**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI**

**KHOA CÔNG NGHỆ THÔNG TIN**

---o0o---

****

**BÁO CÁO BÀI TẬP LỚN**

**MÔN: LẬP TRÌNH WINDOWS**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM**

**QUẢN LÝ KHÁCH SẠN VÀ ĐẶT PHÒNG**

**Giảng viên hướng dẫn:** Ths. Đỗ Ngọc Sơn

**Lớp:** HTTT02 **– Nhóm:** 9

|  |  |
| --- | --- |
| **STT** | **Họ Tên** |
| 1 | Phạm Văn Phi - 2018604197 |
| 2 | Mai Văn Nam - 2018606538 |
| 3 | Lê Hồng Lâm - 2018604048 |
| 4 | Hà Thị Nhiên - 2018604509 |
| 5 | Nguyễn Thị Hương - 2018605177 |

**Hà nội, Năm 2021**

**Bảng phân công công việc**

|  |  |  |
| --- | --- | --- |
| **MSV** | **Họ Tên** | **Công việc được phân** |
| 2018604197 | Phạm Văn Phi | **Phân tích thiết kế hệ thống,thiết kế giao diện chính, Chức năng đặt phòng,Đăng nhập theo quyền ,quản lý tài khoản chức năng phân quyền** |
| 2018606538 | Mai Văn Nam | **Phân tích thiết kế hệ thống, Quản lý phòng hiển thị danh sách phòng và số lượng phòng trống,quản lý khách hàng chức năng sửa,xóa, viết báo cáo** |
| 2018604048 | Lê Hồng Lâm | **Phân tích thiết kế hệ thống, đặt phòng, in hóa đơn, chức năng thanh toán, quản lý khách hàng chức năng thêm tìm kiếm** |
| 2018604509 | Hà Thị Nhiên | **Quản lý dịch vụ, in báo cáo thống kê,quản lý phòng chức năng cập nhật phòng, viết báo cáo mô tả hệ thống** |
| 2018605177 | Nguyễn Thị Hương | **Quản lý danh sách nhân viên, quản lý hóa đơn,báo cáo thống kê , viết báo cáo mô tả hệ thống, quản lý tài khoản chức năng đăng ký, sửa, xóa tài khoản** |

# LỜI MỞ ĐẦU

Trong thời đại công nghệ 4.0 đang ngày càng phát triển, ứng dụng công nghệ thông tin vào đời sống được xem là một trong những yếu tố mang tính quyết định trong các hoạt động của tổ chức, cũng như các công ty, cửa hàng, nó đóng vai trò hết sức quan trọng, có thể tạo ra những bước đột phá mạnh mẽ.

Giờ đây, mọi việc liên quan đến thông tin trở nên thật dễ dàng cho người sử dụng: chỉ cần có một máy tính kết nối internet và một dòng dữ liệu truy tìm thì gần như lập tức… cả thế giới về các vấn đề mà bạn đang quan tâm sẽ hiện ra ngay trước mắt bạn, có đầy đủ thông tin hình ảnh và thậm chí nhiều thứ khác thú vị hơn nữa nếu bạn tìm đến nó…

Trong hoạt động sản xuất, kinh doanh, giờ đây, thương mại điện tử đã khẳng định được vai trò xúc tiến và thúc đẩy sự phát triển của nền kinh tế. Đối với một cửa hàng hay shop, việc quảng bá và giới thiệu đến khách hàng các sản phẩm mới đáp ứng được nhu cầu của khách hàng sẽ là cần thiết. Vì vậy phải quảng bá như thế nào đó là xây dựng được một phần mềm cho các cửa hàng có thể quảng bá tất cả các sản phẩm của mình bán.

Vì vậy chúng em đã thực hiện đề tài **“XÂY DỰNG PHẦN MỀM QUẢN LÝ KHÁCH SẠN VÀ ĐẶT PHÒNG”** để vận dụng những kiến thức chúng em đã tìm tòi được vào thực tiễn, xây dựng một phần mềm cơ bản giúp người sử dụng có thể quản lý dễ dàng khách sạn cửa mình và giúp khách hàng đặt phòng dễ dàng khi đến cửa khách sạn của bạn. Nhân viên sẽ gửi thông tin chi tiết của phòng cho khách hàng khi nhận được đầy đủ thông tin của khách hàng.

Với sự hướng dẫn, động viên tận tình của thầy **Đỗ Ngọc Sơn**, chúng em đã hoàn thành bài báo cáo đề tài này. Do chưa có nhiều kinh nghiệm nghiên cứu, thực hành nên chúng em cũng không tránh khỏi những thiếu sót. Chúng em rất mong nhận được sự thông cảm và góp ý của thầy để đề tài của chúng em được hoàn thiện hơn.

***Chúng em xin chân thành cảm ơn!***

**Sinh viên thực hiện**

Phạm Văn Phi

Mai Văn Nam

Lê Hồng Lâm

Hà Thị Nhiên

Nguyễn Thị Hương

**Mục lục**

[LỜI MỞ ĐẦU 3](#_Toc73814925)

[PHẦN 1: THIẾT KẾ CƠ SỞ DỮ LIỆU 5](#_Toc73814926)

[**1.1.** **Sơ đồ quan hệ các bảng** 5](#_Toc73814927)

[1.2 Cấu trúc các bảng trong database 5](#_Toc73814928)

[PHẦN 2: THIẾT KẾ VÀ CÀI ĐẶT CHƯƠNG TRÌNH 9](#_Toc73814929)

[2.1. Giao diện các trang 9](#_Toc73814930)

[2.1.1. Màn hình Đăng nhập: 9](#_Toc73814931)

[**2.1.2.** Màn hình **Home:** 9](#_Toc73814932)

[2.1.3. Màn hình Quản lý khách hàng: 11](#_Toc73814933)

[2.1.4. Màn hình Quản lý phòng: 12](#_Toc73814934)

[2.1.5. Màn hình **Đặt phòng**: 13](#_Toc73814935)

[**2.1.6.** Màn hình **Quản lý dịch vụ**: 14](#_Toc73814936)

[**2.1.7.** Màn hình **Quản lý hóa đơn**: 15](#_Toc73814937)

[**2.1.8.** Màn hình **Quản lý nhân viên**: 16](#_Toc73814938)

[**2.1.9.** Màn hình **Quản lý tài khoản**: 17](#_Toc73814939)

[2.1.10. Màn hình Báo cáo thống kê: 19](#_Toc73814940)

[**2.2.** Code C# của một số chức năng chính: 19](#_Toc73814941)

[**2.2.1. Code form Đăng nhập**: 20](#_Toc73814942)

[2.2.2. Code formdatphong: 21](#_Toc73814943)

[2.2.3. Code formQuanlyhoadon: 27](#_Toc73814944)

[2.2.4. Code formQuanlyhanvien: 28](#_Toc73814945)

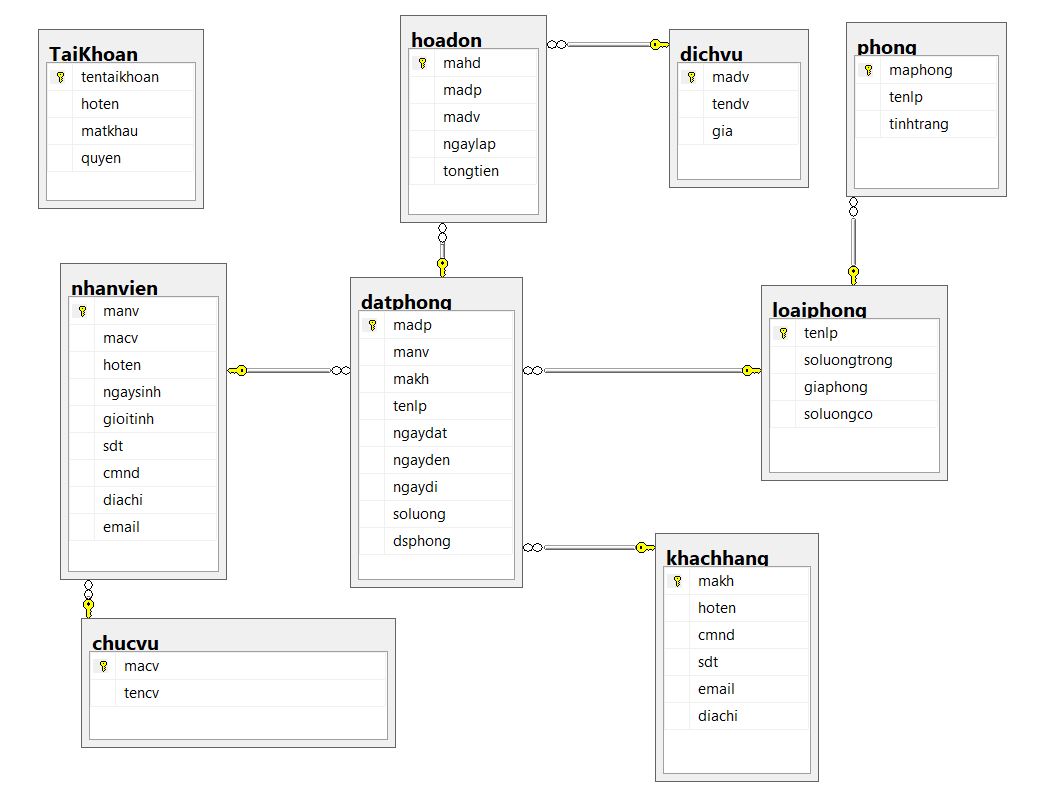
[2.2.5. Code formThongke: 32](#_Toc73814946)

[PHẦN 3: KẾT LUẬN 33](#_Toc73814947)

[TÀI LIỆU THAM KHẢO 34](#_Toc73814948)

# PHẦN 1: THIẾT KẾ CƠ SỞ DỮ LIỆU

* 1. **Sơ đồ quan hệ các bảng**



## Cấu trúc các bảng trong database

* + 1. Bảng **phong** (Phòng):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| maphong | char(4) | primary key |  |
| tenlp | varchar(10) | foreign key |  |
| tinhtrang | bit |  |  |

* + 1. Bảng **chucvu** (Chức vụ):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| macv | char(2) | primary key | Not null |
| tencv | nvarchar(30) |  |  |

