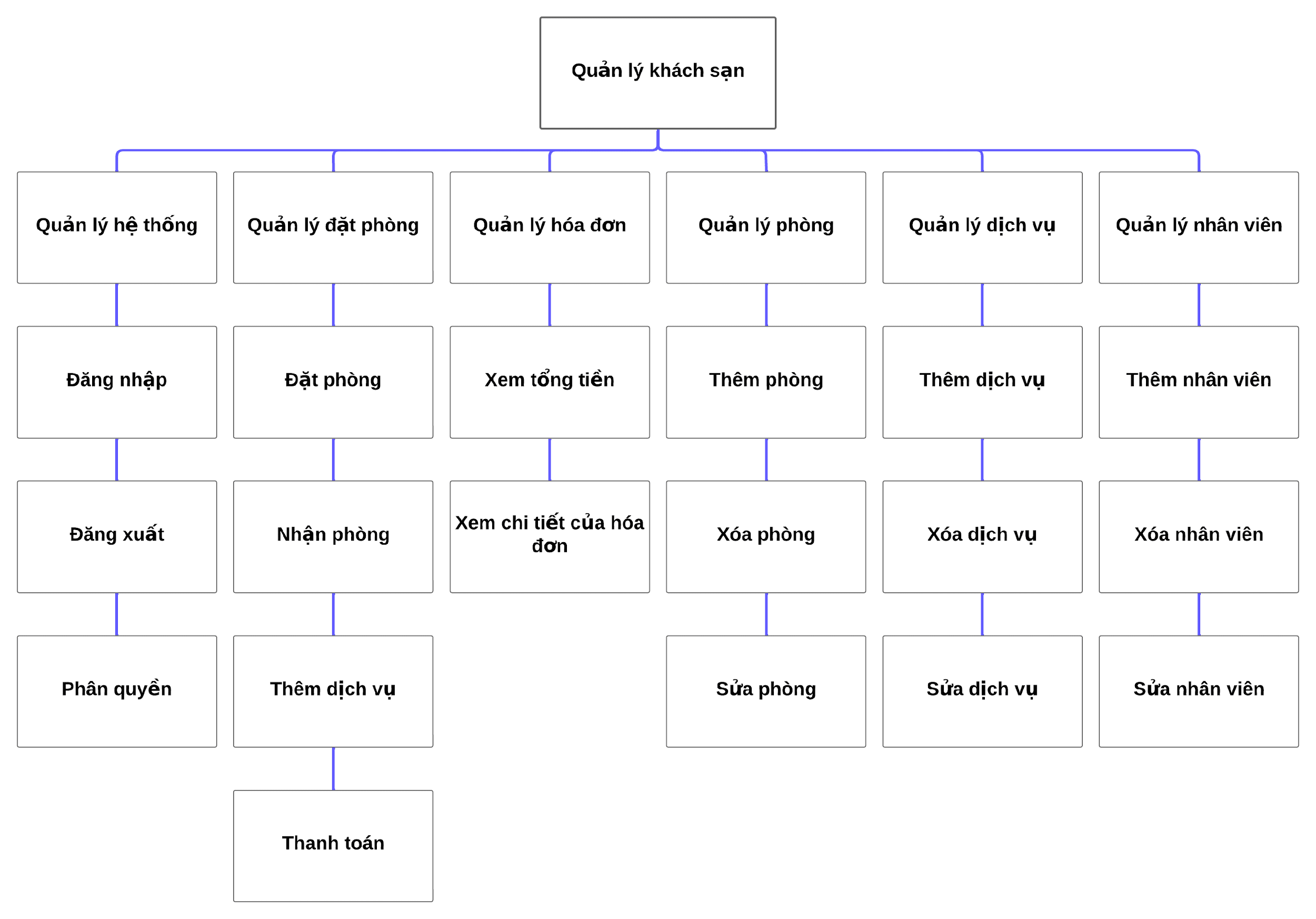
1. ***Yêu cầu***

* Giao diện đơn giản, thân thiện với người dùng.
* Các thành phần bố trí hợp lý, thuận tiện.
* Sau khi thêm thông tin nhân viên hoặc phòng ,… thì thông tin được lưu vào cơ sở dữ liệu.
* Có dung lượng lưu trữ lượng lớn thông tin.
* Có tính bảo mật, phải đăng nhập mới sử dụng được.

1. **Giới thiệu đồ án**
2. **Thông tin tổng quan**

* Trong ngành khách sạn, hệ thống quản lý đóng vai trò quan trọng trong việc tạo nên sự thành công của doanh nghiệp. Một hệ thống quản lý vững chắc không chỉ đảm bảo sự hài lòng của khách hàng mà còn đáp ứng được mong muốn của các nhà quản lý.
* Người làm công tác quản lý trong ngành khách sạn cần phải linh hoạt và liên tục điều chỉnh để đảm bảo nhu cầu của doanh nghiệp và duy trì sự hài lòng của khách hàng. Họ cũng phải bảo vệ quyền lợi của nhân viên và thúc đẩy sự phát triển năng lực bằng các chương trình đào tạo và các chính sách phúc lợi phù hợp.

1. **Cây chức năng**

****

1. **Thời gian thực hiện**

* Thời gian thực hiện từ: 30/4/2024 đến hết 10/5/2024
* Quy cách làm việc nhóm: Zalo

1. **Phân công nhiệm vụ**

| **Họ tên** | **Phân công công việc** |
| --- | --- |
| Nguyễn Trọng Khang | Giao diện dashboard, code và word |
| Ngô Anh Kiệt | Cơ sở dữ liệu , giao diện login |

1. **Đối tượng sử dụng**
2. **Đối tượng người dùng**

* Phần mềm quản lý khách sạn , cho phép người quản lý của khách sạn, có khả năng phụ trách về việc quản lý phòng , dich vụ , hóa đơn và có thể quản lý thông tin của nhân viên trong khách sạn.
* Được xây dựng với mục đích phục vụ cho việc quản lý và thực hiện đặt phòng , quản lý hóa đơn ...
* Admin: Trưởng phòng, phó phòng.
* User: Nhân viên của phòng lễ tân

1. **Mô tả hệ thống**

* Trong quá trình hoạt động, quản lý sẽ xem thông tin phòng trong khách sạn để có thể cập nhật các thông tin mới về đặt phòng , nhận phong , sử dụng dịch vụ và thanh toán ; đồng thời để phát hiện sai sót nhằm tiến hành thay thế, sửa đổi thông tin về đặt phòng, nhận phòng. Việc sửa đồi này được dựa trên cơ sở lấy thông tin cung cấp lại từ phía khách hàng .
* Đồng thời quản lý có thể xem hoặc điều chỉnh thông tin phòng , dịch vụ và nhân viên cho phù hợp yêu cầu.
* Khi khách hàng đặt phòng thì sẽ cung cấp thông tin về cá nhân như họ tên, CCCD , số điện thoại để nhân viên tiến hành đặt phòng và thiết lập ngày nhận phòng theo ý khách hàng
* Nếu khách hàng đặt trước các dịch vụ
* Khi khách hạn nhận phòng thì sẽ tạo phiếu nhận phòng và yêu cầu khách hàng chọn ngày trả phòng ; sau đó sẽ tạo hóa đơn cho khách hàng
* Khi khách hàng muốn trả phòng thì sẽ tạo phiếu in hóa đơn cho khách hàng và cập nhật hóa đơn này vào thanh toán
* Nếu khách hàng nhận phòng hoặc trả phòng trả hẹn thì hệ thống sẽ cảnh báo đồng thời có thể xóa phiếu đặt phòng của khách nhưng không tính phụ phí cho khách hàng đó
* Một nhân viên trong khách sạn thôi không làm việc hoặc được phân bổ ở vị trí khác ; quản lý sẽ tiến hành hủy bỏ thông tin về nhân viên đó hoặc cập nhật lại nhân viên.

1. **Phân tích thiết kế hệ thống**
2. ***Phân tích:***

* Quản lý hệ thống:
* Quản lý đăng nhập hệ thống.
* Quản lý tài khoản hệ thống.
* Phân quyền người dùng.
* Quản lý nhân viên:
* Thông tin nhân viên: mã nhân viên, họ tên nhân viên, chức vụ, số điện thoại, giới tính, địa chỉ, loại nhân viên. Tất cả các thông tin đều được lưu trữ vào cơ sở dữ liệu.
* Cập nhật, lưu trữ dữ liệu hồ sơ của nhân viên trong hệ thống .
* Đầu vào là thông tin nhân viên, bao gồm mã nhân viên, họ tên nhân viên, chức vụ, số điện thoại, giới tính, địa chỉ, loại nhân viên (thêm) . Đầu ra là một bảng danh sách ghi đầu đủ thông tin về nhân viên.
* Quản lý phòng:
* Thông tin phòng: mã phòng, tên phòng, loại phòng, đơn giá , tình trạng. Tất cả các thông tin đều được lưu trữ vào cơ sở dữ liệu.
* Cập nhật, lưu trữ dữ liệu của nhân viên trong hệ thống.
* Đầu vào là thông tin phòng, bao gồm mã phòng, tên phòng, loại phòng, đơn giá , tình trạng (thêm). Đầu ra là một bảng danh sách ghi đầu đủ thông tin về phòng.
* Quản lý dịch vụ:
* Thông tin dịch vụ: mã dịch vụ, tên dịch vụ, loại dịch vụ, đơn giá. Tất cả các thông tin đều được lưu trữ vào cơ sở dữ liệu.
* Cập nhật, lưu trữ dữ liệu của dịch vụ trong hệ thống.
* Đầu vào là thông tin dịch vụ, bao gồm mã dịch vụ, tên dịch vụ, loại dịch vụ, đơn giá(thêm). Đầu ra là một bảng danh sách ghi đầu đủ thông tin về dịch vụ.
* Quản lý hóa đơn:
* Theo dõi hóa đơn đã được thanh toán
* Xem chi tiết hóa đơn đó
* Quản lý đặt phòng:
* Theo dõi tình trạng các phòng
* Theo dõi sử dụng dịch vụ ngày nhận ngày trả và số tiền của khách hàng
* In ra hóa đơn sử dụng phòng của khách hàng
* Công thức tính thành tiền của hóa đơn: Tổng tiền = (Tiền phòng \* số ngày + tiền dịch vụ) + phụ thu.

*Trong đó:*

* Tổng tiền : là tổng tiền của hóa đơn
* Số ngày: là số ngày khách ở
* Tiền dịch vụ : tiền sử dụng dịch vụ của khách hàng
* Phụ thu: 10% giá trị của hóa đơn (vì không có chính sách phạt)

1. ***Cơ sở dữ liệu:***

* **CHUCVU(MACV**,TENCHUCVU)
* **TAIKHOAN**(TAIKHOAN,MATKHAU,CHUCVU)
* **KHACHHANG**(**MAKH,**TENKH,CCCD,SDT,DIACHI,GIOITINH)
* **LOAINHANVIEN**(**MALOAI,**TENLOAI)
* **NHANVIEN**(**MANV,**TENNV,SDT,DIACHI,GIOITINH,LOAINV)
* **TINHTRANHPHONG**(**MATINHTRANG**,TENTINHTRANG)
* **LOAIPHONG**(**MALOAIPHONG,**TENLOAIPHONG,SONGUOITOIDA)
* **PHONG**(**MAPHONG,**TENPHONG,LOAIPHONG,DONGIA,TINHTRANG)
* **LOAIDICHVU**(**MALOAIDV**,TENLOAI)
* **DICHVU**(**MADV,**TENDV,LOAI,DONGIA)
* **PHIEUDATPHONG**(**MADATPHONG**,MAPHONG,NGAYDAT,NGAYNHAN)
* **CHITIETPHIEUDATPHONG**(**MADATPHONG,MAPHONG,**NGAYDAT,NGAYNHAN)
* **PHIEUNHANPHONG**(**MANHANPHONG**,MADATPHONG,MAKHACHHANG)
* **CHITIETPHIEUNHANPHONG**(**MANHANPHONG,MADATPHONG,MAKHACHANG,**NGAYNHAN,NGAYTRADUKIEN,NGAYTRATHUCTE)
* **SUSDUNGDICHVU**(**MASUDUNGDV,MADV,MANHANPHONG,**SOLUONG)
* **HOADON**(**MAHOADON,MANHANVIEN,MAKHACHHANG,MANHANPHONG,**TONGTIEN,NGAYLAP)
* **CHITIETHOADON**(**MAHOADON,MAPHONG,MASUDUNGDV,**PHUTHU,TIENPHONG,TIENDICHVU,SONGAY,THANHTIEN)

|  |
| --- |

1. **Giao diện chương trình**
2. **Login**

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Picturebox** | **Name:** *pb\_logo*  **Sizemode:** *normal*  **Backcolor:** *255, 246, 218* | Hiển thị logo của khách sạn | | **Picturebox** | **Name:** *pb\_account*  **Sizemode:** *StretchImage*  **Backcolor:** *255, 246, 218* | Hiển thị hình ảnh minh họa nơi đăng nhập tài khoản | | **Picturebox** | **Name:** *pb\_password*  **Sizemode:** *StretchImage*  **Backcolor:** *255, 246, 218* | Hiển thị hình ảnh minh họa nơi đăng nhập mật khẩu | | **Textbox** | **Name:** *txt\_dangnhap\_account* | Nơi nhập tài khoản | | **Textbox** | **Name:** *txt\_dangnhap\_password* | Nơi nhập mật khẩu | | **Button** | **Name:** *btn\_dangnhap*  **Backcolor:** *255, 246, 218*  **Text:** *Đăng Nhập* | Nút đăng nhập | | **Button** | **Name:** *btn\_dangky*  **Backcolor:** *255, 246, 218*  **Text:** *Đăng ký* | Nút đăng ký chuyển sang panel đăng nhập | | **Panel** | **Name:** *panel\_dangnhap*  **Backcolor:** *255, 246, 218* | Chứa các control đăng nhập | | **Form** | **Name:** *Login*  **Text:** *Login* | Chứa các control | |

