**TRUỜNG CAO ĐẲNG THỰC HÀNH**

**FPT-POLYTECHNIC ĐÀ NẴNG**



**Dự Án 1**

**Quản Lý Nhà Hàng**

**Lớp : PT12301 Nhóm: 02**

**Các thành viên**

**Nguyễn Minh Thức**

**Võ Quan Thành**

**GVHD: Nguyễn Ngọc Anh**

***TP. Đà Nẵng 11- 2021***

**A. Lý do chọn dự án:**

Trong thời đại ngày nay, Công nghệ thông tin đang được phát triển mạnh mẽ, nhanh chóng và xâm nhập vào nhiều lĩnh vực khoa học, kỹ thuật cũng như trong cuộc sống. Nó trở thành công cụ đắc lực trong nhiều ngành nghề như giao thông, quân sự, y học..... và đặc biệt trong công tác quản lý nói chung và Quản Lý Nhà Hàng nói riêng. Trước đây khi máy tính chưa được ứng dụng rộng rãi, các công việc Quản Lý Nhà Hàng đều được làm thủ công nên rất mất thời gian và tốn kém về nhân lực cũng như tài chính. Ngày nay, với sự phát tri ển công nghệ thông tin mà máy tính đã được sử dụng rộng rãi trong các cơ quan, nhà máy, trường học... giúp cho công việc được tốt hơn. Việc sử dụng máy tính vào công tác Quản Lý Nhà Hàng là một yêu cầu cần thiết nhằm xóa bỏ những phương pháp lạc hậu, lỗi thời gây tốn kém về nhiều mặt. chúng em đã xây dựng đề tài “Quản Lý Nhà Hàng” với mong muốn giúp cho việc quản lý được dễ dàng, thuận tiện và tránh sai sót.

**B. Mục tiêu của dự án**

**Quản lý được các thông tin của nhân viên, bảng lương của nhân viên và lương của quản lý, nhân viên có đi làm đầy đủ, hoặc vắng mặt những ai, phân công các công việc cho nhân viên, các chi phí của nhà hàng, các bảng thống kê, báo cáo, thu nhập của nhà hàng**