* + 1. Bảng **nhanvien** (Nhân viên):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| manv | char(5) | primary key |  |
| macv | char(2) | foreign key | Not null |
| hoten | nvarchar(30) |  | Not null |
| ngaysinh | date |  | Not null |
| gioitinh | nvarchar(3) |  | Not null |
| sdt | varchar(12) |  | Not null |
| diachi | nvarchar(50) |  | Not null |
| cmnd | char(9) |  | Not null |
| email | varchar(30) |  |  |

* + 1. Bảng **TaiKhoan** (Tài Khoản):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| tentaikhoan | nvarchar(20) | primary key | Not null |
| hoten | nvarchar(30) |  | Not null |
| matkhau | nvarchar(8) |  | Not null |
| quyen | nvarchar(20) |  | Not null |

* + 1. Bảng **loaiphong** (Loại phòng):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| tenlp | varchar(10) | primary key |  |
| soluongtrong | int |  | Not null |
| giaphong | int |  | Not null |
| soluongco | int |  | Not null |

* + 1. Bảng **khachhang** (Khách hàng):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| makh | char(5) | primary key |  |
| hoten | nvarchar(30) |  | Not null |
| sdt | varchar(12) |  | Not null |
| diachi | nvarchar(50) |  | Not null |
| cmnd | char(9) |  | Not null |
| email | varchar(30) |  |  |

* + 1. Bảng **dichvu** (Dịch vụ):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| madv | nvarchar(20) | primary key | Not null |
| tendv | nvarchar(20) |  |  |
| gia | money |  |  |

* + 1. Bảng **datphong** (Đặt phòng):

|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| madp | char(5) | primary key |  |
| manv | char(5) | foreign key | Not null |
| makh | char(5) | foreign key | Not null |
| tenlp | varchar(10) | foreign key | Not null |
| ngaydat | date |  | Not null |
| ngayden | date |  | Not null |
| ngaydi | date |  | Not null |
| soluong | int |  | Not null |
| trangthaitt | nvarchar |  | Not null |
| dsphong | Nvarchar(200) |  |  |

* + 1. Bảng **hoadon** (Hóa đơn):

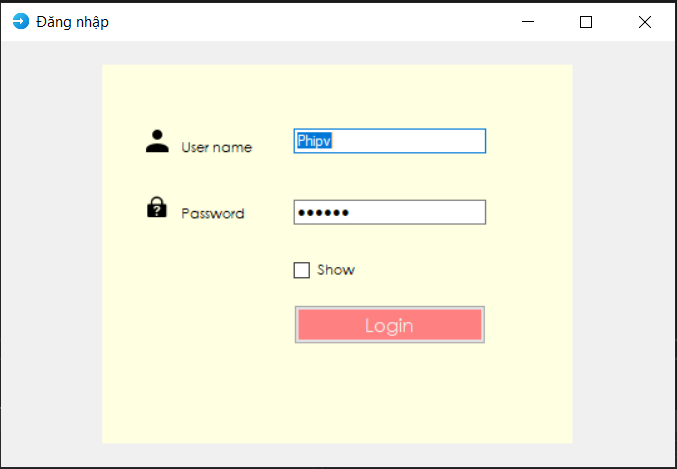
|  |  |  |  |
| --- | --- | --- | --- |
| Column Name | Data Type | Key | Allow null |
| mahd | char(5) | primary key |  |
| madp | char(5) | foreign key | Not null |
| madv | nvarchar(20) | foreign key | Not null |
| ngaylap | date |  | Not null |
| tongtien | int |  | Not null |

# 

# PHẦN 2: THIẾT KẾ VÀ CÀI ĐẶT CHƯƠNG TRÌNH

## 2.1. Giao diện các trang

### 2.1.1. Màn hình Đăng nhập:



* **Mô tả**:

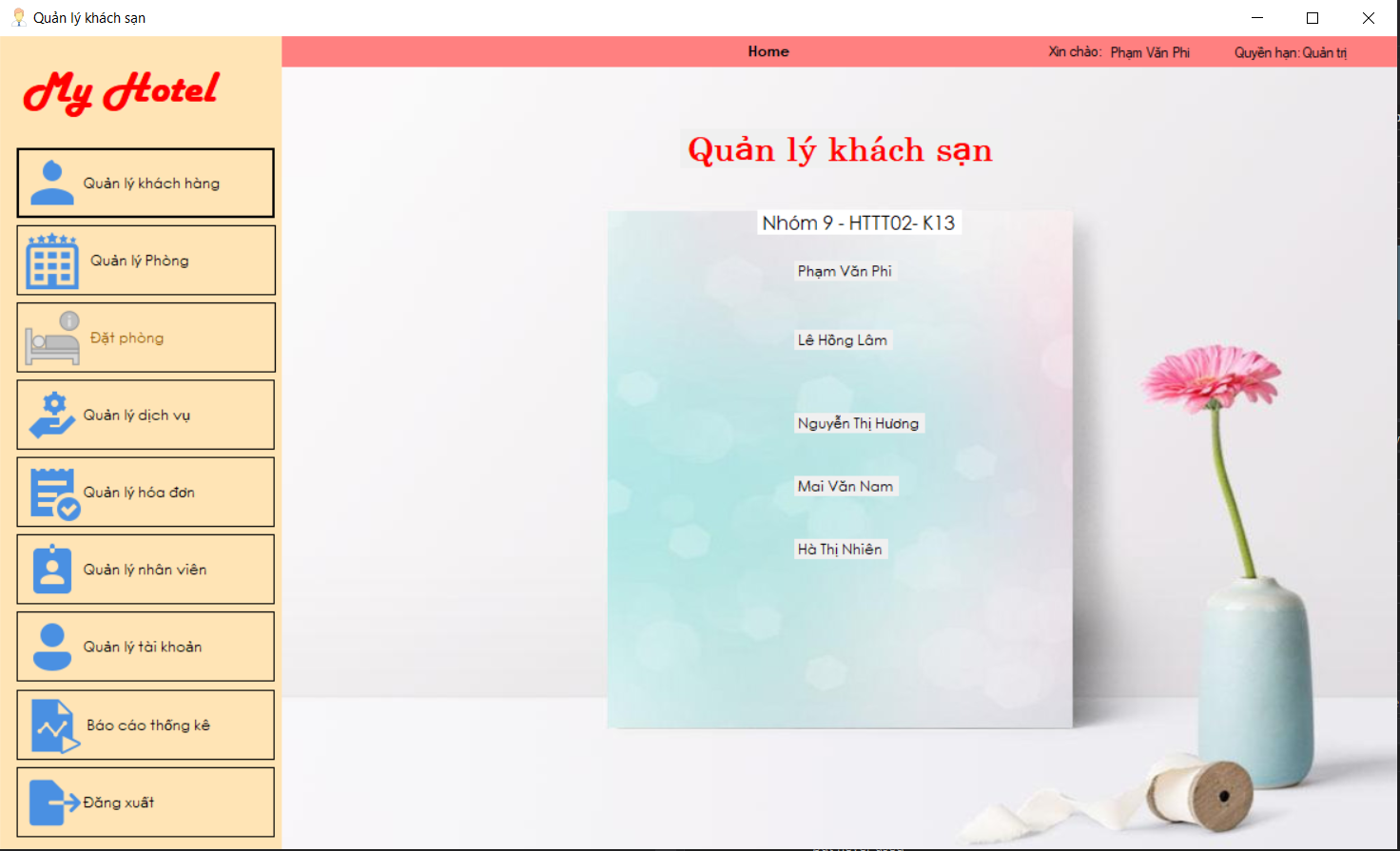
+ Dùng để đăng nhập tài khoản trước khi vào màn hình chính.

+ Đăng nhập với tư cách là admin và quan lý thì sẽ vào giao diện của quản lý, đăng nhập với tư cách nhân viên sẽ vào giao diện của nhân viên.

+ Click 1 lần vào con mắt thì show ra mật khẩu, click lại lần 2 thì con mắt show ra kí tự mã hóa.

+ Để trống mã tài khoản hoặc mật khẩu đưa ra cảnh báo.

**2.1.2.** Màn hình **Home:**



* **Mô tả**: Hiển thị tổng quát toàn bộ chức năng của phần mềm bao gồm:

+ Quản lý khách hàng.

+ Quản lý phòng.

+ Đặt phòng.

+ Quản lý dịch vụ.

+ Quản lý hóa đơn.

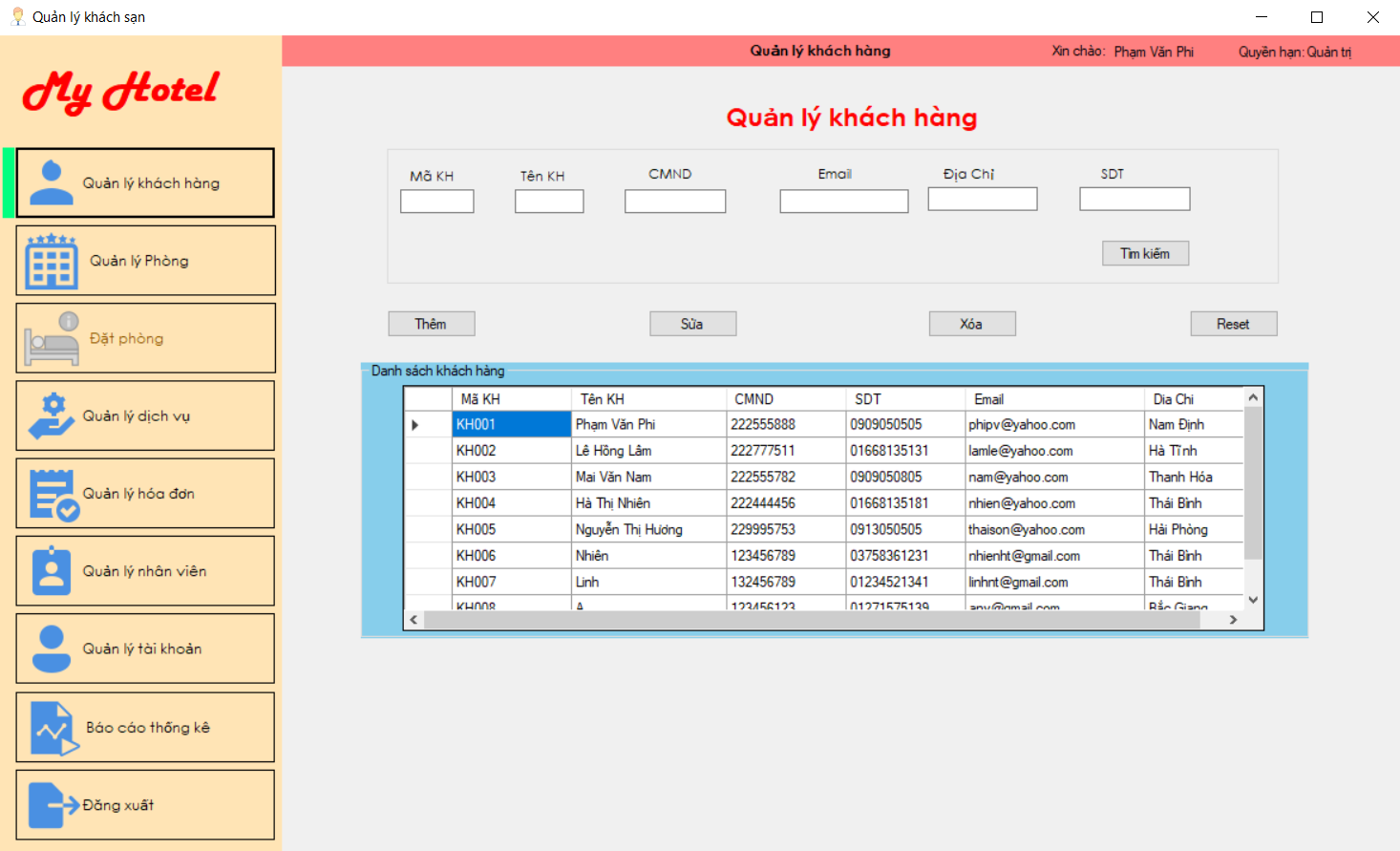
+ Quản lý nhân viên.