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Picturebox** | **Name:** *pb\_logo2*  **Sizemode:** *StretchImage*  **Backcolor:** *255, 246, 218* | Hiển thị logo của khách sạn | | **Label** | **Name:** *lbl\_dangky*  **Forecolor:** *Coral*  **Backcolor:** *255, 246, 218* | Hiển thị nơi đăng ký tài khoản | | **Label** | **Name:** *lbl\_quaylai*  **Forecolor:** *black*  **Backcolor:** *255, 246, 218*  **Image:** *bitmap.back* | Quay lại panel đăng nhập | | **Label** | **Name:** *lbl\_quanly*  **Forecolor:** *black*  **Backcolor:** *255, 246, 218* | Hiển thị chọn quản lý | | **Textbox** | **Name**: *txt\_dangky\_taikhoan* | Nơi nhập tài khoản | | **Textbox** | **Name:** *txt\_dangky\_matkhau* | Nơi nhập mật khẩu | | **Textbox** | **Name:** *txt\_dangky\_matkhau2* | Nơi nhập xác nhận mật khẩu | | **Textbox** | **Name:** *txt\_key* | Nơi nhập mã xác nhận | | **Button** | **Name:** *btn\_dangky\_dangky*  **Backcolor:** *255, 246, 218*  **Text:** *Đăng ký* | Nút đăng ký chuyển sang panel đăng nhập | | **Toggle** | **Name:** *tgl\_quanly* | Chọn như combobox xem có click vào tạo tại khoản quản lý | | **Panel** | **Name:** *panel\_dangky*  **Backcolor:** *255, 246, 218* | Chứa các control đăng ký | | * **Code**   using BUS;  using GUI;using Guna.UI2.WinForms;  namespace DoanLTQL  {  public partial class Login : Form  {  public Login()  {  InitializeComponent();  SetUp();  }  public void SetUp()  {  //Chỉnh kích thước của form lại và ẩn panel đăng ký đi  this.Size = new Size(this.Panel\_dangnhap.Width, 659);  Panel\_dangky.Hide();  //Chỉnh dấu panel\_key và chỉ kích hoạt khi người dùng chọn tạo tài khoản cho admin | quản lý  panel\_key.Hide();  }  private void Form1\_Load(object sender, EventArgs e)  {  }  /\* Nếu đăng ký với vai trò quản lý thì sẽ hiện panel yêu cầu người dùng nhập lisecne key \*/  private void tgl\_quanly\_CheckedChanged(object sender, EventArgs e)  {  Guna2ToggleSwitch obj = (Guna2ToggleSwitch)sender;  if (obj.Checked)  {  panel\_key.Show();  }  else  {  panel\_key.Hide();  }  }  private void TextBox\_Double\_Click(object sender, EventArgs e)  {  Guna2TextBox txt = (Guna2TextBox)sender;  txt.SelectAll();  }  /\* Đổi sang form đăng ký hoặc ngược lại \*/  public void ChuyenForm(int i)  {  if (i == 1)  {  Panel\_dangnhap.Hide();  Panel\_dangky.Show();  }  else  {  Panel\_dangnhap.Show();  Panel\_dangky.Hide();  }  }  #region Liên Quan Đến Nút Đăng Ký  private void btn\_dangky\_Click\_1(object sender, EventArgs e)  {  ChuyenForm(1);  }  /\* Đăng Ký Tài Khoản\*/  private void btn\_dangky\_dangky\_Click(object sender, EventArgs e)  {  string tk = txt\_dangky\_taikhoan.Text;  string mk = txt\_dangky\_matkhau.Text;  string mk2 = txt\_dangky\_matkhau2.Text;  if (tk.Length == 0)  {  MessageBox.Show(@"Vui lòng nhâp tài khoản", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  txt\_dangky\_taikhoan.Focus();  return;  }  if (mk.Length == 0)  {  MessageBox.Show(@"Vui lòng nhâp mật khẩu", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  txt\_dangky\_matkhau.Focus();  return;  }  if (!mk.Equals(mk2))  {  MessageBox.Show(@"Xác nhận mật khẩu không đúng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  txt\_dangky\_matkhau2.Focus();  return;  }  var kiemtra = BUS\_Taikhoan.Select\_By\_TaiKhoan(tk);  if (kiemtra != null)  {  MessageBox.Show(@"Tên tài khoản đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  txt\_dangky\_taikhoan.Focus();  return;  }  /\* Kiễm tra xem người dùng tạo tài khoản quản lý hay nhân viên \*/  if (panel\_key.Visible)  {  string liscence = txt\_key.Text;  if (liscence != "2444")/\* Tài khoản quản lý phải nhập đúng lisence key \*/  {  MessageBox.Show(@"Mã xác nhận không đúng", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  txt\_key.Focus();  return;  }  else  {  BUS\_Taikhoan.Insert(tk, mk, 1);  }  }  else  {  BUS\_Taikhoan.Insert(tk, mk, 2);  }  MessageBox.Show(@"Đăng ký thành công", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  ChuyenForm(2);  }  #endregion  #region Liên Quan Đến Sự Kiện Hover  private void lbl\_quaylai\_MouseEnter(object sender, EventArgs e)  {  Label obj = (Label)sender;  lbl\_quaylai.ForeColor = Color.FromArgb(250, 112, 112);  ChuyenForm(2);  }  private void lbl\_quaylai\_MouseLeave(object sender, EventArgs e)  {  Label obj = (Label)sender;  lbl\_quaylai.ForeColor = Color.Black;  ChuyenForm(2);  }  #endregion  /\*Xử lý khi click nút đăng nhập\*/  private void btn\_dangnhap\_Click(object sender, EventArgs e)  {  string tk = txt\_dangnhap\_account.Text;  string mk = txt\_dangnhap\_matkhau.Text;  /\* Kiễm tra tài khoản \*/  bool kiemtra = BUS\_Taikhoan.KiemTraTaiKhoan(tk, mk);  if (!kiemtra)  {  MessageBox.Show(@"Tài khoản hay mật khẩu không hợp lệ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  txt\_dangnhap\_account.Focus();  return;  }  else  {  MessageBox.Show(@"Đăng nhập thành công!!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  /\* Lấy loại tài khoản để hiển thị \*/  int LoaiDangNhap = (int)BUS\_Taikhoan.Select\_By\_TaiKhoan(tk).MaCV;  DashBoard dashBoard = new DashBoard(LoaiDangNhap);  /\* Ẩn form đăng nhập \*/  this.Hide();  if (dashBoard.ShowDialog() == DialogResult.Cancel) { this.Close(); }  }  }  }  } | | | |

1. **Giao diện chính (DashBoard)**

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Picturebox** | **Name:** *ptb\_avartar*  **Sizemode:** *StretchImage*  **Backcolor:** *Transparent* | Hiển thị avatar ( tùy chỉnh ) | | **Button** | **Name:** *btn\_logo*  **Backcolor:** *253, 201, 127*  **Text:** *Bunbo Hotel*  **Image:***bitmap.logo* | Logo của khách sạn | | **Button** | **Name:** *btn\_sodophong*  **Backcolor:** *253, 201, 127*  **Text:** *Sơ Đồ Phòng*  **Image:***bitmap.diagram* | Hiển thị sơ đồ phòng | | **Button** | **Name:** *btn\_thanhtoan*  **Backcolor:** *253, 201, 127*  **Text:** *Thanh Toán*  **Image:***bitmap.payment* | Hiển thị hóa đơn đã thanh toán và tổng thu | | **Button** | **Name:** *btn\_dichvu*  **Backcolor:** *253, 201, 127*  **Text:** *Dịch Vụ*  **Image:***bitmap.service* | Hiển thị dịch vụ | | **Button** | **Name:** *btn\_nhanvien*  **Backcolor:** *253, 201, 127*  **Text:** *Nhân Viên*  **Image:***bitmap.employee* | Hiển thị nhân viên | | **Button** | **Name:** *btn\_themphong*  **Backcolor:** *253, 201, 127*  **Text:** *Thêm Phòng*  **Image:***bitmap.room* | Hiển thị phòng | | **Panel** | **Name:** *panel\_left*  **Backcolor:** *253, 201, 127* | Chứa các control ở phía trên | | **Panel** | **Name:** *panel\_right*  **Backcolor:** *255, 246, 218* | Chứa các usercontrol | | **Form** | **Name:** *DashBoard*  **Text:** *DashBoard* | Chứa các control | | * **Code**   using Guna.UI2.WinForms;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class DashBoard : Form  {  SoDoPhongControl sodophong;  DichVuControl dichvu;  ThanhToanControl thanhtoan;  NhanVienControl nhanvien;  ThemPhongControl themphong;  int LoaiDangNhap;  public DashBoard(int loaiDangNhap)  {  InitializeComponent();  LoaiDangNhap = loaiDangNhap;  SetUp();  }  public void SetUp()  {  sodophong = new SoDoPhongControl();  dichvu = new DichVuControl();  thanhtoan = new ThanhToanControl();  nhanvien = new NhanVienControl();  themphong = new ThemPhongControl();  panel\_right.Controls.Add(sodophong);  panel\_right.Controls.Add(dichvu);  panel\_right.Controls.Add(thanhtoan);  panel\_right.Controls.Add(nhanvien);  panel\_right.Controls.Add(themphong);  /\* Đăng nhập bằng tài khoản nhân viên\*/  if (LoaiDangNhap == 2)  {  btn\_dichvu.Hide();  btn\_nhanvien.Hide();  btn\_themphong.Hide();  }  Show(sodophong);  }  public void HiddenAllUserControl()  {  foreach (var p in panel\_right.Controls.OfType<UserControl>())  {  p.Hide();  }  }  public void Show(UserControl use)  {  HiddenAllUserControl();  use.Show();  }  public void MouseEnter\_Label(object sender, EventArgs e)  {  var obj = (Label)sender;  obj.ForeColor = Color.FromArgb(130, 174, 255);  }  public void MouseLeave\_Label(object sender, EventArgs e)  {  var obj = (Label)sender;  obj.ForeColor = Color.White;  }  private void btn\_dichvu\_Click(object sender, EventArgs e)  {  Show(dichvu);  }  private void btn\_sodophong\_Click(object sender, EventArgs e)  {  Show(sodophong);  }  private void btn\_thanhtoan\_Click(object sender, EventArgs e)  {  Show(thanhtoan);  }  private void btn\_nhanvien\_Click(object sender, EventArgs e)  {  Show(nhanvien);  }  private void btn\_themphong\_Click(object sender, EventArgs e)  {  Show(themphong);  }  }  } | | | |

1. **Quản lý đặt phòng (sơ đồ phòng)**

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Button** | **Name:** *btn\_phgtrong*  **BorderColor:** *DarkBlue*  **Backcolor:** *Transparent*  **Text:** *Phòng trống*  **Image:***bitmap.bed* | Chỉ hiển thị phòng trống | | **Button** | **Name:** *btn\_cokhach*  **BorderColor:** *DarkBlue*  **Backcolor:** *Transparent*  **Text:** *Có khách*  **Image:***bitmap.sleep* | Chỉ hiển thị phòng có khách | | **Button** | **Name:** *btn\_dadat*  **BorderColor:** *DarkBlue*  **Backcolor:** *Transparent*  **Text:** *Được đặt*  **Image:***bitmap.calander* | Chỉ hiển thị phòng được đặt | | **Panel** | **Name:** *panel\_btn*  **Backcolor:** *Transparent* | Chứa các control ở phía trên | | **Button** | **Name:** *btn\_capnhat*  **BorderColor:** *DarkBlue*  **Backcolor:** *Transparent*  **Text:** *Cập nhật*  **Image:***bitmap.update* | Cập nhật lại sơ đồ phòng | | **Label** | **Name:** *lbl\_phongtrong*  **Backcolor:** *91, 164, 88*  **Forecolor:** *White*  **Text:** *Phòng trống* | Hiển thị số lượng phòng trống | | **Label** | **Name:** *lbl\_dattruoc*  **Backcolor:** *255, 151, 113*  **Forecolor:** *White*  **Text:** *Đặt trước* | Hiển thị số lượng phòng đặt trước | | **Label** | **Name:** *lbl\_dango*  **Backcolor:** *248, 96, 27*  **Forecolor:** *White*  **Text:** *Đang ở* | Hiển thị số lượng phòng đang ở | | **Panel** | **Name:** *Panel\_tool*  **Backcolor:** *253, 169, 169* | Chứa các control ở phía trên | | **FlowLayoutPanel** | **Name:** *panel\_data\_container*  **Backcolor:** *Transparent* | Chứa các usercontrol | | **Panel** | **Name:** *Panel\_showdata*  **Backcolor:** *253, 169, 169* | Chứa các control ở phía trên | | **UserControl** | **Name:** *SoDoPhongControl* | Chứa các control | | * **Code**   using BUS;  using DTO;  using Guna.UI2.WinForms;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class SoDoPhongControl : UserControl  {  private List<RoomItem> room;  int sl\_phongtrong;  int sl\_cokhach;  int sl\_duocdat;  public enum TypeofDisplay  {  All = 0,  Empty = 1,  Lived = 2,  Booked = 3,  }  public SoDoPhongControl()  {  InitializeComponent();  room = new List<RoomItem>();  DisplayRoom(TypeofDisplay.All);  }  /\* Hiển thị lên flowlayoutpanel \*/  public void DisplayRoom(TypeofDisplay loai)  {  /\* Xóa hết dữ liệu \*/  panel\_data\_container.Controls.Clear();  room.Clear();  GetData();  /\* Gán lên flowlayoutpanel \*/  foreach (RoomItem item in room)  {  if (loai == TypeofDisplay.All)  {  panel\_data\_container.Controls.Add(item);  }  else if (loai == TypeofDisplay.Empty && item.Getlbl\_trangthai.Text.Contains("trống"))  {  panel\_data\_container.Controls.Add(item);  }  else if (loai == TypeofDisplay.Lived && item.Getlbl\_trangthai.Text.Contains("khách"))  {  panel\_data\_container.Controls.Add(item);  }  else if (loai == TypeofDisplay.Booked && item.Getlbl\_trangthai.Text.Contains("đặt"))  {  panel\_data\_container.Controls.Add(item);  }  }  }  /\*Lấy dữ liệu\*/  public void GetData()  {  /\* Lấy dữ liệu phòng \*/  var phong = BUS\_Phong.Select\_All();  /\* Chỉnh lại số lượng \*/  sl\_phongtrong = 0;  sl\_cokhach = 0;  sl\_duocdat = 0;  /\* Đưa dữ liệu vào room \*/  foreach (Phong phg in phong)  {  LoaiPHG loai = BUS\_LoaiPHG.Select\_By\_MaLoai(phg.LoaiPHG);  TinhTrangPHG tinhtrang = BUS\_TinhTrangPHG.Select\_By\_MaTinhTrang((int)phg.TinhTrang);  RoomItem roomItem = new RoomItem(phg.MaPHG, loai.TenloaiPHG, tinhtrang.TenTinhTrg, phg.Dongia);  room.Add(roomItem);  /\* Đếm số lượng phòng theo tình trạng \*/  if (tinhtrang.MaTinhTrg == 1)  {  sl\_phongtrong++;  }  else if (tinhtrang.MaTinhTrg == 2)  {  sl\_cokhach++;  }  else if (tinhtrang.MaTinhTrg == 3)  {  sl\_duocdat++;  }  }  lbl\_phongtrong.Text = string.Format($"Phòng trống ({sl\_phongtrong.ToString()})");  lbl\_dango.Text = string.Format($"Có khách ({sl\_cokhach.ToString()})");  lbl\_dattruoc.Text = string.Format($"Được đặt ({sl\_duocdat.ToString()})");  }  /\* Hàm hiển thị danh sách phòng theo loại tình trạng \*/  private void btn\_Click(object sender, EventArgs e)  {  var btn = (Guna2Button)sender;  /\* Click 2 lần vào một button \*/  if (btn.BorderThickness == 4)  {  btn.BorderThickness = 0;  DisplayRoom(TypeofDisplay.All);  return;  }  /\* Click 1 lần vào một button có thể khác nhau \*/  foreach (Control control in panel\_btn.Controls)  {  var button = (Guna2Button)control;  /\* Tắt chọn cho tât cả button \*/  if (button.BorderThickness == 4)  {  button.BorderThickness = 0;  }  }  if (btn == btn\_phgtrong)  {  btn.BorderThickness = 4;  DisplayRoom(TypeofDisplay.Empty);  }  else if (btn == btn\_cokhach)  {  btn.BorderThickness = 4;  DisplayRoom(TypeofDisplay.Lived);  }  else if (btn == btn\_dadat)  {  btn.BorderThickness = 4;  DisplayRoom(TypeofDisplay.Booked);  }  }  private void btn\_capnhat\_Click(object sender, EventArgs e)  {  DisplayRoom(TypeofDisplay.All);  }  }  } | | | |



1. **Quản lý hóa đơn**

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Button** | **Name:** *btn\_logo*  **BorderColor:** *Black*  **BorderThickness:** *2*  **Backcolor:** *253, 201, 127*  **Forecolor:** *info*  **Text:** *Payment*  **Image:***bitmap.logo* | Sử dụng như label | | **Picturebox** | **Name:** *ptb\_tim*  **Sizemode:** *Zoom*  **Backcolor:** *Transparent* | Hiển thị logo tìm kiếm | | **Textbox** | **Name:** *txt\_timkiem*  **Backcolor:** *Transparent* | Nơi nhập tên để tìm kiếm | | **Panel** | **Name:** *Panel\_tool*  **Backcolor:** *253, 201, 127* | Chứa các control ở phía trên | | **DataGridView** | **Name:** *dgv\_chitiethoadon* | Hiển thị dữ liệu | | **Panel** | **Name:** *pn\_tongtien*  **Backcolor:** *58, 100, 100* | Chứa hiển thị doanh thu | | **Label** | **Name:** *lbl\_doanhthu*  **Backcolor:** *Transparent*  **Forecolor:** *White*  **Text:** *Tổng danh thu* | hiển thị doanh thu | | **Textbox** | **Name:** *txt\_doanhthu*  **Backcolor:** *58, 100, 100*  **Forecolor:** *100, 188, 176* | hiển thị doanh thu | | **UserControl** | **Name:** *ThanhToanControl* | Chứa các control | | * **Code**   using BUS;  using DAO;  using DTO;  using Guna.UI2.WinForms;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class ThanhToanControl : UserControl  {  public ThanhToanControl()  {  InitializeComponent();  UpLoadData(GetData());  }  /\* HÀM CẬP NHẬT DOANH THU \*/  public void CapNhatDoanhThu()  {  float doanhthu = BUS\_HoaDon.GetDoanhSo();  txt\_doanhthu.Text = string.Format($"{doanhthu:C0}");  }  /\* HÀM LẤY DỮ LIỆU \*/  public object GetData()  {  object data = BUS\_HoaDon.DanhSachHoaDonDaThanhToan();  return data;  }  /\* ĐỔ DỮ LIỆU LÊN DATAGRIDVIEW \*/  public void UpLoadData(object data)  {  HienThiChuan();  CapNhatDoanhThu();  dgv\_chitiethoadon.DataSource = data;  }  /\* Hàm hiển thị đúng số cột như design \*/  public void HienThiChuan()  {  dgv\_chitiethoadon.AutoGenerateColumns = false;  foreach (DataGridViewColumn col in dgv\_chitiethoadon.Columns)  {  if (col is DataGridViewTextBoxColumn) col.Name = col.DataPropertyName;  }  }  /\*Hàm Chức NĂNG TỰ ĐỘNG TÌM KIẾM THEO TÊN KHÁCH HÀNG\*/  private void txt\_timkiem\_TextChanged(object sender, EventArgs e)  {  if (txt\_timkiem.Text.Length == 0)  {  UpLoadData(GetData());  return;  }  var data = BUS.BUS\_ChiTietHoaDon.TimKiemTheoTen(txt\_timkiem.Text.Trim());  if (data != null)  {  UpLoadData(data);  }  }  /\* CLICK VÀO BUTTON CHI TIẾT \*/  private void dgv\_chitiethoadon\_CellClick(object sender, DataGridViewCellEventArgs e)  {  var reciver = (Guna2DataGridView)sender;  if (e.RowIndex < 0 || e.RowIndex >= reciver.RowCount)  {  return;  }  if (e.ColumnIndex < 0 || e.ColumnIndex >= reciver.ColumnCount)  {  return;  }  /\*Kiễm tra click vào button chi tiết\*/  if (reciver.Columns[e.ColumnIndex].Name == "ChiTiet")  {  int MaChtietHD = Convert.ToInt32(dgv\_chitiethoadon.Rows[e.RowIndex].Cells["MaHD"].Value.ToString());  MoPhieuChiTietHoaDon(MaChtietHD);  }  }  /\* Xử lý khi người dùng click vào chi tiết hóa đơn \*/  private void MoPhieuChiTietHoaDon(int MaHD)  {  var obj = new ChiTietHoaDonControl();  obj.MaHD = MaHD;  /\*Cap nhat du lieu truoc khi them vao from\*/  obj.UpLoadData();  var form = new ShowUserControl();  form.StartPosition = FormStartPosition.CenterScreen;  form.Controls.Add(obj);  form.SetSuKien(obj);  form.Size = obj.Size;  form.ShowDialog();  }  /\* Cập nhật lại dữ liệu \*/  private void btn\_capnhat\_Click(object sender, EventArgs e)  {  UpLoadData(GetData());  }  }  } | | | |