+ Quản lý tài khoản.

+ Báo cáo thống kê.

+ Đăng xuất.

### 2.1.3. Màn hình Quản lý khách hàng:



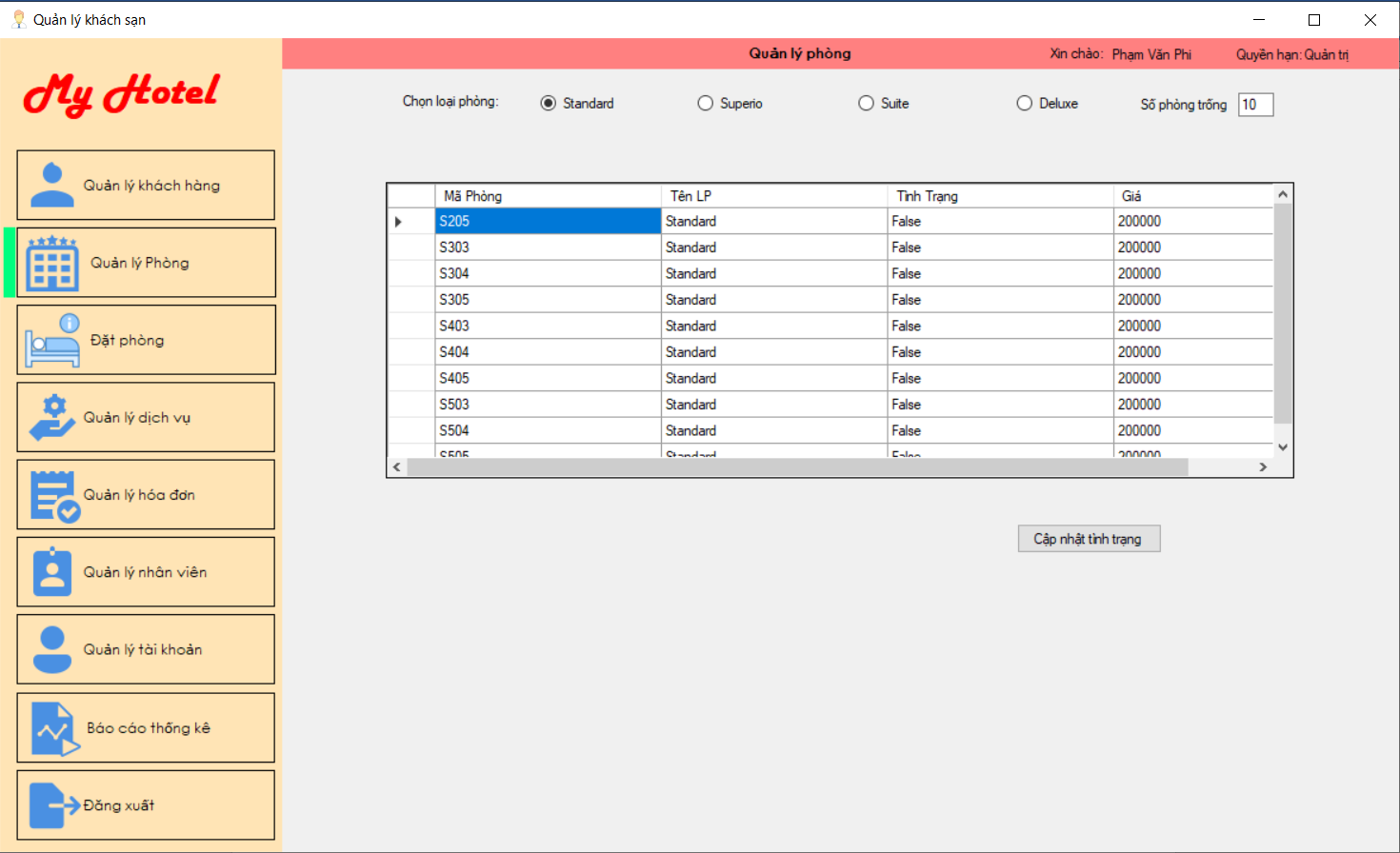
* **Mô tả**:

+ Hiển thị toàn bộ khách hàng đã check in tại khách sạn khi đã có hóa đơn thanh toán.

+ Bao gồm các nút chức năng như: thêm, sửa, xóa, tìm kiếm khách hàng

+ Người dùng không có quyền xóa hay thêm hay cập nhật gì từ màn hình này, người dùng muốn thêm khách hàng thì chỉ được thêm khi đã có hóa đơn từ màn hình chi tiết hóa đơn.

### 2.1.4. Màn hình Quản lý phòng:

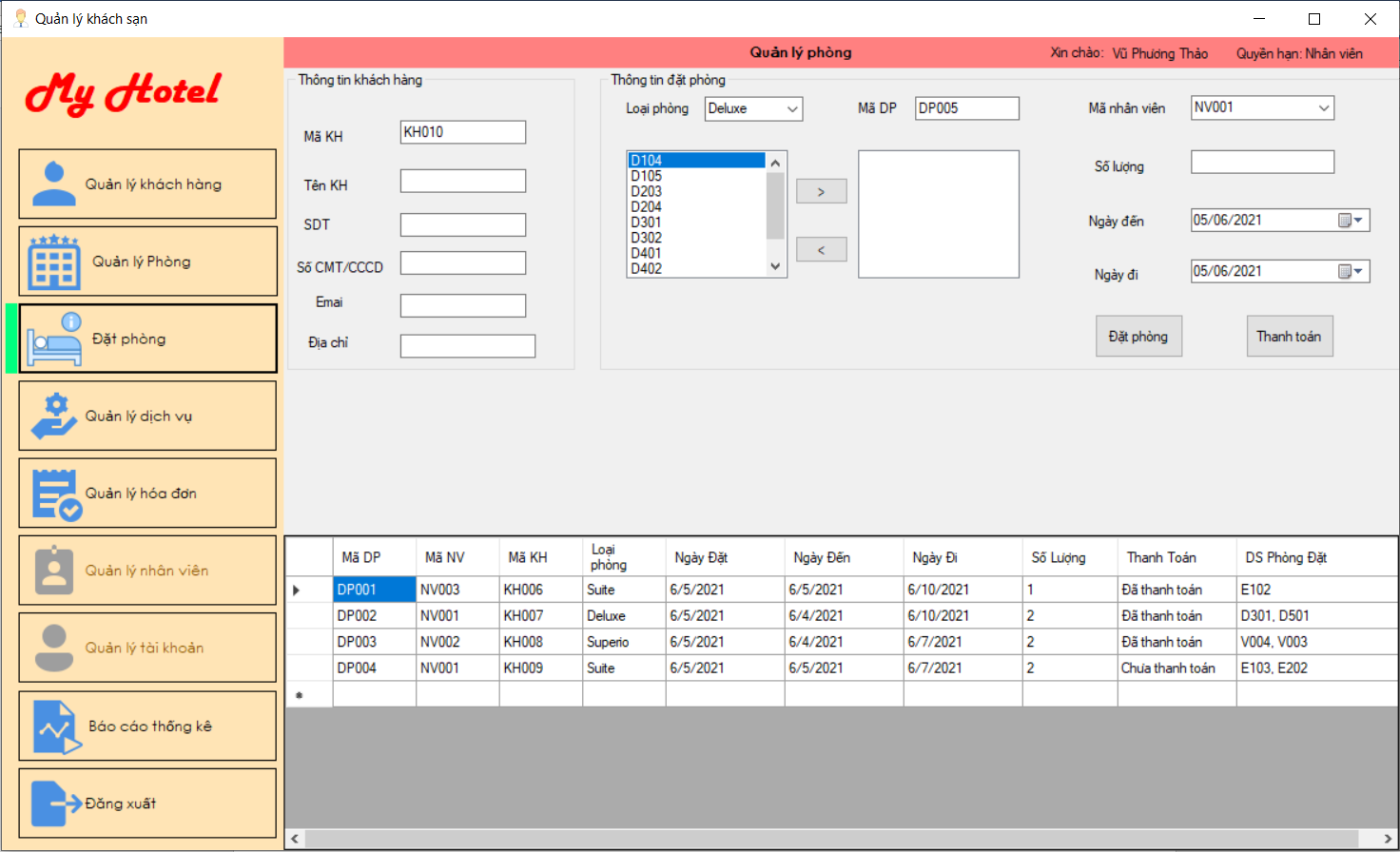


* **Mô tả**:

+ Tùy vào loại phòng, hệ thống sẽ hiển thị toàn bộ số phòng đang có trong khách sạn, bao gồm số phòng, loại phòng, tình trạng và giá phòng trên đó, người dùng không được thay đổi thông tin gì trên màn hình này.

+ Click vào “Cập nhật tình trạng”, dữ liệu về tình trạng phòng sẽ được update lại. Nếu phòng đã được đặt thì tình trạng là “true”, ngược lại sẽ là “false”.