1. **Quản lý dịch vụ**

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Button** | **Name:** *btn\_logo*  **BorderColor:** *Black*  **BorderThickness:** *2*  **Backcolor:** *253, 201, 127*  **Forecolor:** *info*  **Text:** *Service*  **Image:***bitmap.logo* | Sử dụng như label | | **DataGridView** | **Name:** *dgv\_dichvu* | Hiển thị dữ liệu | | **Panel** | **Name:** *Panel\_bottom*  **Backcolor:** *253, 201, 127* | Chứa các control liên quan đến thêm | | **Button** | **Name:** *btn\_datphong*  **Backcolor:** *59, 106, 10*  **Text:** *Thêm* | Mở user control đăng ký dịch vụ | | **UserControl** | **Name:** *DichVuControl* | Chứa các control | | * **Code**   using BUS;  using DAO;  using Guna.UI2.WinForms;  using Microsoft.VisualBasic.Logging;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class DichVuControl : UserControl  {  public DichVuControl()  {  InitializeComponent();  UploadData();  }  public void UploadData()  {  /\*HIỂN THỊ CỘT CHUẨN\*/  HienThiChuan();  var data = BUS\_DichVu.Select\_All();  //data.ForEach(x => x.Print());  dgv\_dichvu.DataSource = data;  /\*ĐỔ DỮ LIỆU VÔ COMBOBOX CỦA DATAGRID VÀ ĐIỀU CHỈNH CHO PHÙ HỢP\*/  var cbb\_source = (from p in BUS.BUS\_LoaiDV.Select\_All()  select p).ToList();  var combobox = (DataGridViewComboBoxColumn)dgv\_dichvu.Columns["LoaiDV"];  combobox.DataSource = cbb\_source;  combobox.DisplayMember = "TenLoaiDV";  combobox.ValueMember = "MaLoaiDV";  }  /\* HÀM HIỂN THỊ CỘT GIỐNG NHƯ BÊN DESIGN \*/  public void HienThiChuan()  {  dgv\_dichvu.AutoGenerateColumns = false;  /\* ĐẶT LẠI NAME CHO TỪNG CỘT \*/  foreach (var c in dgv\_dichvu.Columns)  {  if (c is DataGridViewTextBoxColumn)  {  DataGridViewColumn col = (DataGridViewColumn)c;  col.Name = col.DataPropertyName;  }  if (c is DataGridViewComboBoxColumn)  {  DataGridViewComboBoxColumn col = (DataGridViewComboBoxColumn)c;  col.Name = col.DataPropertyName;  }  }  }  /\* CẬP NHẬT VÀ XÓA \*/  private void dgv\_dicvu\_CellClick(object sender, DataGridViewCellEventArgs e)  {  /\* KIỄM TRA XEM NGƯỜI DÙNG CÓ CLICK ĐÚNG DÒNG HAY CỘT TRONG DATAGRIDVIEW HAY KHÔNG \*/  var reciver = (Guna2DataGridView)sender;  if (e.RowIndex < 0 || e.RowIndex >= reciver.RowCount)  {  return;  }  if (e.ColumnIndex < 0 || e.ColumnIndex >= reciver.ColumnCount)  {  return;  }  /\*Kiễm tra click vào nut update\*/  if (reciver.Columns[e.ColumnIndex].Name == "Update")  {  var check = MessageBox.Show("Bạn có muốn cập nhật dịch vụ này ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (check == DialogResult.Yes)  {  var dichvu = dgv\_dichvu.Rows[e.RowIndex];  var result = BUS.BUS\_DichVu.Update((int)dichvu.Cells["MaDV"].Value, dichvu.Cells["TenDV"].Value + "", Convert.ToInt32(dichvu.Cells["LoaiDV"].Value), float.Parse(dichvu.Cells["Dongia"].Value.ToString()));  if (result)  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  /\* Cập nhật lại dữ liệu lên datagridview \*/  UploadData();  }  else  {  MessageBox.Show("Thất Bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  /\*Kiễm tra click vào nut Delete\*/  if (reciver.Columns[e.ColumnIndex].Name == "Delete")  {  var check = MessageBox.Show("Bạn có muốn xóa dịch vụ này ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (check == DialogResult.Yes)  {  var maphong = (int)reciver.Rows[e.RowIndex].Cells["MaDV"].Value;  var result = BUS.BUS\_DichVu.Delete(maphong);  if (result)  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  /\*Cập nhật lại dữ liệu lên datagridview\*/  UploadData();  }  else  {  MessageBox.Show("Thất Bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  }  /\* Hàm xử lý khi click vào nút thêm nhân viên \*/  private void btn\_datphong\_Click(object sender, EventArgs e)  {  /\* Nhóm các thao tác mở form để thực hiện thêm nhân viên \*/  var obj = new ThemDichVu();  var form = new ShowUserControl();  form.StartPosition = FormStartPosition.CenterScreen;  form.Controls.Add(obj);  form.Size = obj.Size;  form.SetSuKien(obj);  var result = form.ShowDialog();  if (result == DialogResult.Yes) { UploadData(); }  }  }  } | | | |

1. **Quản lý nhân viên**

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Button** | **Name:** *btn\_logo*  **BorderColor:** *Black*  **BorderThickness:** *2*  **Backcolor:** *253, 201, 127*  **Forecolor:** *info*  **Text:** *Employee*  **Image:***bitmap.logo* | Sử dụng như label | | **Button** | **Name:** *btn\_themnv*  **Backcolor:** *Transparent*  **Text:** *Thêm nhân viên*  **Image:***bitmap.add* | Mở user control thêm nhân viên | | **Button** | **Name:** *btn\_tim*  **Backcolor:** *Transparent*  **Image:***bitmap.find* | Sử dụng như picturebox | | **Textbox** | **Name:** *txt\_timkiem*  **Backcolor:** *Transparent* | Nơi nhập tên nhân viên để tìm kiếm | | **Panel** | **Name:** *Panel\_tool*  **Backcolor:** *253, 201, 127* | Chứa các control liên quan đến thêm | | **DataGridView** | **Name:** *dgv\_nhanvien* | Hiển thị dữ liệu | | **UserControl** | **Name:** *NhanVienControl* | Chứa các control | | * **Code**   using BUS;  using DAO;  using DTO;  using Guna.UI2.WinForms;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class NhanVienControl : UserControl  {  public NhanVienControl()  {  InitializeComponent();  UpLoadData(GetData());  }  /\* Hàm lấy dữ liệu \*/  public List<NhanVien> GetData()  {  return BUS.BUS\_NhanVien.Select\_All();  }  /\*Hàm Đổ dữ liệu nhân viên lên datagirdview\*/  private void UpLoadData(List<NhanVien> data)  {  /\*Hiển thị đúng cột cho datagridview\*/  HienThiChuan();  var cbb\_source = BUS.BUS\_LoaiNV.Select\_All();  dgv\_nhanvien.DataSource = data;  /\*Hiển thị dữ liệu loại nhân viên lên cột combobox của datagridview\*/  var combobox = (DataGridViewComboBoxColumn)dgv\_nhanvien.Columns["LoaiNV"];  combobox.DisplayMember = "TenLoai";  combobox.ValueMember = "MaLoai";  combobox.DataSource = cbb\_source;  }  /\*Hàm Giúp Hiển thị đúng cột cho datagridview\*/  private void HienThiChuan()  {  dgv\_nhanvien.AutoGenerateColumns = false;  }  /\* Hàm xử lý khi click vào nút thêm nhân viên \*/  private void btn\_themnv\_Click(object sender, EventArgs e)  {  /\* Nhóm các thao tác mở form để thực hiện thêm nhân viên \*/  var obj = new ThemNhanVien();  var form = new ShowUserControl();  form.StartPosition = FormStartPosition.CenterScreen;  form.Controls.Add(obj);  form.Size = obj.Size;  form.SetSuKien(obj);  var result = form.ShowDialog();  if (result == DialogResult.Yes) { UpLoadData(GetData()); }  }  /\*CẬP NHẬT VÀ XÓA NHÂN VIÊN\*/  private void dgv\_nhanvien\_CellClick(object sender, DataGridViewCellEventArgs e)  {  /\* KIỄM TRA XEM NGƯỜI DÙNG CÓ CLICK ĐÚNG DÒNG HAY CỘT TRONG DATAGRIDVIEW HAY KHÔNG \*/  var reciver = (Guna2DataGridView)sender;  if (e.RowIndex < 0 || e.RowIndex >= reciver.RowCount)  {  return;  }  if (e.ColumnIndex < 0 || e.ColumnIndex >= reciver.ColumnCount)  {  return;  }  /\*Kiễm tra click vào nut update\*/  if (reciver.Columns[e.ColumnIndex].Name == "Update")  {  var check = MessageBox.Show("Bạn có muốn cập nhật nhân viên này ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (check == DialogResult.Yes)  {  var phong = dgv\_nhanvien.Rows[e.RowIndex];  var result = BUS.BUS\_NhanVien.Update((int)phong.Cells["Manv1"].Value, phong.Cells["TenNV1"].Value + "", phong.Cells["SDT1"].Value + "", phong.Cells["Diachi"].Value + "", phong.Cells["Gioitinh"].Value + "", Convert.ToInt32(phong.Cells["LoaiNV"].Value));  if (result)  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  /\*Cập nhật lại dữ liệu lên datagridview\*/  UpLoadData(GetData());  }  else  {  MessageBox.Show("Thất Bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  /\*Kiễm tra click vào nut Delete\*/  if (reciver.Columns[e.ColumnIndex].Name == "Delete")  {  var check = MessageBox.Show("Bạn có muốn xóa nhân viên này ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (check == DialogResult.Yes)  {  var maphong = (int)reciver.Rows[e.RowIndex].Cells["Manv1"].Value;  var result = BUS.BUS\_NhanVien.Delete(maphong);  if (result)  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  /\*Cập nhật lại dữ liệu lên datagridview\*/  UpLoadData(GetData());  }  else  {  MessageBox.Show("Thất Bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  }  /\* HÀM XỬ LÝ TÌM KIẾM NHÂN VIÊN THEO TÊN \*/  private void txt\_timkiem\_TextChanged(object sender, EventArgs e)  {  var data = BUS.BUS\_NhanVien.TimNhanVien(txt\_timkiem.Text.Trim());  if (data != null)  {  UpLoadData(data);  }  }  }  } | | | |

1. **Quản lý phòng**

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Button** | **Name:** *btn\_logo*  **BorderColor:** *Black*  **BorderThickness:** *2*  **Backcolor:** *253, 201, 127*  **Forecolor:** *info*  **Text:** *Room*  **Image:***bitmap.logo* | Sử dụng như label | | **Label** | **Name:** *lbl\_thongtinchung*  **Backcolor:** *Transparent*  **Text:** *Thông tin chung* | Hiển thị nơi thêm phòng | | **Textbox** | **Name:** *lbl\_A*  **Backcolor:** *Transparent*  **Forecolor:** *192, 0, 0*  **BorderThickness:** *2*  **BorderColor:** *Black*  **Text:** *A* | Sử dụng như label mặc định tên phòng bắt đầu từ ‘A’ | | **Textbox** | **Name:** *txt\_tenphong*  **Backcolor:** *Transparent* | Nơi nhập tên phòng | | **Label** | **Name:** *lbl\_loai*  **Backcolor:** *Transparent*  **Text:** *Loại phòng* | Hiển thị | | **Combobox** | **Name:** *cb\_loaiphong*  **Backcolor:** *Transparent*  **Forecolor:** *68, 88, 112* | Hiển thị chọn loại phòng | | **Textbox** | **Name:** *txt\_gia*  **Backcolor:** *Transparent* | Nơi nhập giá phòng | | **Button** | **Name:** *btn\_them*  **Backcolor:** *Transparent*  **Text:** *Thêm phòng*  **Forecolor:** *13, 152, 235*  **Image:***bitmap.add* | Thêm phòng vừa mới tạo vào csdl | | **Button** | **Name:** *btn\_capnhat*  **Backcolor:** *Transparent*  **Text:** *Cập nhật*  **Forecolor:** *13, 152, 235*  **Image:***bitmap.updatre* | Cập nhật lại phòng | | **Panel** | **Name:** *Panel\_tool*  **Backcolor:** *255, 249, 230* | Chứa các control liên quan đến thêm | | **DataGridView** | **Name:** *dgv\_phong* | Hiển thị dữ liệu | | **UserControl** | **Name:** *PhongControl* | Chứa các control | | * **Code**   using DTO;  using Guna.UI2.WinForms;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Security.Cryptography.X509Certificates;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  using static System.Windows.Forms.VisualStyles.VisualStyleElement;  namespace GUI  {  public partial class ThemPhongControl : UserControl  {  public int Count\_Update = 0;  public ThemPhongControl()  {  InitializeComponent();  UpLoadData();  }  private void ThemPhongControl\_Load(object sender, EventArgs e)  {  /\* DO NOTTHING \*/  }  /\*PHƯƠNG THỨC ĐỔ DỮ LIỆU LÊN DATAGRIDVIEW\*/  private void UpLoadData()  {  /\* Lấy dữ liệu phòng và loai phòng\*/  var data = BUS.BUS\_Phong.Select\_All();  var loaiphg = BUS.BUS\_LoaiPHG.Select\_All();  dgv\_phong.DataSource = data;  /\*Điều chỉnh lại giá trị dữ liệu\*/  dgv\_phong.Columns["Dongia"].DefaultCellStyle.Format = "C0";  var combobox = (DataGridViewComboBoxColumn)dgv\_phong.Columns["loai"];  combobox.DisplayMember = "TenloaiPHG";  combobox.ValueMember = "MaloaiPHG";  combobox.DataSource = loaiphg;  /\*Đổ dữ liệu lên combobox\*/  UpLoadCombobox();  }  /\*PHƯƠNG THỨC ĐỔ DỮ LIỆU LÊN COMBOBOX\*/  private void UpLoadCombobox()  {  var loaiphg = BUS.BUS\_LoaiPHG.Select\_All();  cb\_loaiphong.DisplayMember = "TenloaiPHG";  cb\_loaiphong.ValueMember = "MaloaiPHG";  cb\_loaiphong.DataSource = loaiphg;  cb\_loaiphong.SelectedIndex = 0;  }  /\*PHƯƠNG THỨC BUTTON ĐỔ LẠI DỮ LIỆU LÊN DATAGRIDVIEW\*/  private void btn\_capnhat\_Click(object sender, EventArgs e)  {  UpLoadData();  UpdateButtonCapNhat(0);  }  private void UpdateButtonCapNhat(int sign)  {  if (sign == 0)  {  Count\_Update = 0;  btn\_capnhat.Text = "Cập nhật";  }  else  {  Count\_Update += 1;  btn\_capnhat.Text = $"({Count\_Update}) Cập nhật";  }  }  /\*PHƯƠNG THỨC BUTTON THÊM MỘT PHÒNG \*/  private void btn\_them\_Click(object sender, EventArgs e)  {  string tenphong = txt\_tenphong.Text;  string loaiphong = cb\_loaiphong.SelectedValue.ToString();  /\* Giả sử người sử dụng không nhập giá tiền thì sẽ mặc định gán dongia = 0 \*/  float dongia = float.TryParse(txt\_gia.Text, out dongia) ? dongia : 0;  bool result = BUS.BUS\_Phong.Insert(tenphong, loaiphong, dongia);  if (result)  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  UpdateButtonCapNhat(1);  }  else  {  MessageBox.Show("Thất Bại \nĐã có sự cố khi thêm phòng có thể là do 'tên phòng' đã trùng hoặc 'tên phòng' không hợp lệ xin hãy kiễm tra lại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  txt\_tenphong.Focus();  }  }  /\*THAO TÁC XÓA VÀ CẬP NHẬT DỮ LIỆU\*/  private void dgv\_phong\_CellClick(object sender, DataGridViewCellEventArgs e)  {  var reciver = (Guna2DataGridView)sender;  /\* KIỄM TRA XEM NGƯỜI DÙNG CÓ CLICK ĐÚNG DÒNG HAY CỘT TRONG DATAGRIDVIEW HAY KHÔNG \*/  if (e.RowIndex < 0 || e.RowIndex >= reciver.RowCount)  {  return;  }  if (e.ColumnIndex < 0 || e.ColumnIndex >= reciver.ColumnCount)  {  return;  }  /\*Kiễm tra click vào nut update\*/  if (reciver.Columns[e.ColumnIndex].Name == "Update")  {  var check = MessageBox.Show("Bạn có muốn cập nhật phòng này ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (check == DialogResult.Yes)  {  var phong = dgv\_phong.Rows[e.RowIndex];  var result = BUS.BUS\_Phong.Update((int)phong.Cells["Maphg"].Value, (string)phong.Cells["tenphong"].Value, (string)phong.Cells["Loai"].Value, (float)phong.Cells["Dongia"].Value);  if (result)  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  /\* THÔNG BÁO CHO NGƯỜI DÙNG BIẾT LÀ DATAGRIDVIEW ĐÃ CÓ SỰ THAY ĐỔI CẦN CLICK VÀO BUTTON CẬP NHẬT ĐỂ THẤY RÕ \*/  UpdateButtonCapNhat(1);  }  else  {  MessageBox.Show("Thất Bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  /\*Kiễm tra click vào nut Delete\*/  if (reciver.Columns[e.ColumnIndex].Name == "Delete")  {  var check = MessageBox.Show("Bạn có muốn xóa phòng này ?", "Thông báo", MessageBoxButtons.YesNo, MessageBoxIcon.Question);  if (check == DialogResult.Yes)  {  var maphong = (int)reciver.Rows[e.RowIndex].Cells["Maphg"].Value;  var result = BUS.BUS\_Phong.Delete(maphong);  if (result)  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  /\* THÔNG BÁO CHO NGƯỜI DÙNG BIẾT LÀ DATAGRIDVIEW ĐÃ CÓ SỰ THAY ĐỔI CẦN CLICK VÀO BUTTON CẬP NHẬT ĐỂ THẤY RÕ \*/  UpdateButtonCapNhat(1);  }  else  {  MessageBox.Show("Thất Bại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  }  }  }  }  }  } | | | |