2.1.5. Màn hình **Đặt phòng**:



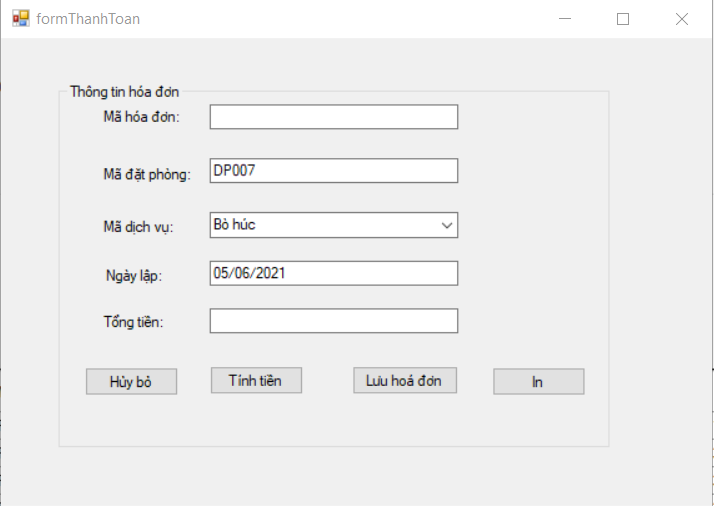
* **Mô tả**:

+ Hiển thị khung Thông tin khách hàng để nhập thông tin khách hàng đặt phòng.

+ Hiển thị khung Thông tin đặt phòng bao gồm: Loại phòng, Mã DP, Mã nhân viên, Số lượng phòng đặt, Ngày đến và Ngày đi.

+ Click vào button “Đặt phòng” để xác nhận đặt phòng:

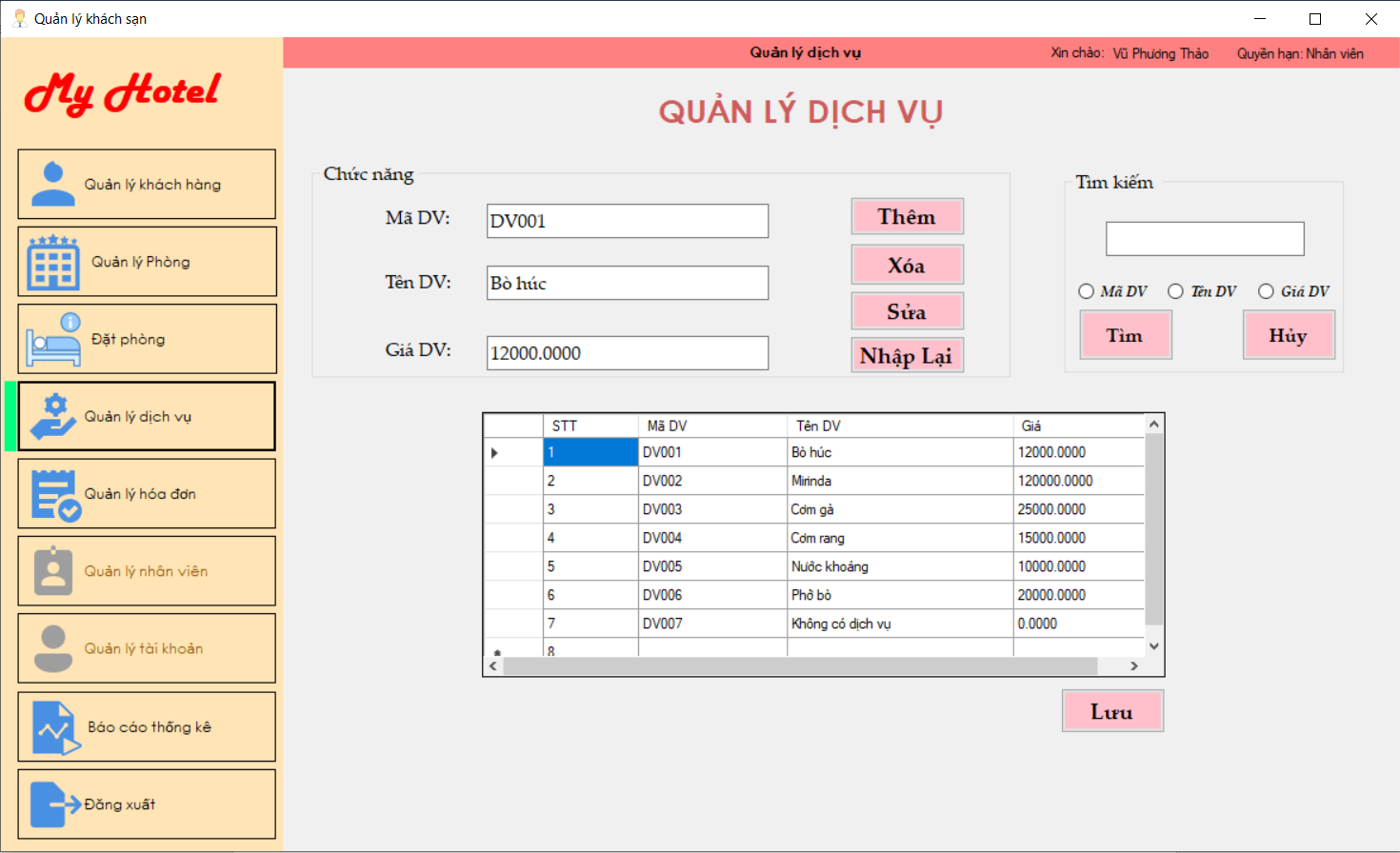
+ Click vào button “Thanh toán” sẽ hiển thị form Thanh toán:



* Trong form Thanh toán có các chức năng:
* Click vào button “Hủy bỏ” để hủy thao tác Thanh toán.
* Click vào button “Tính tiền” để tính tổng tiền cần thanh toán.
* Click vào button “Lưu hóa đơn” để lưu lại hóa đơn đó.
* Click vào button “In” để in ra hóa đơn đó.

+ Hiển thị trong datagridview toàn bộ đơn đặt phòng đã được đặt cùng tình trạng thanh toán.

**2.1.6.** Màn hình **Quản lý dịch vụ**:



* **Mô tả**:

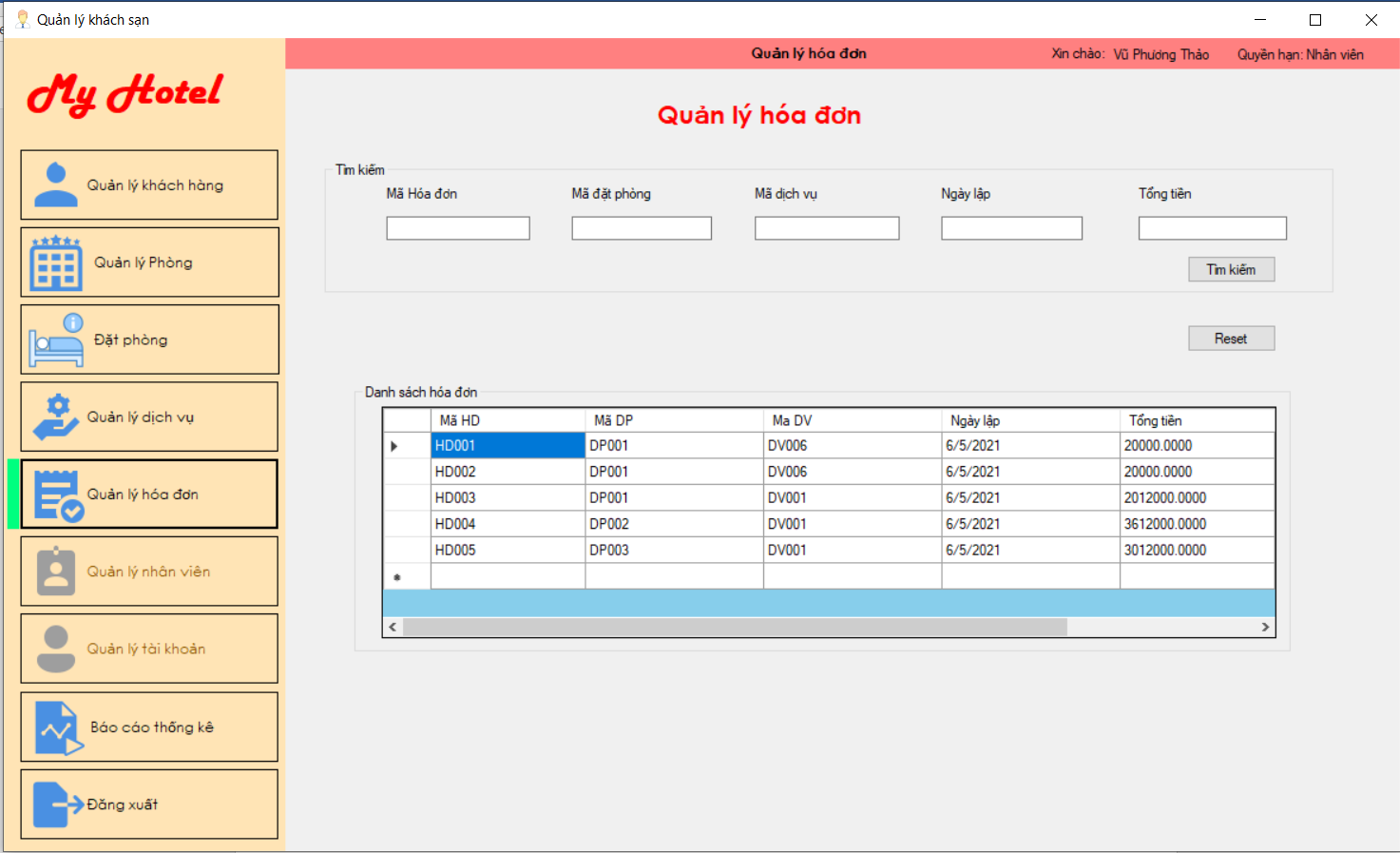
+ Hiển thị toàn bộ dịch vụ đang có trong khách sạn như: bò húc, Mirinda, cơm gà, cơm rang, nước khoáng, phở bò....

+ Có các nút chức năng như: thêm, sửa, xóa dịch vụ, cập nhật lại dịch vụ dựa theo thao tác chọn dòng từ datagridview của người dùng. Mỗi thao tác sẽ được cập nhật lại xuống database.