1. **Form trung gian**

|  |
| --- |
| | **ĐỐI TƯỢNG** | **THUỘC TÍNH** | **CHỨC NĂNG** | | --- | --- | --- | | **Form** | **Name:** *ShowUserControl* | Chứa các usercontrol | | * **Code**   using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class ShowUserControl : Form  {  private bool dragging = false;  private Point CursorPoint;  private Point FormPoint;  public ShowUserControl()  {  InitializeComponent();  }  /\*HÀM THIẾT LẬP GÁN CÁC SỰ KIỆN ĐỂ THẢ SỰ KIỆN CHO CÁC USERCONTROL KHI ĐƯỢC ADD VÀO FORM \*/  public void SetSuKien(object b)  {  if(b is ThemDichVu)  {  ThemDichVu a = b as ThemDichVu;  a.panel\_toolbar.MouseDown += DropMouseDown;  a.panel\_toolbar.MouseMove += DropMouseMove;  a.panel\_toolbar.MouseUp += DropMouseUp;  }  if (b is ThemNhanVien)  {  ThemNhanVien a = b as ThemNhanVien;  a.panel\_toolbar.MouseDown += DropMouseDown;  a.panel\_toolbar.MouseMove += DropMouseMove;  a.panel\_toolbar.MouseUp += DropMouseUp;  }  if (b is ChiTietHoaDonControl)  {  ChiTietHoaDonControl a = b as ChiTietHoaDonControl;  a.panel\_toolbar.MouseDown += DropMouseDown;  a.panel\_toolbar.MouseMove += DropMouseMove;  a.panel\_toolbar.MouseUp += DropMouseUp;  }  if (b is DatPhong)  {  DatPhong a = b as DatPhong;  a.panel\_toolbar.MouseDown += DropMouseDown;  a.panel\_toolbar.MouseMove += DropMouseMove;  a.panel\_toolbar.MouseUp += DropMouseUp;  }  if (b is ChiTietPhong)  {  ChiTietPhong a = b as ChiTietPhong;  a.panel\_toolbar.MouseDown += DropMouseDown;  a.panel\_toolbar.MouseMove += DropMouseMove;  a.panel\_toolbar.MouseUp += DropMouseUp;  }  }  /\*SỰ KIỆN KHI MouseDown SẼ THẢ SỰ KIỆN MouseDown\*/  private void DropMouseDown(object? sender, MouseEventArgs e)  {  OnMouseDown(e);  }  /\*SỰ KIỆN KHI MouseMove SẼ THẢ SỰ KIỆN MouseMove\*/  private void DropMouseMove(object? sender, MouseEventArgs e)  {  OnMouseMove(e);  }  /\*SỰ KIỆN KHI MouseUp SẼ THẢ SỰ KIỆN MouseUp\*/  private void DropMouseUp(object? sender, MouseEventArgs e)  {  OnMouseUp(e);  }  /\* CHUỖI CÁC SỰ KIỆN DÙNG ĐỂ DI CHUYỂN FORM \*/  private void MouseDownn(object sender, MouseEventArgs e)  {  dragging = true;  CursorPoint = Cursor.Position;  FormPoint = this.Location;  }  private void MouseMovee(object sender, MouseEventArgs e)  {  if (dragging)  {  var differ = Point.Subtract(CursorPoint, new Size(Cursor.Position));  this.Location = Point.Subtract(FormPoint, new Size(differ));  }  }  private void MouseUpp(object sender, MouseEventArgs e)  {  dragging = false;  }  }  } | | | |