+ Có chức năng tìm kiếm theo các trường dữ liệu để người dùng kiểm soát được các dịch vụ hiện có của khách sạn.

+ Click vào sự kiện “Lưu” để lưu toàn bộ thay đổi.

**2.1.7.** Màn hình **Quản lý hóa đơn**:



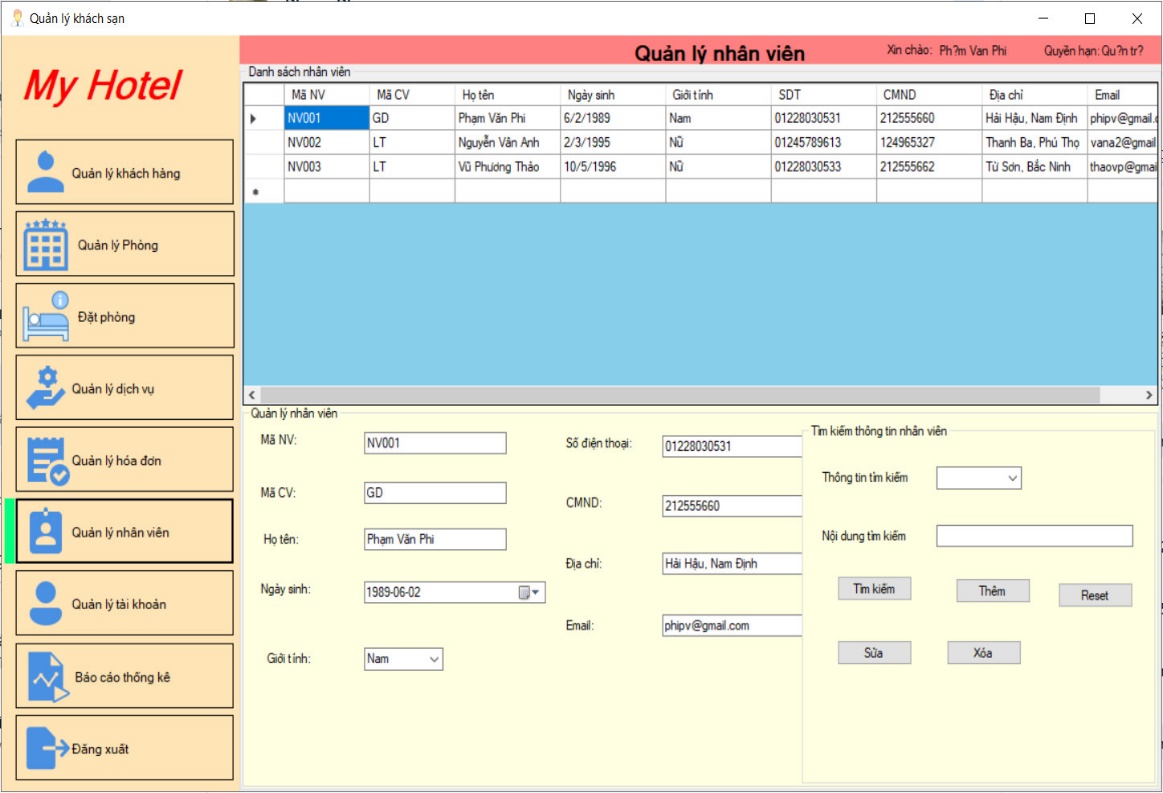
* **Mô tả**:

+ Cung cấp chức năng tìm kiếm cho người dùng theo các trường dữ liệu: Mã hóa đơn, mã đặt phòng, mã dịch vụ, ngày lập, tổng tiền.

+ Click vào button “Reset” để nhập lại các dữ liệu.

+ Hiển thị toàn bộ hóa đơn đã được lưu khi đặt phòng.

**2.1.8.** Màn hình **Quản lý nhân viên**:



* **Mô tả**:

**+** Tài khoản được phân quyền là “quản trị” mới được sử dụng chức năng này.

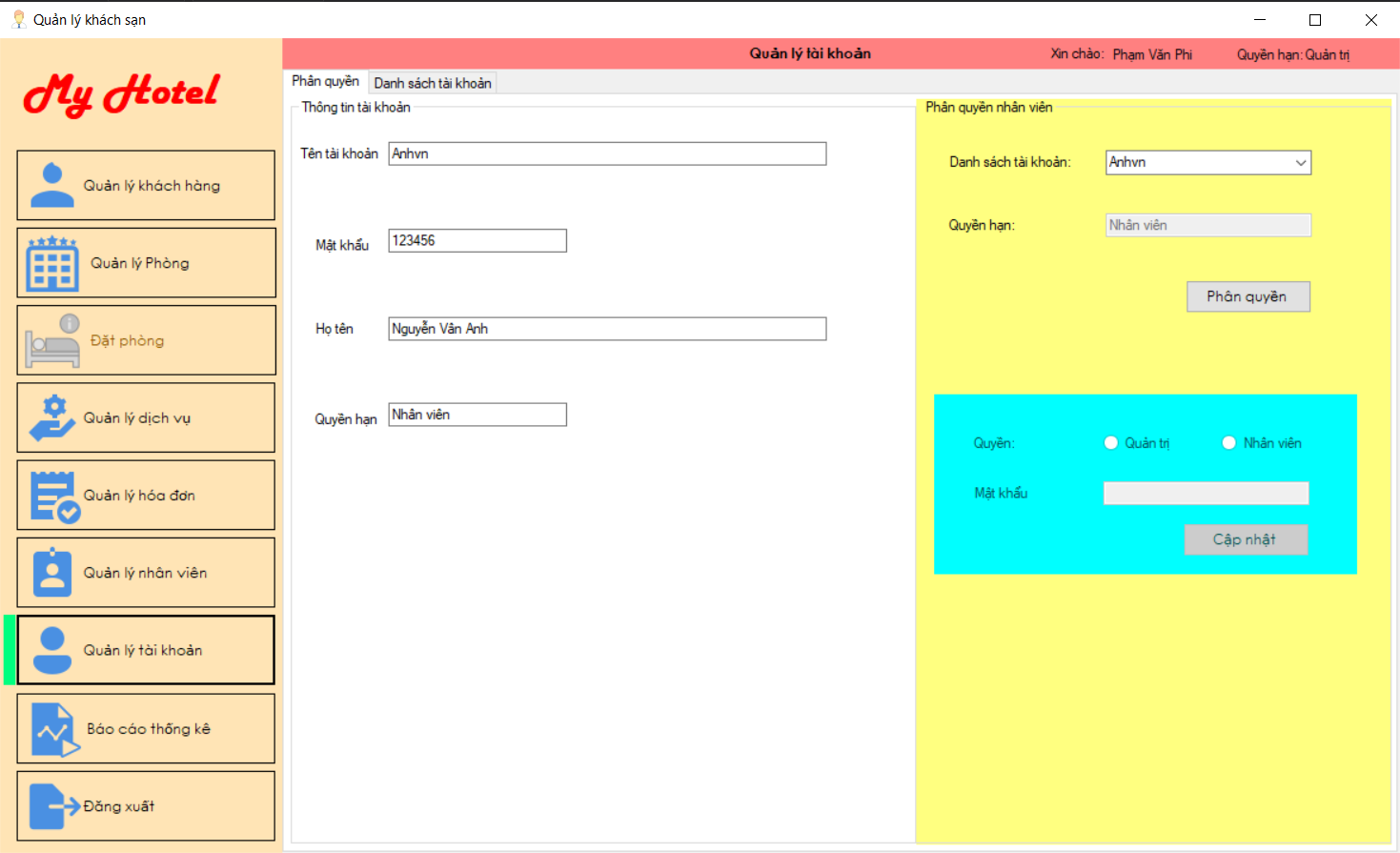
+ Hiển thị toàn bộ thông tin cá nhân của nhân viên có trong khách sạn bao gồm: Mã NV, Mã CV, Họ tên, Ngày sinh, Giới tính, SĐT, CMND, Địa chỉ, email.

+ Cung cấp các chức năng Thêm, Sửa, Xóa nhân viên. Dữ liệu sau đó được update lên datagridview.

+ Cung cấp chức năng tìm kiếm nhân viên theo các trường dữ liệu trong comboBox sau đó nhập thông tin vào nội dung tìm kiếm.

+ Click vào nút “Reset” để load lại form ban đầu.

**2.1.9.** Màn hình **Quản lý tài khoản**:



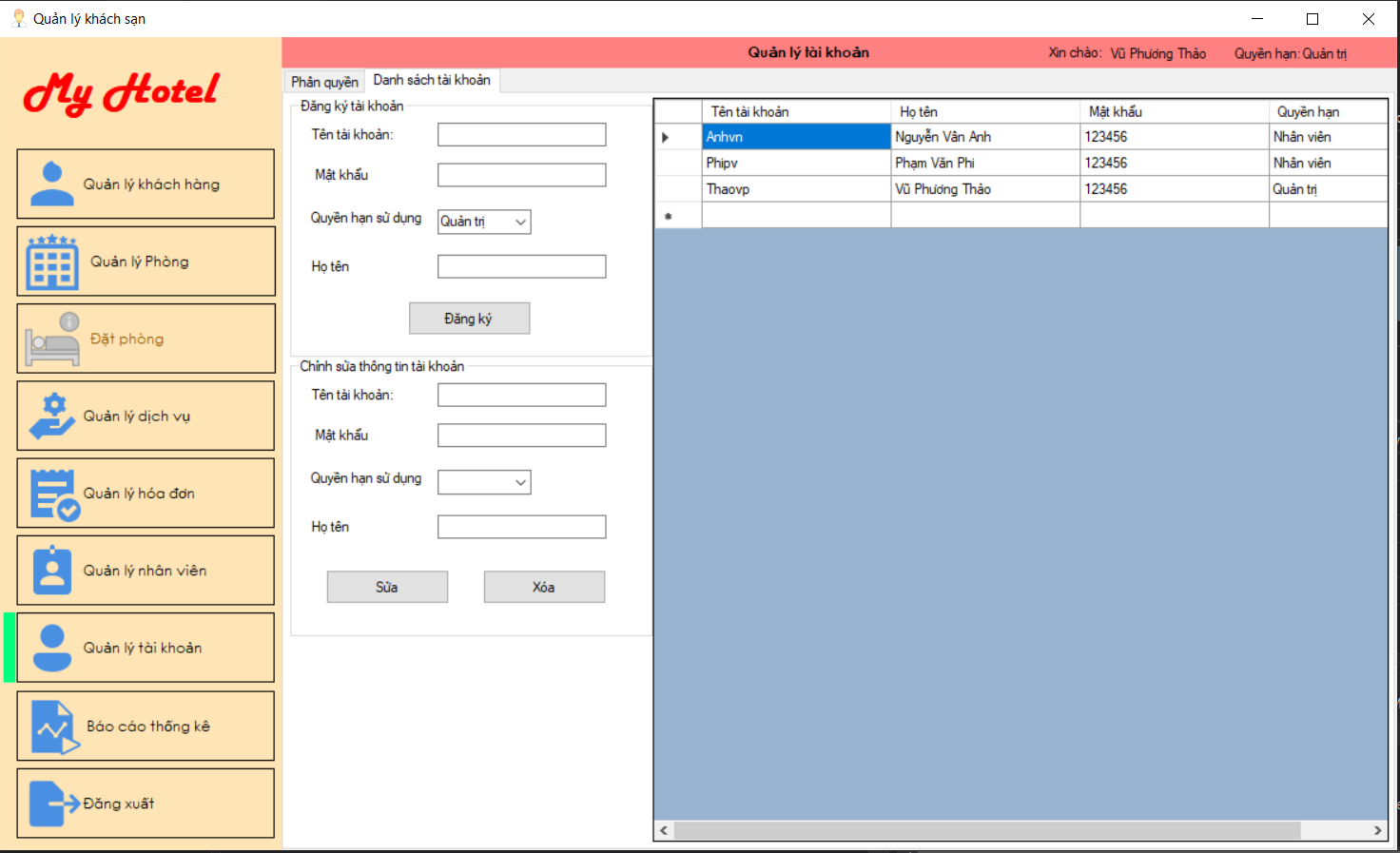
* Mô tả:

**+** Tài khoản được phân quyền là “quản trị” mới được sử dụng chức năng này.

+ Trong tab “Phân quyền”:

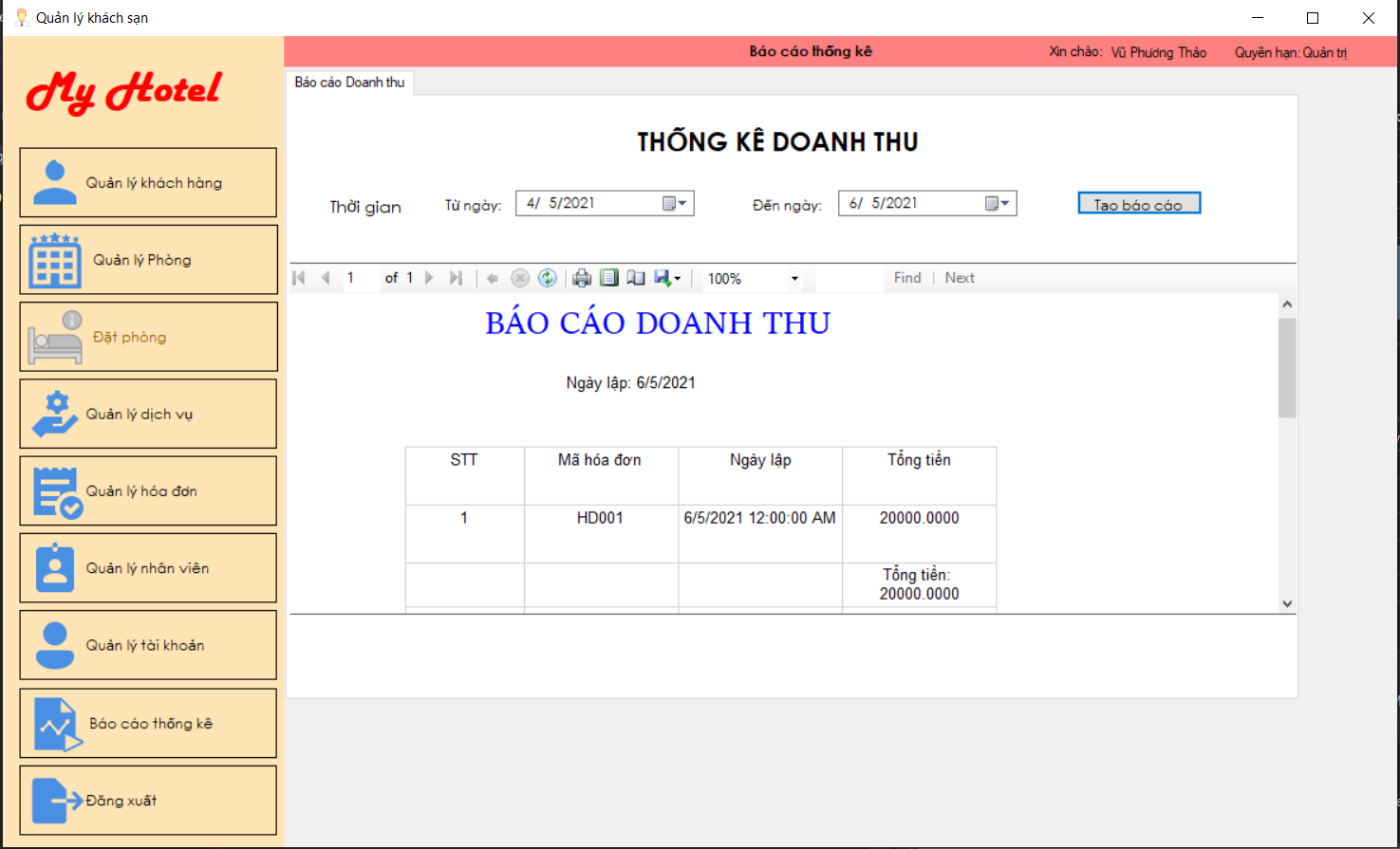
* Hiển thị thông tin tài khoản.
* Phân quyền đăng nhập cho các tài khoản.

+ Trong tab “**Danh sách tài khoản**”:



* Hiển thị toàn bộ tài khoản trong hệ thống quản lý.
* Cung cấp chức năng đăng ký tài khoản mới.
* Cung cấp chức năng chỉnh sửa thông tin tài khoản đã có.

### 2.1.10. Màn hình Báo cáo thống kê:

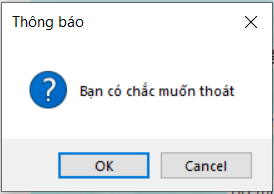


* **Mô tả**:

+ Cung cấp chức năng thống kê doanh thu theo ngày của khách sạn.

+ Click vào “Tạo báo cáo” để xuất ra báo cáo.

**2.1.11.** Màn hình **Đăng xuất**:



* **Mô tả**:

+ Hệ thống sẽ đưa ra thông báo có muốn thoát hay không?

+ Nếu muốn đăng xuất thì click “ok”, ngược lại click “cancel”.

**2.2.** Code C# của một số chức năng chính:

**2.2.1. Code form Đăng nhập**:

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;

using BUS;

using DTO;

namespace PMQLKS

{

public partial class formLogin : Form

{

public formLogin()

{

InitializeComponent();

}

DataTranferObject nv = new DataTranferObject();

private void formLogin\_Load(object sender, EventArgs e)

{

tbUserName.Text = "";

tbMatKhau.Text = "123456";

}

private void tbUserName\_MouseLeave(object sender, EventArgs e)

{

if (tbUserName.Text == "")

{

tbUserName.Text = "Tài khoản";

tbUserName.ForeColor = Color.Gray;

}

}

private void tbUserName\_MouseEnter(object sender, EventArgs e)

{

if (tbUserName.Text == "Tài khoản")

{

tbUserName.Text = "";

tbUserName.ForeColor = Color.Blue;

}

}

private void tbMatKhau\_MouseEnter(object sender, EventArgs e)

{

if (tbMatKhau.Text == "Mật khẩu")

{

tbMatKhau.Text = "";

tbMatKhau.ForeColor = Color.Blue;

}

}

public bool isLogin(string manv, string password)

{

return BusinessLogicLayer.Instance.Login(manv, password);

}

public string getPermission(string maNv, string password)

{

return BusinessLogicLayer.Instance.getUserIDPermision(maNv, password);

}

private void checkBoxShowOrHide\_CheckedChanged(object sender, EventArgs e)

{

if (checkBoxShowOrHide.Checked)

{

tbMatKhau.UseSystemPasswordChar = false;

checkBoxShowOrHide.Text = "Hide";

}

else

{

tbMatKhau.UseSystemPasswordChar = true;

checkBoxShowOrHide.Text = "Show";

}

}

private void btnDangNhap\_Click(object sender, EventArgs e)

{

nv.Tentaikhoan = tbUserName.Text.Trim();

nv.Matkhau= tbMatKhau.Text.Trim();

//getPermission(maNV, password);

string tenNV = BusinessLogicLayer.Instance.getTenNhanVien(nv.Tentaikhoan);

if (nv.Tentaikhoan == "" || nv.Matkhau == "")

{

MessageBox.Show("Bạn chưa điền mật khẩu hoặc password");

}

else if (isLogin(nv.Tentaikhoan, nv.Matkhau))

{

MessageBox.Show("Xin chào" + " " + tenNV, "Đăng nhập thành công", MessageBoxButtons.OK, MessageBoxIcon.Information);

formMain fm = new formMain(nv.Tentaikhoan, nv.Matkhau);

fm.Show();

this.Hide();

}

else

{

MessageBox.Show("Sai tên đăng nhập hoặc mật khẩu");

}

}

}

}

### 2.2.2. Code formdatphong:

using System;

using System.Collections;

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;

using BUS;

using DTO;

namespace PMQLKS

{

public partial class datPhong : Form

{

string trangthaithanhtoan;

string tenloaiphong;

string madatphong;

int songaythue;

int soluongphongdat;

string dsPhong;

string tenphong;

IList<string> arrDSPhong = new List<string>();

IList<string> arr = new List<string>();

IList<string> arr1 = new List<string>();

public datPhong()

{

InitializeComponent();

}

public void loadComboxLoaiPhong()

{

comboBoxLoaiPhong.DataSource = BusinessLogicLayer.Instance.getLoaiPhong();

comboBoxLoaiPhong.ValueMember = "tenlp";

comboBoxLoaiPhong.DisplayMember = "tenlp";

}

private void datPhong\_Load(object sender, EventArgs e)

{

loadComboxLoaiPhong();

loadListPhong();

loadDatgridView();

loadMaNV();

txtmakh.Text = setmakh();

tbMaDP.Text = setmadp();