1. **Ngoài ra còn có các UserControl và Form hổ trợ**

| **RoomItem**   |  | | --- | | * **Code**   using BUS;  using DAO;  using DTO;  using Microsoft.Data.SqlClient;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Linq.Expressions;  using System.Net.NetworkInformation;  using System.Reflection.Metadata.Ecma335;  using System.Runtime.CompilerServices;  using System.Security.Cryptography.X509Certificates;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class RoomItem : UserControl  {    /\* get các controls trong roomitem thay vì sử dụng public \*/  public Label Getlbl\_gia => this.lbl\_gia;  public Label Getlbl\_loaiphong => this.lbl\_loaiphong;  public Label Getlbl\_sophong => this.lbl\_sophong;  public Label Getlbl\_trangthai => this.lbl\_trangthai;  public Guna.UI2.WinForms.Guna2Panel Getpanel\_container => panel\_container;  public Guna.UI2.WinForms.Guna2Button Getbtn\_trangthai => btn\_trangthai;  enum TrangThaiPhong  {  Empty = 1,  Lived = 2,  Booked = 3  }  /\* Constructor không tham số user control \*/  public RoomItem()  {  InitializeComponent();  }  /\* Constructor co tham so cua user control \*/  public RoomItem(int SoPhong, string LoaiPhong, string TrangThai, float Gia)  {  InitializeComponent();  /\* Chỉnh lại kích thước để hiện thị cho đẹp \*/  this.Margin = new Padding(50, 10, 20, 10);  if (LoaiPhong == "Phòng tổng thống")  {  this.lbl\_loaiphong.Location = new Point(12, 53);  }  /\* Gán giá trị \*/  this.lbl\_sophong.Text = SoPhong.ToString();  this.lbl\_loaiphong.Text = LoaiPhong;  this.lbl\_trangthai.Text = TrangThai;  this.lbl\_gia.Text = Gia.ToString("C0");    UpdateTrangThai(TrangThai);    }  /\* Cập nhật lại trạng thái của phòng (phòng trống, có khách, đã được đặt) \*/  public void UpdateTrangThai(string TrangThai)  {  ChiTietPhieuDatPhong chiTietphieudatphong = null;  ChiTietPhieuNhanPhong ct = null;  /\* Để lấy ngày nhận phòng hoặc ngày trả phòng \*/  if (!TrangThai.Contains("Phòng trống"))  {  int sophong = Convert.ToInt32(Getlbl\_sophong.Text);  /\* Từ mã phòng lấy danh sách ai đã từng đặt phòng này \*/  var list\_ctphieudatphong = BUS\_ChiTietDatPhong.Select\_By\_MaPHG(sophong);  /\* Lấy chitietdatphong mới nhất \*/  chiTietphieudatphong = list\_ctphieudatphong[list\_ctphieudatphong.Count - 1];  /\* Lấy phieudatphong \*/  var phieudatphong = BUS\_PhieuDatPhong.Select\_By\_MaDatPHG(chiTietphieudatphong.MaDatPHG);  /\* Lấy khachang\*/  var khachhang = BUS\_KhachHangcs.Select\_By\_ID((int)phieudatphong.MaKH);  /\* Lấy phiếu nhận phòng \*/  var phieunhanphong = BUS\_PhieuNhanPhong.Select\_By\_MaPhieuDat(phieudatphong.MaDatPHG);  if(TrangThai.Contains("khách")) ct = BUS\_ChiTietNhanPhong.TimKiem(phieunhanphong.MaNhanPHG, phieudatphong.MaDatPHG, khachhang.MaKH);  }  switch (TrangThai)  {  case "Phòng trống":  {  this.panel\_container.FillColor = Color.FromArgb(91, 164, 88);  this.btn\_trangthai.Image = Properties.Resources.bed;  this.btn\_trangthai.ImageSize = new Size(60, 57);  this.lbl\_trangthai.Location = new Point(167, 53);  this.lbl\_ngaynhan.Text = "";  break;  }  case "Có khách":  {  this.panel\_container.FillColor = Color.FromArgb(236, 97, 33);  this.btn\_trangthai.Image = Properties.Resources.sleeping;  this.btn\_trangthai.ImageSize = new Size(60, 57);  this.lbl\_trangthai.Location = new Point(186, 53);  this.lbl\_ngaynhan.Text = ct.NgayTraDuKien.ToString("D");  break;  }  case "Được đặt":  {  this.panel\_container.FillColor = Color.FromArgb(250, 163, 0);  this.btn\_trangthai.Image = Properties.Resources.calendar;  this.btn\_trangthai.ImageSize = new Size(30, 30);  this.lbl\_trangthai.Location = new Point(185, 53);  this.lbl\_ngaynhan.Text = chiTietphieudatphong.NgayNhan.ToString("D");  break;  }  }  this.lbl\_trangthai.Text = TrangThai;  }  /\* Cap nhat lai Phong \*/  private void UpdatePhong(TrangThaiPhong trangthai)  {  int sophong = Convert.ToInt32(Getlbl\_sophong.Text);  Phong phong = BUS\_Phong.Select\_By\_MaPHG(sophong);  bool result = true;  if (trangthai == TrangThaiPhong.Empty)  {  result = BUS\_Phong.Update\_With\_tinhtrang(phong.MaPHG, phong.TenPHG, phong.LoaiPHG, phong.Dongia,1);  }  else if (trangthai == TrangThaiPhong.Lived)  {  result = BUS\_Phong.Update\_With\_tinhtrang(phong.MaPHG, phong.TenPHG, phong.LoaiPHG, phong.Dongia,2);  }  else if (trangthai == TrangThaiPhong.Booked)  {  result = BUS\_Phong.Update\_With\_tinhtrang(phong.MaPHG, phong.TenPHG, phong.LoaiPHG, phong.Dongia,3);  }  }  /\* Hàm xử lý sự kiện click vào room  - Khi phòng đang trống thì mở UserControlDatPhong  - Còn phòng được đặt thì ConTrolChiTietPhong để tiến hành nhận phòng hoặc thêm dịch vụ  \*/  public void OnClickRoomItem(object sender, MouseEventArgs e)  {  var selectd\_item = (RoomItem)((Guna.UI2.WinForms.Guna2Panel)sender).Parent;  switch (selectd\_item.Getlbl\_trangthai.Text)  {  case "Phòng trống":  {  DatPhong();  break;  }  default:  {  ChiTietPhong();  break;  }  }    }  /\* Hàm dùng để đặt phòng\*/  public void DatPhong()  {  var result = MoControlDatPhong();  switch (result)  {  case DialogResult.OK:  {  UpdateTrangThai("Được đặt");  UpdatePhong(TrangThaiPhong.Booked);  break;  }  default:  {  break;  }  }  }  /\* Hàm dùng để xem chi tiết phòng\*/  public void ChiTietPhong()  {  var result = MoConTrolChiTietPhong();  switch (result)  {  case DialogResult.OK:  {  /\* Thay đổi tình trạng phòng\*/  UpdatePhong(TrangThaiPhong.Lived);  UpdateTrangThai("Có khách");  break;  }  case DialogResult.No:  {  /\* Thay đổi tình trạng phòng\*/  UpdatePhong(TrangThaiPhong.Empty);  UpdateTrangThai("Phòng trống");  break;  }  }  }  /\* Mở UserControlDatPhong để tiến hành đặt phòng  nếu trả về DialogResult.Oke là dặt thành công ngược lại thì không \*/  public DialogResult MoControlDatPhong()  {  var obj = new DatPhong();  obj.roomitem = this;  var form = new ShowUserControl();  /\*Các thuộc tính chỉnh kích thước form vừa đúng với UserControlDatPhong\*/  form.StartPosition = FormStartPosition.CenterScreen;  form.Size = obj.Size;  form.SetSuKien(obj);  form.Controls.Add(obj);  return form.ShowDialog();  }  /\* Mở UserConTrolChiTietPhong để tiến hành nhận phòng hoặc thêm các dịch vụ và xem thông tin chi tiêt \*/  public DialogResult MoConTrolChiTietPhong()  {  var obj = new ChiTietPhong();  obj.RoomItem = this;  obj.SetUp();  var form = new ShowUserControl();  /\*Các thuộc tính chỉnh kích thước form vừa đúng với UserConTrolChiTietPhong\*/  form.SetSuKien(obj);  form.StartPosition = FormStartPosition.CenterScreen;  form.Controls.Add(obj);  form.Size = obj.Size;  return form.ShowDialog();    }  }  } | |
| --- | --- | --- |
| **ThemNhanVienControl**   |  | | --- | | **Code**  using BUS;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Net;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class ThemNhanVien : UserControl  {  public delegate void ThemNhanVienEventHandler();  public ThemNhanVien()  {  InitializeComponent();  UpLoadComboBoxLoaiNV();  }  /\* Upload loại nhân viên lên combobox \*/  public void UpLoadComboBoxLoaiNV()  {  var data = BUS.BUS\_LoaiNV.Select\_All();  cbb\_loainv.DataSource = data;  cbb\_loainv.DisplayMember = "TenLoai";  cbb\_loainv.ValueMember = "MaLoai";  cbb\_loainv.SelectedIndex = 0;  }  /\* khi muốn thoát form đăng ký \*/  private void btn\_thoat\_Click(object sender, EventArgs e)  {  var choice = MessageBox.Show("Bạn có muốn thoát", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);  if (choice == DialogResult.OK) this.Close(DialogResult.Cancel);  }  /\* khi click vào nút ĐĂNG KÝ \*/  private void btn\_themnhanvien\_Click(object sender, EventArgs e)  {  if (Insert())  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  this.Close(DialogResult.Yes);  }  else MessageBox.Show(@"Đã có sự cố khi thêm nhân viên  Hãy kiễm tra lại Tên và SĐT và Địa chỉ", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);    }  /\*Hàm thêm nhân viên\*/  public bool Insert()  {  /\* Lấy dữ liệu và gán cho các biến \*/  string tennv = txt\_tennv.Text;  string sdt = txt\_sdt.Text;  string diachi = txt\_diachi.Text;  int loainv = (int)cbb\_loainv.SelectedValue;  string gioitinh = rdb\_nam.Checked ? "Nam" : "Nữ";  /\* Thực hiện thêm nhân viên \*/  var result = BUS\_NhanVien.Insert(tennv,sdt,diachi,gioitinh,loainv);  return result;  }  }  } | |
| **ThemDichVuControl**   |  | | --- | | * **Code**   using BUS;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Text.RegularExpressions;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class ThemDichVu : UserControl  {  public ThemDichVu()  {  InitializeComponent();  UpLoadComBoBox();  }  /\* ĐỔ DỮ LIỆU LÊN COMBOBOX \*/  public void UpLoadComBoBox()  {  var source = from p in BUS\_LoaiDV.Select\_All()  select p;  cbb\_loaidv.DataSource = source.ToList();  cbb\_loaidv.DisplayMember = "TenLoaiDV";  cbb\_loaidv.ValueMember = "MaLoaiDV";  cbb\_loaidv.SelectedIndex = 0;  }  private void btn\_thoat\_Click(object sender, EventArgs e)  {  var choice = MessageBox.Show("Bạn có muốn thoát", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);  if (choice == DialogResult.OK) this.Close(DialogResult.Cancel);  }  private void btn\_them\_Click(object sender, EventArgs e)  {  if (Insert())  {  MessageBox.Show("Thành công!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);  this.Close(DialogResult.Yes);  }  else MessageBox.Show(@"Đã có sự cố khi thêm dịch vụ  Hãy kiễm tra lại Tên dịch vụ có trùng không", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);  }  /\*Hàm thêm dịch vụ\*/  public bool Insert()  {  /\* Lấy dữ liệu và gán cho các biến \*/  string tendv = txt\_tendichvu.Text;  int loainv = (int)cbb\_loaidv.SelectedValue;  /\* Đảm bảo giá trị của đơn giá là không thể bị null\*/  float dongia = float.TryParse(txt\_dongia.Text, out dongia) ? dongia : 0;  /\* Thực hiện thêm nhân viên \*/  var result = BUS\_DichVu.Insert(tendv, loainv, dongia);  return result;  }  /\* Chỉnh hover cho button thoát \*/  private void btn\_thoat\_MouseEnter(object sender, EventArgs e)  {  btn\_thoat.FillColor = Color.Red;  }  private void btn\_thoat\_MouseLeave(object sender, EventArgs e)  {  btn\_thoat.FillColor = Color.FromArgb(58, 61, 76);  }  /\* Hàm nhập đơn giá nó sẽ tự động hiển thị kiểu tiện tệ \*/  private void txt\_dongia\_TextChanged(object sender, EventArgs e)  {  string text = txt\_dongia.Text.ToString();  /\* Kiễm tra chuỗi đang rỗng \*/  if (text.Length == 0) return;  /\* Kiễm tra người dùng có chỉ nhập số hay không \*/  if (!Regex.IsMatch(text, "^[0-9]"))  {  MessageBox.Show("Vui Lòng Nhập Số");  return;  }  }  }  } | |