}

public void loadListPhong()

{

tenphong = comboBoxLoaiPhong.SelectedValue.ToString();

//lbPhongTrong.DataBindings.Clear();

lbPhongTrong.DataSource = BusinessLogicLayer.Instance.getListPhong(tenphong);

lbPhongTrong.DisplayMember = "maphong";

lbPhongTrong.ValueMember = "maphong";

}

public void loadDatgridView()

{

dataGridViewDP.DataSource = BusinessLogicLayer.Instance.getListDatPhong();

}

private void comboBoxLoaiPhong\_SelectedValueChanged(object sender, EventArgs e)

{

loadListPhong();

}

private void setarr1()

{

DataTable dtb = new DataTable();

string tenlp = comboBoxLoaiPhong.SelectedValue.ToString();

dtb = BusinessLogicLayer.Instance.getListPhong(tenphong);

for (int i = 0; i < dtb.Rows.Count; i++)

{

DataRow r = dtb.Rows[i];

arr1.Add(r[0].ToString());

}

}

private void buttonAdd\_Click(object sender, EventArgs e)

{

if (arr1.Count == 0)

{

setarr1();

}

lbPhongDat.DataSource = null;

if (lbPhongTrong.SelectedIndex > -1)

{

arr.Add(lbPhongTrong.SelectedValue.ToString());

arr1.RemoveAt(lbPhongTrong.SelectedIndex);

}

lbPhongDat.ClearSelected();

BindingSource bsOne = new BindingSource();

bsOne.DataSource = arr;

lbPhongDat.DataSource = bsOne;

lbPhongTrong.DataSource = null;

lbPhongTrong.DataSource = arr1;

int soluong = arr.Count;

tbSoluong.Text = soluong.ToString();

}

private void btnBack\_Click(object sender, EventArgs e)

{

string tmp = arr[lbPhongDat.SelectedIndex].ToString();

arr.RemoveAt(lbPhongDat.SelectedIndex);

arr1.Add(tmp);

lbPhongTrong.DataSource = null;

lbPhongTrong.DataBindings.Clear();

lbPhongTrong.DataSource = arr1;

lbPhongDat.DataSource = null;

lbPhongDat.DataBindings.Clear();

lbPhongDat.DataSource = arr;

int soluong = arr.Count;

tbSoluong.Text = soluong.ToString();

}

private string setmadp()

{

int id;

id = int.Parse(BusinessLogicLayer.Instance.GetLastID("datphong", "madp").Substring(2, 3));

if (id < 9)

return "DP00" + (id + 1).ToString();

else

if (id < 99)

return "DP0" + (id + 1).ToString();

else

return "DP" + (id + 1).ToString();

}

private string setmakh()

{

int id;

id = int.Parse(BusinessLogicLayer.Instance.GetLastID("khachhang", "makh").Substring(2, 3));

if (id < 9)

return "KH00" + (id + 1).ToString();

else

if (id < 99)

return "KH0" + (id + 1).ToString();

else

return "KH" + (id + 1).ToString();

}

public void loadMaNV()

{

comboBoxNV.DataSource = BusinessLogicLayer.Instance.getMaNV();

comboBoxNV.ValueMember = "manv";

comboBoxNV.DisplayMember = "manv";

}

private void insertDatPhong()

{

dbDataContext db = new dbDataContext();

datphong dp = new datphong();

tbMaDP.Text = setmadp();

dp.madp = tbMaDP.Text;

dp.manv = comboBoxNV.SelectedValue.ToString();

dp.makh = txtmakh.Text;

dp.tenlp = comboBoxLoaiPhong.SelectedValue.ToString();

dp.ngaydat = DateTime.Now;

dp.ngayden = dateTimeDen.Value;

dp.ngaydi = dateTimeDi.Value;

tbSoluong.Text = arr.Count.ToString();

dp.soluong = Int16.Parse(tbSoluong.Text);

dp.trangthaitt = "Chưa thanh toán";

dp.dsphong = String.Join(", ", arr);

db.datphongs.InsertOnSubmit(dp);

db.SubmitChanges();

}

private void insertKhachHang()

{

dbDataContext db = new dbDataContext();

khachhang newkh = new khachhang();

newkh.makh = txtmakh.Text;

newkh.hoten = txttenkh.Text;

newkh.cmnd = txtcmt.Text;

newkh.sdt = txtsdt.Text;

newkh.diachi = txtdc.Text;

newkh.email = txtemail.Text;

db.khachhangs.InsertOnSubmit(newkh);

db.SubmitChanges();

}

private void buttonDatPhong\_Click(object sender, EventArgs e)

{

dbDataContext db = new dbDataContext();

DateTime ngayDen = dateTimeDen.Value;

DateTime ngayDi = dateTimeDi.Value;

int reslut = DateTime.Compare(ngayDen, ngayDi);

try

{

if (txtmakh.Text == "" ||

txttenkh.Text == "" ||

txtcmt.Text == "" ||

txtsdt.Text == "" ||

txtdc.Text == "" ||

txtemail.Text == "" ||

comboBoxLoaiPhong.Text == "" ||

comboBoxNV.Text == "" ||

comboBoxLoaiPhong.Text == "" ||

tbSoluong.Text == ""

)

{

MessageBox.Show("Bạn phải điền đầy dủ thông tin!!!");

}

else if (reslut == 1)

{

MessageBox.Show("Ngày đi phải lớn hơn ngày đến");

}

else

{

DialogResult dlr = MessageBox.Show("Bạn chắc chắn muốn đặt phòng?", "Thông báo", MessageBoxButtons.YesNo);

if (dlr == DialogResult.Yes)

{

insertKhachHang();

insertDatPhong();

for (int i = 0; i < arr.Count; i++)

{

BusinessLogicLayer.Instance.capNhatTrangThaiPhongDat(arr[i].ToString());

}

BusinessLogicLayer.Instance.capNhatSoLuongPhong(comboBoxLoaiPhong.SelectedValue.ToString(), arr.Count);

loadDatgridView();

MessageBox.Show("Đặt phòng thành công!");

}

}

}

catch (Exception ex)

{

MessageBox.Show("Error" + ex);

}

}

private void buttoThanhToan\_Click(object sender, EventArgs e)

{

if (this.trangthaithanhtoan == "Đã thanh toán")

{

MessageBox.Show("Bạn hãy chọn đặt phòng chưa thanh toán trong bảng dữ liệu!");

}

else

{

formThanhToan f = new formThanhToan(madatphong, tenloaiphong, songaythue, soluongphongdat, arrDSPhong);

f.Show();

}

}

private void dateTimeDen\_ValueChanged(object sender, EventArgs e)

{

}

private void dataGridViewDP\_CellClick(object sender, DataGridViewCellEventArgs e)

{

DataGridViewRow row = this.dataGridViewDP.Rows[e.RowIndex];

this.trangthaithanhtoan = row.Cells[9].Value.ToString();

this.madatphong = row.Cells[0].Value.ToString();

this.tenloaiphong = row.Cells[3].Value.ToString();

this.songaythue = Convert.ToInt32((DateTime.ParseExact(row.Cells[6].Value.ToString(), "M/d/yyyy hh:mm:ss tt", null)).Subtract(DateTime.ParseExact(row.Cells[5].Value.ToString(), "M/d/yyyy hh:mm:ss tt", null)).TotalDays);

this.soluongphongdat = Convert.ToInt32(row.Cells[7].Value.ToString());

this.dsPhong = row.Cells[8].Value.ToString();

arrDSPhong = dsPhong.Split(',');

}

}

}

### 2.2.3. Code formQuanlyhoadon:

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 PMQLKS

{

public partial class formQuanLyHoaDon : Form

{

public formQuanLyHoaDon()

{

InitializeComponent();

}

private void btnTimkiem\_Click(object sender, EventArgs e)

{

try

{

dbDataContext db = new dbDataContext();

IQueryable<hoadon> table = db.hoadons.AsQueryable();

if (!string.IsNullOrWhiteSpace(tbMahd.Text))

table = table.Where(hd => hd.mahd.Contains(tbMahd.Text));

if (!string.IsNullOrWhiteSpace(tbMadp.Text))

table = table.Where(hd => hd.madp.Contains(tbMadp.Text));

if (!string.IsNullOrWhiteSpace(tbMadv.Text))

table = table.Where(hd => hd.madv.Contains(tbMadv.Text));

if (!string.IsNullOrWhiteSpace(tbNgaylap.Text))

table = table.Where(hd => hd.ngaylap.ToString().Contains(tbNgaylap.Text));

if (!string.IsNullOrWhiteSpace(tbTongtien.Text))

table = table.Where(hd => hd.tongtien == Int32.Parse(tbNgaylap.Text));

dataGridViewHoaDon.DataSource = table.Select(hd => hd);

}

catch (Exception ex)

{

MessageBox.Show("Lỗi tìm kiếm!" + ex);

}

}

private void dataGridViewQLHoaDon\_Load()

{

dbDataContext db = new dbDataContext();

var hoadon = from hd in db.hoadons

select hd;

dataGridViewHoaDon.DataSource = hoadon;

}

private void btnReset\_Click(object sender, EventArgs e)

{

dataGridViewQLHoaDon\_Load();

}

private void formQuanLyHoaDon\_Load(object sender, EventArgs e)

{

dataGridViewQLHoaDon\_Load();

}

}

}

### 2.2.4. Code formQuanlyhanvien:

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;

using BUS;

using DTO;

namespace PMQLKS

{

public partial class formQuanLyNhanVien : Form

{

public formQuanLyNhanVien()

{

InitializeComponent();

}

DataTranferObject dto = new DataTranferObject();

private void formQuanLyNhanVien\_Load(object sender, EventArgs e)

{

loadDataGridViewNV();

reset();

}

private void loadDataGridViewNV()

{

dataGridViewListNhanVien.DataSource = BusinessLogicLayer.Instance.getNhanVienInfor();

}

public void reset()

{

textboxMaNV.ResetText();

textBoxHoTenNV.ResetText();

comboBoxGioiTinhNV.Text = "";

dateTimePickerNSNV.Value = DateTime.Today;

textBoxDiaChiNV.ResetText();

textBoxSDTNV.ResetText();

textBoxEmailNV.ResetText();

textBoxCMNDNV.ResetText();

textBoxMaCVNV.ResetText();

loadDataGridViewNV();

}

private void buttonThemNV\_Click\_1(object sender, EventArgs e)