| **DatPhongControl**   |  | | --- | | *    **Code**    using BUS;  using DTO;  using Microsoft.EntityFrameworkCore.Query;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class DatPhong : UserControl  {  /\* tạo delegate để tryền dữ liệu cho RoomItem \*/  public RoomItem roomitem;  public DatPhong()  {  InitializeComponent();  /\* Mặc định giới tính là nam \*/  cbb\_gioitinh.SelectedIndex = 0;  }  /\*Hàm Click vào btn hủy đặt phòng\*/  private void btn\_huy\_Click(object sender, EventArgs e)  {  this.Close(DialogResult.Cancel);  }  /\*Hàm Click vào đặt phòng \*/  private void btn\_datphong\_Click(object sender, EventArgs e)  {  /\* Lấy thông tin \*/    string hoten = txt\_diachi.Text;  string cccd = txt\_cccd.Text;  string sdt = txt\_sdt.Text;  string diachi = txt\_diachi.Text;  string gioitinh = cbb\_gioitinh.SelectedItem as string;  DateTime ngaydat = DateTime.Parse(DateTime.Now.ToString("D"));  DateTime ngaynhan = dtp\_ngaynhan.Value;  int maphong = Convert.ToInt32(roomitem.Getlbl\_sophong.Text);    /\* Lấy hoặc tạo khách hàng \*/  KhachHang kh = BUS\_KhachHangcs.TimKhachHang\_OrCreate(hoten, cccd, sdt, diachi, gioitinh);    if (kh == null)  {  MessageBox.Show(@"Thất bại  Có lỗi khi tạo đặt phòng có thể do hoten để trống hoặc CCCD đã tồn tại", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  return;  }  /\* Tạo PhieuDatPhong \*/  BUS\_PhieuDatPhong.Insert(kh.MaKH);  /\* Lấy PhieuDatPhong \*/  PhieuDatPhong phieudatphg = BUS\_PhieuDatPhong.Select\_By\_MaKH\_Last(kh.MaKH);  /\* Tạo ChitietPhieuDatPhong \*/  BUS\_ChiTietDatPhong.Insert(maphong,phieudatphg.MaDatPHG,ngaydat, ngaynhan);    this.Close(DialogResult.OK);  }  /\*Hàm Click vào nút thoát\*/  private void btn\_thoat\_Click(object sender, EventArgs e)  {  var choice = MessageBox.Show("Bạn có muốn thoát", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);  if (choice == DialogResult.OK) { this.Close(DialogResult.Cancel); }  }  }  } | |
| **ChiTietPhongControl**   |  | | --- | | *    **Code**    using BUS;  using DAO;  using DTO;  using Guna.UI2.WinForms;  using Microsoft.EntityFrameworkCore.Metadata.Internal;  using System;  using System.CodeDom;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Runtime.CompilerServices;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class ChiTietPhong : UserControl  {  Color color;  public RoomItem RoomItem { get; set; }  ChiTietPhieuDatPhong chiTietphieudatphong;  PhieuDatPhong phieudatphong;  PhieuNhanPhong phieunhanphong;  KhachHang khachhang;  Phong phong;  HoaDon hoadon = null;  NhanVien nhanvien;  private readonly int PhuThu = 10;  public void SetUp()  {  int maphong = Convert.ToInt32(RoomItem.Getlbl\_sophong.Text);  /\* Từ mã phòng lấy danh sách ai đã từng đặt phòng này \*/  var list\_ctphieudatphong = BUS\_ChiTietDatPhong.Select\_By\_MaPHG(maphong);  /\* Lấy khách hàng đặt phòng mới nhất mới nhất \*/  chiTietphieudatphong = list\_ctphieudatphong[list\_ctphieudatphong.Count - 1];  /\* Lấy phieudatphong \*/  phieudatphong = BUS\_PhieuDatPhong.Select\_By\_MaDatPHG(chiTietphieudatphong.MaDatPHG);  /\* Lấy khachang\*/  khachhang = BUS\_KhachHangcs.Select\_By\_ID((int)phieudatphong.MaKH);  /\* Lấy phiếu nhận phòng \*/  phieunhanphong = BUS\_PhieuNhanPhong.Select\_By\_MaPhieuDat(phieudatphong.MaDatPHG);  /\* Lấy phòng \*/  phong = BUS\_Phong.Select\_By\_MaPHG(maphong);  /\* Load dữ liệu về dịch vụ lên datagrid dịch vụ \*/  LoadDichVu();  /\* Load nhân viên \*/  LoadNhanVien();  /\* Load dữ liệu về phòng ngày đặt ngày trả \*/  LoadThongTinPhong();  }  public ChiTietPhong()  {  InitializeComponent();  }  /\* Hàm tải danh sách dịch vụ lên \*/  public void LoadDichVu()  {  var source = BUS\_DichVu.Select\_All();  dgv\_dichvu.AutoGenerateColumns = false;  dgv\_dichvu.DataSource = source;  }  public void LoadNhanVien()  {  var source = BUS\_NhanVien.Select\_LeTan();  cbb\_nhanvien.DataSource = source;  cbb\_nhanvien.DisplayMember = "TenNV";  cbb\_nhanvien.ValueMember = "Manv";  }  public void LoadThongTinPhong()  {  txt\_sophong.Text = RoomItem.Getlbl\_sophong.Text;  btn\_loaiphong.Text = RoomItem.Getlbl\_loaiphong.Text;  txt\_ngaylap.Text = chiTietphieudatphong.NgayDat.ToString("D");  dtp\_ngaybatdau.Value = chiTietphieudatphong.NgayNhan;  dtp\_ngaytraphong.Value = chiTietphieudatphong.NgayNhan;  /\* Khách hàng chưa có phiếu nhận phòng (phòng vừa mới đặt) [Click lần 1] \*/  if (phieunhanphong == null)  {  /\* Tạo phiếu nhận phòng \*/  BUS\_PhieuNhanPhong.Insert(phieudatphong.MaDatPHG, khachhang.MaKH);  phieunhanphong = BUS\_PhieuNhanPhong.Select\_By\_MaPhieuDat(phieudatphong.MaDatPHG);  txt\_nhanvien.Hide();  btn\_thanhtoan.Hide();  btn\_thoat\_2.Show();  }  else  {  /\* Hiển thị danh sách sử dụng dịch vụ theo mã nhận phòng \*/  UpdateBangSuDungDichVu(-1);  /\* Kiễm tra khách đã nhận phòng chưa [Click lần 2] ( Có phiếu nhận phòng nhưng chua lấy ) \*/  ChiTietPhieuNhanPhong ct = BUS\_ChiTietNhanPhong.TimKiem(phieunhanphong.MaNhanPHG, phieudatphong.MaDatPHG, khachhang.MaKH);  btn\_thanhtoan.Hide();  btn\_thoat\_2.Show();  /\* Khách hàng đã có phiếu nhận phòng nhưng chưa lấy phòng \*/  if (ct == null) return;  /\* Khách hàng đã lấy phòng \*/  hoadon = BUS\_HoaDon.Select\_By\_MaNhanPHG(phieunhanphong.MaNhanPHG);  nhanvien = BUS\_NhanVien.Select\_By\_MaNV((int)hoadon.MaNV);  /\* Cập nhật hiển thị \*/  txt\_nhanvien.Text = nhanvien.TenNV;  txt\_nhanvien.Show();  cbb\_nhanvien.Hide();  txt\_mathue.Text = ct.MaNhanPHG.ToString();  dtp\_ngaybatdau.Value = ct.NgayNhan;  dtp\_ngaybatdau.Enabled = false;  dtp\_ngaytraphong.Value = ct.NgayTraDuKien;  dtp\_ngaytraphong.Enabled = false;  /\* Khách thanh toán mới được trả phòng \*/  btn\_thoat\_2.Hide();  btn\_thanhtoan.Show();  UpdateBangThanhTien();  }  }  private void btn\_thoat\_2\_Click(object sender, EventArgs e)  {  /\* Nếu khách hàng muốn bỏ đặt thì xóa phiếu nhận phòng và xóa phiếu đặt phòng \*/  BUS\_PhieuNhanPhong.Delete\_By\_MaDatPHG(phieudatphong.MaDatPHG);  BUS\_PhieuDatPhong.Delete(phieudatphong.MaDatPHG);  this.Close(DialogResult.No);  }  private void btn\_thoat\_Click(object sender, EventArgs e)  {  var choice = MessageBox.Show("Bạn có muốn thoát", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);  if (choice == DialogResult.OK) this.Close(DialogResult.Cancel);  }  private void btn\_capnhat\_Click(object sender, EventArgs e)  {  /\* Kiễm tra khách hàng đã lấy phòng chưa [đã có phiếu nhận phòng] \*/  /\* Có phiếu nhận phòng nhưng lấy nhận phòng \*/  if (hoadon == null)  {  /\* Tạo chi tiết nhận phòng khi khách hàng lấy phòng \*/  BUS\_ChiTietNhanPhong.Insert(phieunhanphong.MaNhanPHG, phieudatphong.MaDatPHG, khachhang.MaKH, dtp\_ngaybatdau.Value, dtp\_ngaytraphong.Value, DateTime.MinValue);  /\* Tạo hóa đơn khi khách nhận phòng \*/  int manv = (int)cbb\_nhanvien.SelectedValue;  /\* tạo hóa đơn nhưng chưa bao gồm tổng tiền \*/  BUS\_HoaDon.Insert(manv, khachhang.MaKH, phieunhanphong.MaNhanPHG, dtp\_ngaybatdau.Value);  hoadon = BUS\_HoaDon.Select\_By\_MaNhanPHG(phieunhanphong.MaNhanPHG);  /\* Tạo chi tiết hóa đơn khi khách nhận phòng \*/  /\* Lấy danh sách sử dụng dịch vụ của phòng này \*/  var list\_sddv = BUS\_SuDungDichVu.Select\_By\_MaNhanPHG(hoadon.MaNhanPHG);  /\* Thêm các chi tiết hóa đơn \*/  list\_sddv.ForEach(sd =>  {  float tiendv = BUS\_SuDungDichVu.GetTongTien\_By\_MaSDDV(sd.MaSuDungDV);  int songay = DAO\_HoaDon.GetSoNgay(hoadon.MaHD);  BUS\_ChiTietHoaDon.Insert(hoadon.MaHD, phong.MaPHG, sd.MaSuDungDV, phong.Dongia, tiendv, songay);  });  var data = BUS\_ChiTietHoaDon.Select\_All\_By\_MaHD(hoadon.MaHD);  /\* Cập nhật lại tổng tiền của hóa đơn sau khi có chi tiết hóa đơn    tiền cùa chi tiết hóa đơn chưa bao gồm tiền phòng và đã có phụ thu 10%  tiền của hóa đơn sẽ là : (tiền phòng \* số ngày) + tiền dịchvụ  \*/  float tiendv = BUS\_SuDungDichVu.GetTongTien\_By\_MaNhanPHG\_SauPhuThu(phieunhanphong.MaNhanPHG);  BUS.BUS\_HoaDon.Update\_TongTien(phieunhanphong.MaNhanPHG, phong.MaPHG, tiendv);  UpdateBangThanhTien();  }  else  {  /\* Có phiếu nhận phòng và đã nhận phòng rồi \*/  /\* Cập nhật lại các thay đổi có thể có \*/  /\* Thay đổi về ngày trả phòng \*/  /\* Thay đổi về sử dụng dịch vụ\*/  /\* [Không cần phải sửa bởi vì nó đã được xử lý trước rồi] \*/  }  this.Close(DialogResult.OK);  }  private void btn\_thanhtoan\_Click(object sender, EventArgs e)  {  /\* Cập nhật lại ngày trả thực tế cho chi tiết nhận phòng \*/  if (hoadon != null)  {  BUS\_ChiTietNhanPhong.Update\_NgayTraThucTe(phieunhanphong.MaNhanPHG, phieudatphong.MaDatPHG, khachhang.MaKH, DateTime.Now);  /\* Mở form in hóa đơn lên \*/  Inhoadon form = new Inhoadon();  form.MaHD = hoadon.MaHD;  form.LoadData();  form.StartPosition = FormStartPosition.CenterScreen;  DialogResult kq = form.ShowDialog();  if(kq == DialogResult.OK)this.Close(DialogResult.No);  }  }  /\* Hàm thêm dịch vụ \*/  private void dgv\_dichvu\_CellClick(object sender, DataGridViewCellEventArgs e)  {  var reciver = (Guna2DataGridView)sender;  if (e.RowIndex < 0 || e.RowIndex >= reciver.RowCount)  {  return;  }  if (e.ColumnIndex < 0 || e.ColumnIndex >= reciver.ColumnCount)  {  return;  }  /\*Kiễm tra click vào button thêm\*/  if (reciver.Columns[e.ColumnIndex].Name == "Them")  {  int madv\_dachon = Convert.ToInt32(reciver.Rows[e.RowIndex].Cells["MaDV"].Value.ToString());  int maphong = Convert.ToInt32(RoomItem.Getlbl\_sophong.Text);  /\* Lấy mã SDDV \*/  SuDungDV sudungdichvu = BUS\_SuDungDichVu.Select\_By\_MaDV\_MaNhanPHG(madv\_dachon, phieunhanphong.MaNhanPHG);  /\* Kiễm tra xem là dịch vụ này được đặt chưa \*/  if (sudungdichvu == null)/\* Chưa đặt \*/  {  BUS\_SuDungDichVu.Insert(madv\_dachon, phieunhanphong.MaNhanPHG, 1);  /\* Kiễm tra xem đã có hóa đơn chưa \*/  if (hoadon != null)/\* Nếu đã có rồi thì cập nhật lên chi tiêt hóa đơn rồi cập nhật lại tổng tiền của hóa đơn \*/  {  /\* Lấy Sudungdichvu vừa mới thêm vào \*/  var SuDungDichVu = BUS\_SuDungDichVu.Select\_By\_MaNhanPHG\_Last\_Element(hoadon.MaNhanPHG);  /\* Lấy số ngày và tiền dv \*/  int songay = BUS\_HoaDon.GetSoNgay(hoadon.MaHD);  float tiendv = BUS\_DichVu.GetDichVu\_By\_MaDV(madv\_dachon).Dongia;  /\* Thêm chi tiết hóa đơn \*/  BUS\_ChiTietHoaDon.Insert(hoadon.MaHD, phong.MaPHG, (int)SuDungDichVu.MaSuDungDV, phong.Dongia, tiendv, songay);  /\* Cập nhật lại hóa đơn \*/  tiendv = BUS\_SuDungDichVu.GetTongTien\_By\_MaNhanPHG\_SauPhuThu(phieunhanphong.MaNhanPHG);  BUS.BUS\_HoaDon.Update\_TongTien(hoadon.MaNhanPHG, phong.MaPHG, tiendv);  UpdateBangThanhTien();  }  }  else  {  /\* Cập nhật lại số lượng \*/  BUS\_SuDungDichVu.Update\_SoLuong(sudungdichvu.MaSuDungDV, phieunhanphong.MaNhanPHG, 1);  /\* Nếu khách đã lấy phòng rồi thì phải cập lại vào hóa đơn và chi tiết hóa đơn \*/  if (hoadon != null)  {  /\* Cập nhật lại chi tiết hóa đơn của mã sử dụng dịch vụ đó\*/  BUS\_ChiTietHoaDon.Update\_TienDV(hoadon.MaHD, sudungdichvu.MaSuDungDV);  /\* Cập nhật lại hóa đơn\*/  float tiendv = BUS\_SuDungDichVu.GetTongTien\_By\_MaNhanPHG\_SauPhuThu(phieunhanphong.MaNhanPHG);  BUS.BUS\_HoaDon.Update\_TongTien(hoadon.MaNhanPHG, phong.MaPHG, tiendv);  UpdateBangThanhTien();  }  }  UpdateBangSuDungDichVu(-1);  }  }  /\* Cập nhật những dữ liệu đã được sử dụng lên datagridview \*/  private void UpdateBangSuDungDichVu(int index)  {  var data\_source = BUS\_SuDungDichVu.Select\_Display\_MaNhanPHG(phieunhanphong.MaNhanPHG);  dgv\_dicvudasudung.AutoGenerateColumns = false;  /\*Xóa hết dự liệu rồi cập nhật lại datagridview\*/  dgv\_dicvudasudung.DataSource = data\_source;  /\* Chỉnh lại dòng được chọn là rỗng \*/  dgv\_dicvudasudung.ClearSelection();  if (index >= 0 && index < dgv\_dicvudasudung.Rows.Count)  {  dgv\_dicvudasudung.Rows[index].Selected = true;  }  else  {  dgv\_dicvudasudung.ClearSelection();  }  }  /\* Hàm Xóa bớt dịch vụ \*/  private void dgv\_dicvudasudung\_CellClick(object sender, DataGridViewCellEventArgs e)  {  var reciver = (Guna2DataGridView)sender;  if (e.RowIndex < 0 || e.RowIndex >= reciver.RowCount)  {  return;  }  if (e.ColumnIndex < 0 || e.ColumnIndex >= reciver.ColumnCount)  {  return;  }  /\*Kiễm tra click vào button delete\*/  if (reciver.Columns[e.ColumnIndex].Name == "Delete")  {  int madv = Convert.ToInt32(reciver.Rows[e.RowIndex].Cells["MaDV1"].Value.ToString());  /\* Lấy mã SDDV vừa được chọn để xóa \*/  SuDungDV sudungdichvu = BUS\_SuDungDichVu.Select\_By\_MaDV\_MaNhanPHG(madv, phieunhanphong.MaNhanPHG);  /\* Kiễm tra khách lấy phòng chưa \*/  if (hoadon == null)  {  /\* Cập nhật lại số lượng bình thường không cần phải cập nhật lại hóa đơn \*/  BUS\_SuDungDichVu.Update\_SoLuong(sudungdichvu.MaSuDungDV, phieunhanphong.MaDatPHG, -1);  }  else  {  BUS\_SuDungDichVu.Update\_SoLuong(sudungdichvu.MaSuDungDV, hoadon.MaNhanPHG, -1);  UpdateBangThanhTien();  }  /\* Console.WriteLine("/n/n/n---------Danh sach su dung dich vu--------/n/n/n");  var dt1 = BUS\_SuDungDichVu.Select\_By\_MaNhanPHG(phieunhanphong.MaNhanPHG);  dt1.ToList().ForEach(x => x.Print());  if (hoadon != null)  {  Console.WriteLine("/n/n/nDanh sach chi tiet hoa don/n/n/n");  var dt2 = BUS\_ChiTietHoaDon.Select\_All\_By\_MaHD(hoadon.MaHD);  dt2.ToList().ForEach(x => x.Print());  }\*/  UpdateBangSuDungDichVu(e.RowIndex);  }  }  /\* Tới hạn nhưng khách chưa trả phòng \*/  private void dtp\_ngaytraphong\_ValueChanged(object sender, EventArgs e)  {  /\* Nếu ngày trả phòng nhỏ hơn ngày nhận thì báo lỗi\*/  if (hoadon == null)  {  var dtp = (Guna2DateTimePicker)sender;  DateTime ngaykt = DateTime.Parse(dtp.Value.ToString("D"));  DateTime ngaybd = DateTime.Parse(dtp\_ngaybatdau.Value.ToString("D"));  if (ngaykt < ngaybd)  {  MessageBox.Show("Ngày trả phòng không thể bé hơn ngày nhận", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Error);  dtp\_ngaytraphong.Value = chiTietphieudatphong.NgayNhan;  }  }  if (hoadon != null)  {  /\* Kiễm tra tới hạn trả mà khách chưa trả phòng \*/  ChiTietPhieuNhanPhong ct = BUS\_ChiTietNhanPhong.TimKiem(phieunhanphong.MaNhanPHG, phieudatphong.MaDatPHG, khachhang.MaKH);  DateTime ngayhientai = DateTime.Parse(DateTime.Now.ToString("D"));  DateTime ngaytradukien = DateTime.Parse(dtp\_ngaytraphong.Value.ToString("D"));  if (ngaytradukien >= ngayhientai)  {  /\* Khách trả phòng trước hoặc đúng hạn \*/  lbl\_traphong.ForeColor = Color.Green;  }  else  {  /\* Khách trả phòng quá hạn \*/  lbl\_traphong.ForeColor = Color.Red;  }  }  }  private void dtp\_ngaybatdau\_ValueChanged(object sender, EventArgs e)  {  if (hoadon == null)  {  DateTime day1 = DateTime.Parse(dtp\_ngaybatdau.Value.ToString("D"));  DateTime day2 = DateTime.Parse(chiTietphieudatphong.NgayNhan.ToString("D"));  if (day1 <= day2)  {  /\* Khách nhận phòng trước hoặc đúng hạn \*/  lbl\_nhanphong.ForeColor = Color.Green;  }  else  {  /\* Khách nhận phòng quá hạn\*/  lbl\_nhanphong.ForeColor = Color.Red;  }  color = lbl\_nhanphong.ForeColor;  }  else  {  lbl\_nhanphong.ForeColor = color;  }  }  /\* Hiển thị liên quan đến tiền \*/  public void UpdateBangThanhTien()  {  if (hoadon != null)  {  txt\_tien\_giaphong.Text = string.Format($"{phong.Dongia:C0} x {BUS\_HoaDon.GetSoNgay(hoadon.MaHD)}");  txt\_tien\_dichvu.Text = string.Format($"{BUS\_SuDungDichVu.GetTongTien\_By\_MaNhanPHG(phieunhanphong.MaNhanPHG):C0}");  txt\_tien\_phuthu.Text = PhuThu + "%";  float tiendv = BUS\_SuDungDichVu.GetTongTien\_By\_MaNhanPHG(phieunhanphong.MaNhanPHG);  float tongtien = BUS\_HoaDon.GetTongTienChuaPhuThu(phieunhanphong.MaNhanPHG, phong.MaPHG, tiendv);  txt\_tien\_tongtien.Text = string.Format($"{tongtien:C0}");  txt\_tien\_phaitra.Text = string.Format($"{hoadon.TongTien:C0}");  }  }  }  } | |
| **ChiTietHoaDonControl**   |  | | --- | | *    **Code**    using BUS;  using DTO;  using Guna.UI2.WinForms;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Linq;  using System.Linq.Expressions;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class ChiTietHoaDonControl : UserControl  {  public int MaHD;  public ChiTietHoaDonControl()  {  InitializeComponent();  }  /\* ĐỔ DỮ LIỆU LÊN \*/  public void UpLoadData()  {  HoaDon hd = BUS\_HoaDon.Select\_By\_MaHD(MaHD);  ChiTietHoaDon ct\_hoadon = BUS\_ChiTietHoaDon.Select\_By\_MaHD(MaHD);  NhanVien nv = BUS\_NhanVien.Select\_By\_MaNV((int)hd.MaNV);  ChiTietPhieuNhanPhong ct\_nhanphg = BUS\_ChiTietNhanPhong.Select\_By\_MaNhanPhong(hd.MaNhanPHG);  ChiTietPhieuDatPhong ct\_datphg = BUS\_ChiTietDatPhong.Select\_By\_MaDatPHG(ct\_nhanphg.MaDatPHG);  Phong phong = BUS\_Phong.Select\_By\_MaPHG((int)ct\_hoadon.MaPHG);  LoaiPHG loaiphg = BUS\_LoaiPHG.Select\_By\_MaLoai(phong.LoaiPHG);  var ds\_sudungdv = BUS\_SuDungDichVu.Select\_Display\_MaNhanPHG(ct\_nhanphg.MaNhanPHG);  string MaNhanPHG = hd.MaNhanPHG + "";  txt\_sophong.Text = phong.TenPHG + "";  txt\_mathue.Text = hd.MaNhanPHG + "";  btn\_loaiphong.Text = loaiphg.TenloaiPHG;  txt\_nhanvien.Text = nv.TenNV;  txt\_ngaylap.Text = ct\_datphg.NgayDat.ToString("D");  txt\_nhanphong.Text = ct\_nhanphg.NgayNhan.ToString("D");  txt\_traphong.Text = ct\_nhanphg.NgayTraThucTe.ToString("D");  dgv\_dicvudasudung.AutoGenerateColumns = false;  dgv\_dicvudasudung.DataSource = ds\_sudungdv;  }  private void btn\_thoat\_Click(object sender, EventArgs e)  {  var choice = MessageBox.Show("Bạn có muốn thoát", "Thông báo", MessageBoxButtons.OKCancel, MessageBoxIcon.Question);  if (choice == DialogResult.OK) this.Close(DialogResult.Cancel);  }  }  } | |
| **InhoadonForm**   |  | | --- | | *    **Code**    using BUS;  using DTO;  using System;  using System.Collections.Generic;  using System.ComponentModel;  using System.Data;  using System.Drawing;  using System.Drawing.Printing;  using System.Linq;  using System.Text;  using System.Threading.Tasks;  using System.Windows.Forms;  namespace GUI  {  public partial class Inhoadon : Form  {  public int MaHD = 0;  private bool dragging = false;  private Point CursorPoint;  private Point FormPoint;  public Inhoadon()  {  InitializeComponent();  }  private void btn\_thoat\_Click(object sender, EventArgs e)  {  this.Close();  }  private void Print()  {  getprintarea();  printPreviewDialog1.Document = printDocument1;  printDocument1.PrintPage += new PrintPageEventHandler(printDocument1\_PrintPage);  printPreviewDialog1.ShowDialog();  }  private void printDocument1\_PrintPage(object sender, PrintPageEventArgs e)  {  Rectangle pagearea = e.MarginBounds;  // Vẽ header ở giữa trên trang  int headerX = (pagearea.Width) / 2 - (pagearea.Width / 2) / 2;  int headerY = pagearea.Top;  e.Graphics.DrawImage(header, headerX, headerY);  // Vẽ body ở giữa trang  int bodyX = (pagearea.Width) / 2 - (pagearea.Width / 2) / 2 + 20;  int bodyY = headerY + header.Height - 100;  e.Graphics.DrawImage(body, bodyX, bodyY);  // Vẽ footer ở giữa dưới trang  int minus = 0;  if (dgv.Rows.Count > 5) minus = 40;  else minus = 50;  int footerX = (pagearea.Width) / 2 - (pagearea.Width / 2) / 2;  int footerY = bodyY + body.Height / 2 - minus;  e.Graphics.DrawImage(footer, footerX, footerY);  }  private Bitmap header;  private Bitmap body;  private Bitmap footer;  private string row\_header;  private string row\_data;  private void getprintarea()  {  /\* Vẽ hình footer \*/  header = new Bitmap(this.panel\_hoadon1.Width, this.panel\_hoadon1.Height);  this.panel\_hoadon1.DrawToBitmap(header, new Rectangle(0, 0, this.panel\_hoadon1.Width, this.panel\_hoadon1.Height));  /\* Vẽ hình body \*/  int Height = dgv.Height;  dgv.Height = dgv.Rows.Count \* dgv.RowTemplate.Height \* 2;  body = new Bitmap(this.dgv.Width, dgv.Height);  this.dgv.DrawToBitmap(body, new Rectangle(0, 0, this.dgv.Width, dgv.Height));  dgv.Height = Height;  /\* Vẽ hình footer \*/  footer = new Bitmap(this.panel\_hoadon2.Width, this.panel\_hoadon2.Height);  this.panel\_hoadon2.DrawToBitmap(footer, new Rectangle(0, 0, this.panel\_hoadon2.Width, this.panel\_hoadon2.Height));  }  private void btn\_in\_Click(object sender, EventArgs e)  {  Print();  this.DialogResult = DialogResult.OK;  this.Close();  }  private void Inhoadon\_Load(object sender, EventArgs e)  {  }  public void LoadData()  {  dgv.AutoGenerateColumns = false;  dgv.ReadOnly = true;  if (MaHD != 0)  {  HoaDon hd = BUS\_HoaDon.Select\_By\_MaHD(MaHD);  NhanVien nv = BUS\_NhanVien.Select\_By\_MaNV((int)hd.MaNV);  ChiTietPhieuNhanPhong ct\_phieunhan = BUS\_ChiTietNhanPhong.Select\_By\_MaNhanPhong((int)hd.MaNhanPHG);  ChiTietPhieuDatPhong ct\_phieudat = BUS\_ChiTietDatPhong.Select\_By\_MaDatPHG(ct\_phieunhan.MaDatPHG);  Phong phong = BUS\_Phong.Select\_By\_MaPHG((int)ct\_phieudat.MaPHG);  LoaiPHG loaiphg = BUS\_LoaiPHG.Select\_By\_MaLoai(phong.LoaiPHG);  KhachHang kh = BUS\_KhachHangcs.Select\_By\_ID((int)hd.MaKH);  var datasource = BUS\_SuDungDichVu.Select\_Display\_MaNhanPHG(hd.MaNhanPHG);  txt\_tenkh.Text = kh.TenKH;  txt\_ngaylap.Text = ct\_phieudat.NgayDat.ToString("D");  txt\_mahd.Text = hd.MaHD.ToString();  txt\_nhanvien.Text = nv.TenNV;  txt\_sophong.Text = phong.MaPHG.ToString();  txt\_loaiphong.Text = loaiphg.TenloaiPHG;  txt\_songay.Text = BUS\_HoaDon.GetSoNgay(hd.MaHD).ToString();  dgv.DataSource = datasource;  dgv.ClearSelection();  txt\_tongtien.Text = String.Format($"{hd.TongTien:C0}");  lbl\_ngaylaphoadon.Text = DateTime.Now.ToString("D");  }  }  private void MouseDownn(object sender, MouseEventArgs e)  {  dragging = true;  CursorPoint = Cursor.Position;  FormPoint = this.Location;  }  private void MouseMovee(object sender, MouseEventArgs e)  {  if (dragging)  {  var differ = Point.Subtract(CursorPoint, new Size(Cursor.Position));  this.Location = Point.Subtract(FormPoint, new Size(differ));  }  }  private void MouseUpp(object sender, MouseEventArgs e)  {  dragging = false;  }  }  } | |

1. **Kết luận và hướng phát triển:**

* **Kết luận**
* Chương trình có giao diện thân thiện, thuận tiện cho người sử dụng.
* Hỗ trợ nhiều chức năng phù hợp với thực tế của việc tìm kiếm thông tin, trao đổi
* Dễ dàng sử dụng và quản lý.
* Một số chức năng còn chưa hoạt động tối ưu
* Chưa giải quyết triệt để các vấn đề nảy sinh trong quá trình quản lý.
* **Hướng phát triển của phần mềm :**
* Tìm hiểu và hoàn thiện một số chức năng chưa thực hiện trong ứng dụng.
* Hỗ trợ khả năng bảo mật, tốc độ truy vấn, khả năng quản lý dữ liệu tốt hơn.

1. **Tài liệu tham khảo**

* [**https://stackoverflow.com/**](https://stackoverflow.com/)
* [**https://www.google.com/**](https://www.google.com/)
* <https://fr.slideshare.net/nHngdn/quan-ly-khach-san-cshare>
* <https://www.youtube.com/watch?v=zoGmnMNG0pM&t=408s>
* <https://www.youtube.com/watch?v=7rnPI7K-Ow8>