{

try

{

dto.Manv = textboxMaNV.Text.Trim().ToString();

dto.Macv = textBoxMaCVNV.Text.Trim().ToString();

dto.Hoten = textBoxHoTenNV.Text.ToString();

dto.Ngaysinh = dateTimePickerNSNV.Value;

dto.Gioitinh = comboBoxGioiTinhNV.SelectedItem?.ToString();

dto.Diachi = textBoxDiaChiNV.Text.ToString();

dto.Sdt = textBoxSDTNV.Text.ToString();

dto.Email = textBoxEmailNV.Text.ToString();

dto.Cmnd = textBoxCMNDNV.Text.ToString();

// MessageBox.Show(dto.Manv);

if (textboxMaNV.Text == "" || textBoxMaCVNV.Text == "" || textBoxHoTenNV.Text == "" || comboBoxGioiTinhNV.Text == "" || dateTimePickerNSNV.Value == null || textBoxDiaChiNV.Text == "" || textBoxSDTNV.Text == "" || textBoxEmailNV.Text == "" || textBoxCMNDNV.Text == "")

{

MessageBox.Show("Vui lòng điền đầy đủ thông tin!!!");

}

else

{

BusinessLogicLayer.Instance.insertNhanvien(dto.Manv, dto.Macv, dto.Hoten, dto.Ngaysinh, dto.Gioitinh, dto.Sdt, dto.Cmnd, dto.Diachi, dto.Email);

MessageBox.Show("Thêm nhân viên thành công!!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loadDataGridViewNV();

}

}

catch(Exception ex)

{

MessageBox.Show("Xảy ra lỗi khi thêm nhân viên"+ "\n" + ex);

}

}

private void buttonReset\_Click(object sender, EventArgs e)

{

reset();

}

private void buttonEdit\_Click(object sender, EventArgs e)

{

try

{

if (textboxMaNV.Text == "")

{

MessageBox.Show("Bạn phải nhập mã nhân viên cần sửa!!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else if (textBoxMaCVNV.Text == "" || textBoxHoTenNV.Text == "" || comboBoxGioiTinhNV.Text == "" || dateTimePickerNSNV.Value == null || textBoxDiaChiNV.Text == "" || textBoxSDTNV.Text == "" || textBoxEmailNV.Text == "" || textBoxCMNDNV.Text == "")

{

MessageBox.Show("Bạn phải nhập đầy đủ thông tin", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

else

{

dto.Manv = textboxMaNV.Text.Trim().ToString();

dto.Macv = textBoxMaCVNV.Text.Trim().ToString();

dto.Hoten = textBoxHoTenNV.Text.ToString();

dto.Ngaysinh = dateTimePickerNSNV.Value;

dto.Gioitinh = comboBoxGioiTinhNV.SelectedItem?.ToString();

dto.Diachi = textBoxDiaChiNV.Text.ToString();

dto.Sdt = textBoxSDTNV.Text.ToString();

dto.Email = textBoxEmailNV.Text.ToString();

dto.Cmnd = textBoxCMNDNV.Text.ToString();

BusinessLogicLayer.Instance.updateNhanvien(dto.Manv, dto.Macv, dto.Hoten, dto.Ngaysinh, dto.Gioitinh, dto.Sdt, dto.Cmnd, dto.Diachi, dto.Email);

MessageBox.Show("Thay đổi thông tin thành công!!!", "Thông báo", MessageBoxButtons.OK, MessageBoxIcon.Information);

loadDataGridViewNV();

}

}

catch(Exception ex)

{

MessageBox.Show("Xảy ra lỗi khi sửa nhân viên " + "\n" + ex);

}

}

private void dataGridViewListNhanVien\_RowEnter(object sender, DataGridViewCellEventArgs e)

{

int d = e.RowIndex;

if (dataGridViewListNhanVien.Rows[d].Cells[0].Value == null || dataGridViewListNhanVien.Rows[d].Cells[1].Value == null || dataGridViewListNhanVien.Rows[d].Cells[2].Value == null

|| dataGridViewListNhanVien.Rows[d].Cells[3].Value == null || dataGridViewListNhanVien.Rows[d].Cells[4].Value == null || dataGridViewListNhanVien.Rows[d].Cells[5].Value == null ||

dataGridViewListNhanVien.Rows[d].Cells[6].Value == null || dataGridViewListNhanVien.Rows[d].Cells[7].Value == null || dataGridViewListNhanVien.Rows[d].Cells[8].Value == null)

{

textboxMaNV.Text = "";

textBoxMaCVNV.Text = "";

textBoxHoTenNV.Text = "";

comboBoxGioiTinhNV.Text = "";

dateTimePickerNSNV.Text = "";

textBoxDiaChiNV.Text ="";

textBoxSDTNV.Text = "";

textBoxEmailNV.Text = ""; textBoxCMNDNV.Text = "";

textBoxCMNDNV.Text = "";

}

else

{

textboxMaNV.Text = dataGridViewListNhanVien.Rows[d].Cells[0].Value.ToString();

textBoxMaCVNV.Text = dataGridViewListNhanVien.Rows[d].Cells[1].Value.ToString();

textBoxHoTenNV.Text = dataGridViewListNhanVien.Rows[d].Cells[2].Value.ToString();

dateTimePickerNSNV.Text = dataGridViewListNhanVien.Rows[d].Cells[3].Value.ToString();

comboBoxGioiTinhNV.Text = dataGridViewListNhanVien.Rows[d].Cells[4].Value.ToString();

textBoxSDTNV.Text = dataGridViewListNhanVien.Rows[d].Cells[5].Value.ToString();

textBoxCMNDNV.Text = dataGridViewListNhanVien.Rows[d].Cells[6].Value.ToString();

textBoxDiaChiNV.Text = dataGridViewListNhanVien.Rows[d].Cells[8].Value.ToString();

textBoxEmailNV.Text = dataGridViewListNhanVien.Rows[d].Cells[7].Value.ToString();

}

}

private void buttonDel\_Click(object sender, EventArgs e)

{

if (textboxMaNV.TextLength == 0)

{

MessageBox.Show("Bạn cần chọn mã nhân viên để xóa!");

}

else

{

dto.Manv = textboxMaNV.Text;

BusinessLogicLayer.Instance.deleteNhanvien(dto.Manv);

MessageBox.Show("Đã xóa thành công!");

loadDataGridViewNV();

}

}

private void buttonTimKiemInfor\_Click(object sender, EventArgs e)

{

string noiDungTimKiem = textBoxNoiDung.Text.ToString();

string dieuKien = comboBoxSearchField.Text.ToString();

DataTable dtb = BusinessLogicLayer.Instance.serchNhanVien(dieuKien, noiDungTimKiem);

dataGridViewListNhanVien.DataSource = dtb;

}

}

}

### 2.2.5. Code formThongke:

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;

using System.Data.SqlClient;

using Microsoft.Reporting.WinForms;

namespace PMQLKS

{

public partial class FormThongke : Form

{

public FormThongke()

{

InitializeComponent();

}

private void FormThongke\_Load(object sender, EventArgs e)

{

this.rpvDoanhthu.RefreshReport();

}

private void btnTaobaocao\_Click(object sender, EventArgs e)

{

SqlConnection conn = new SqlConnection();

conn.ConnectionString = Properties.Settings.Default.HotelManagerNew1ConnectionString;

SqlCommand cmd = new SqlCommand();

cmd.CommandText = "Doanhthu1";

cmd.CommandType = CommandType.StoredProcedure;

cmd.Connection = conn;

cmd.Parameters.Add(new SqlParameter("@tungay", dtTungay.Value.Date));

cmd.Parameters.Add(new SqlParameter("@denngay", dtDenngay.Value.Date));

DataSet ds = new DataSet();

SqlDataAdapter da = new SqlDataAdapter(cmd);

da.Fill(ds);

rpvDoanhthu.ProcessingMode = ProcessingMode.Local;

rpvDoanhthu.LocalReport.ReportPath = "RptDoanhthu1.rdlc";

if(ds.Tables[0].Rows.Count > 0)

{

ReportDataSource rds = new ReportDataSource();

rds.Name = "DsDoanhthu1";

rds.Value = ds.Tables[0];

rpvDoanhthu.LocalReport.DataSources.Clear();

rpvDoanhthu.LocalReport.DataSources.Add(rds);

rpvDoanhthu.RefreshReport();

}

}

}

}

# PHẦN 3: KẾT LUẬN

Sau thời gian thực hiện dưới sự hướng dẫn của thầy Đỗ Ngọc Sơn, chúng em đã hoàn thành đề tài “Xây dựng phần mềm quản lý khách sạn và đặt phòng”. Mặc dù gặp không ít khó khăn trong quá trình tìm hiểu về đề tài, qua đó đã giúp em trau dồi thêm kiến thức và kinh nghiệm lập trình.

Trong quá trình thực hiện đề tài, chúng em đã thu thập được những kết quả như sau:

* Nắm được kiến thức về C#.
* Nắm được cách thiết kế ứng dụng với C#.
* Nắm được mô hình 3 lớp MVC.
* Tìm hiểu và nắm được cách kết nối CDSL.
* Nắm được kĩ thuật LinQ to SQL.

Các Chức Năng Đã Thực Hiện Được:

* Chức năng Đăng Nhập.
* Chức năng Quản Lý Khách hàng.
* Chức năng Quản Lý Phòng.
* Chức năng Đặt phòng.
* Chức năng Quản Lý Dịch vụ.
* Chức năng Quản Lý Hóa đơn.
* Chức năng Quản Lý Nhân viên.
* Chức năng Quản Lý Tài khoản.
* Chức năng Báo cáo thống kê.
* Chức năng Đăng Xuất.

Do kiến thức còn hạn chế, nên trong ứng dụng không tránh khỏi những thiếu sót. Trong thời gian tới, chúng em sẽ cố gắng tìm hiểu thêm về ngôn ngữ C# để có thể phát triển chương trình đa dạng và thân thiện hơn.

# TÀI LIỆU THAM KHẢO

1. Đỗ Ngọc Sơn, Vũ Thị Tâm, Vũ Đức Huy (2016), Giáo Trình Lập Trình Windows, Trường Đại Học Công Nghiệp Hà Nội.
2. Phương Lan (2008), Lập Trình Windows với C#.Net, Nhà Xuất Bản Lao Động-Xã Hội.
3. Nguyễn Ngọc Bình Phương, Thái Thanh Phong (2006), Các Giải Pháp Lập Trình C#, Nhà Xuất Bản Giao Thông Vân Tải.