**C. Các chức năng của dự án:**

**II Use**

+ Đăng nhập, đăng xuất

+ Thêm mới user, xóa user, vô hiệu hóa user

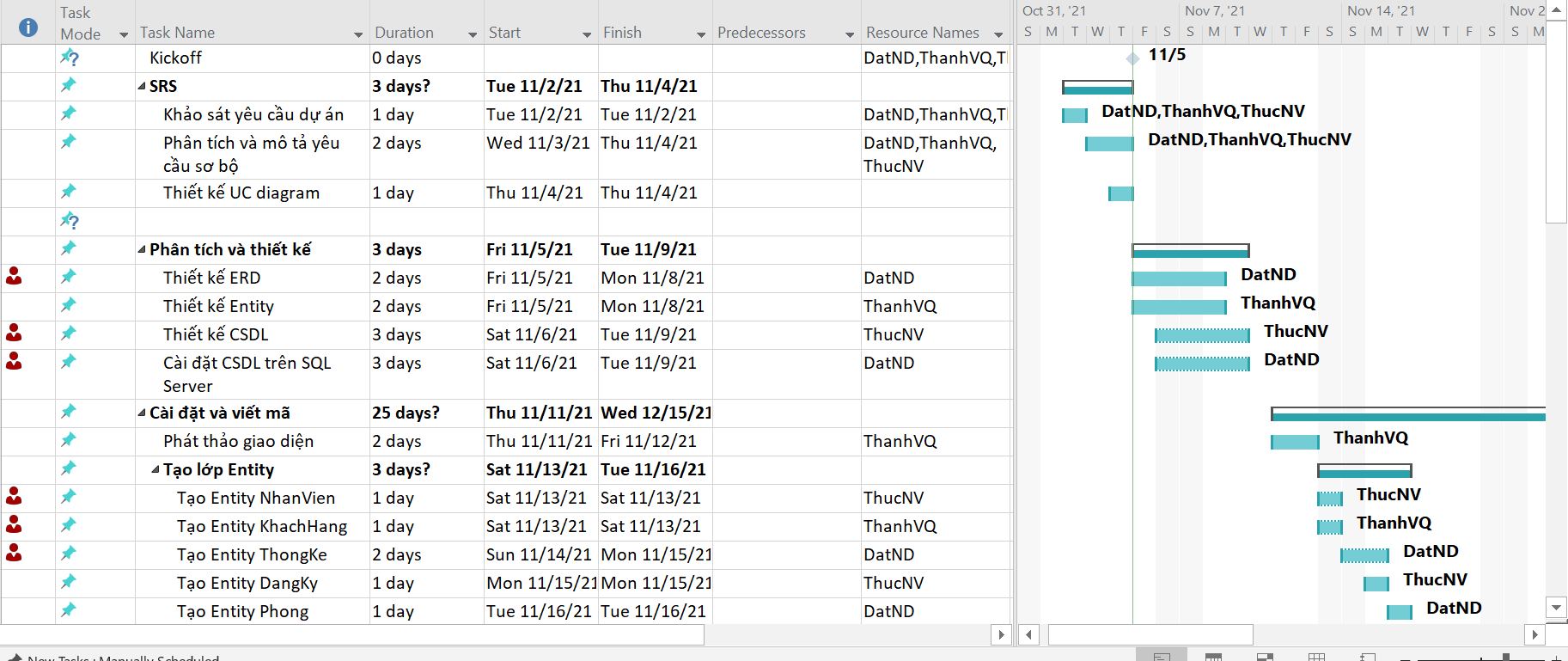
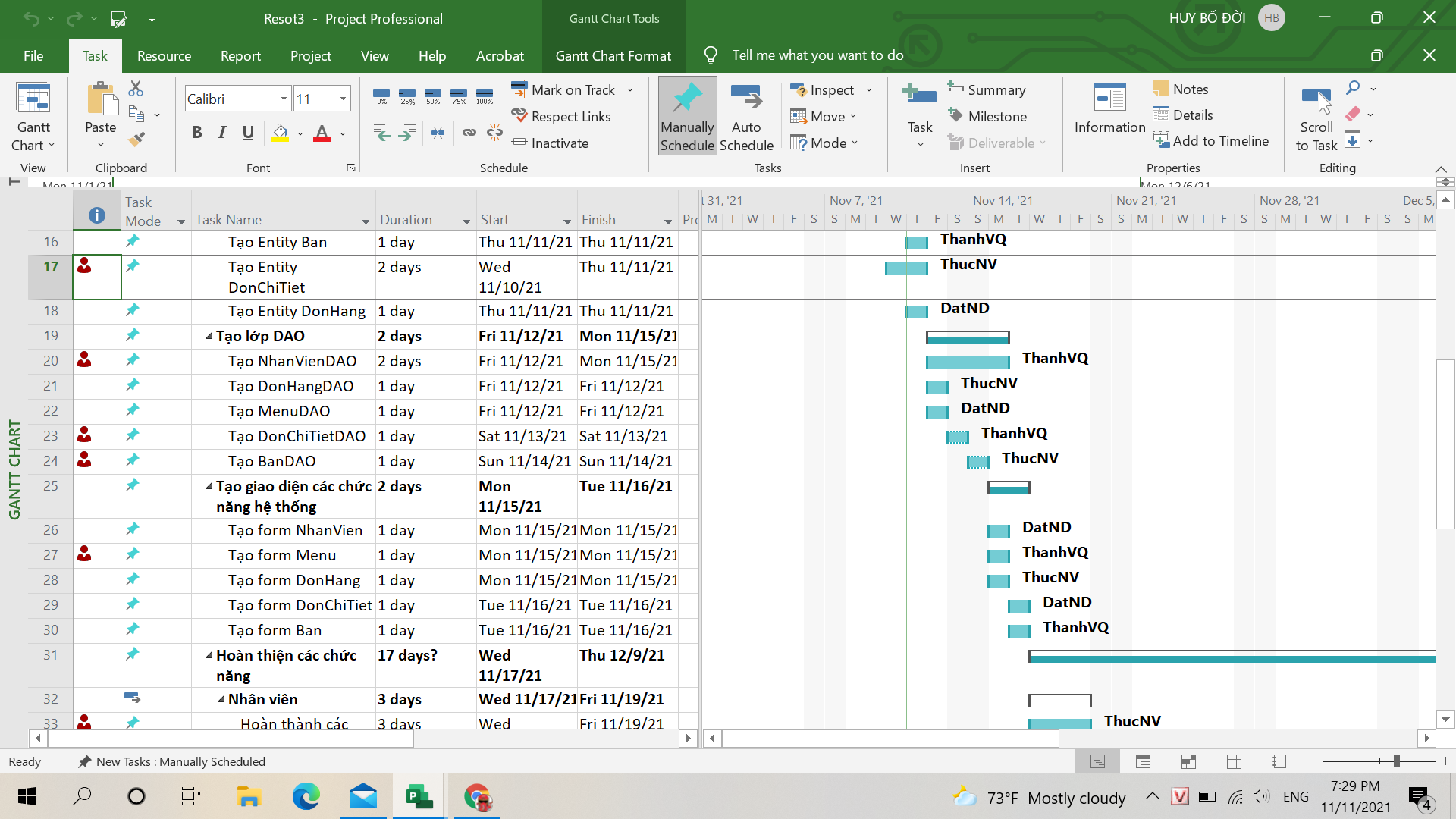
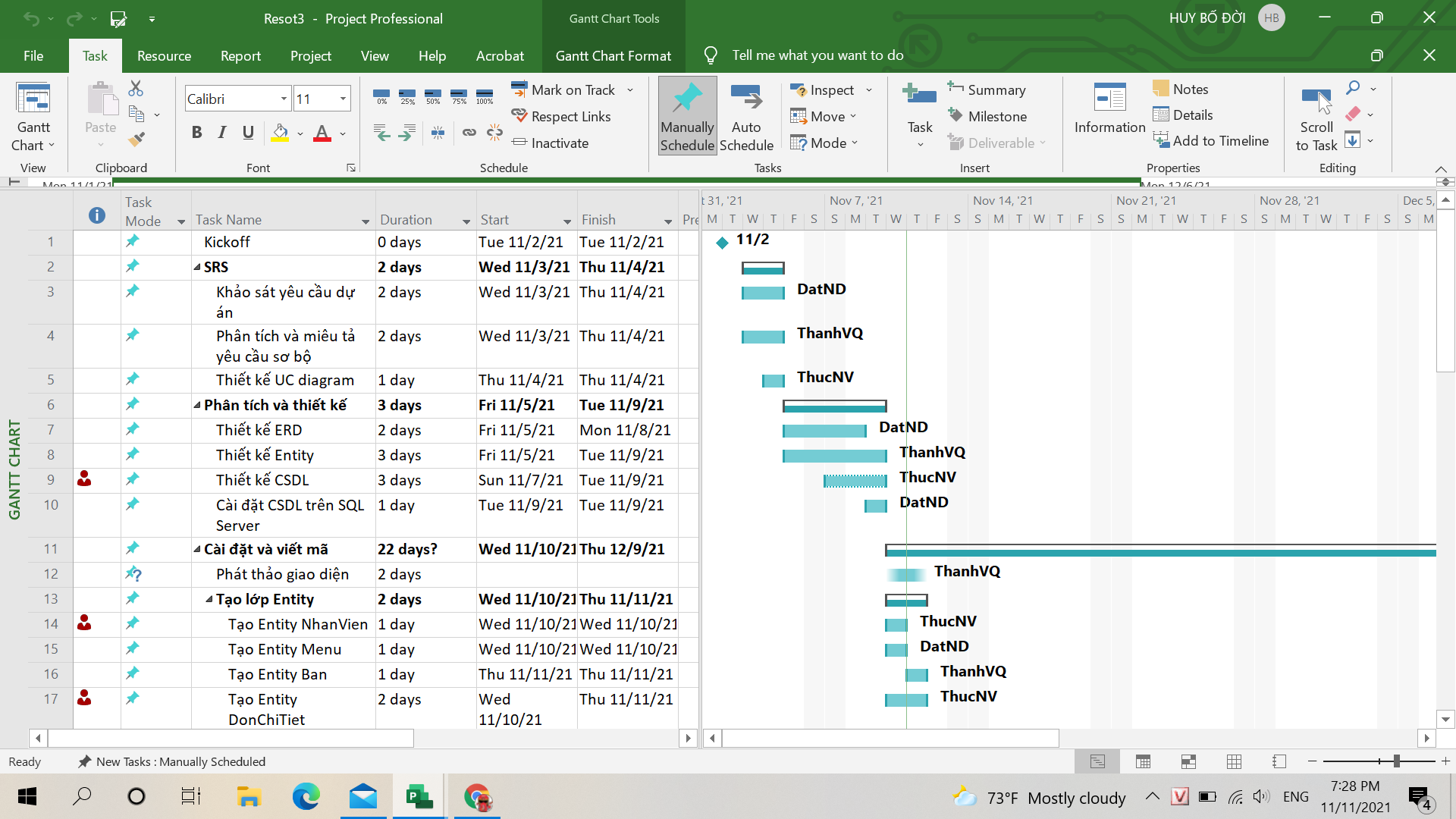
+ Thêm, xóa, sửa, làm mới

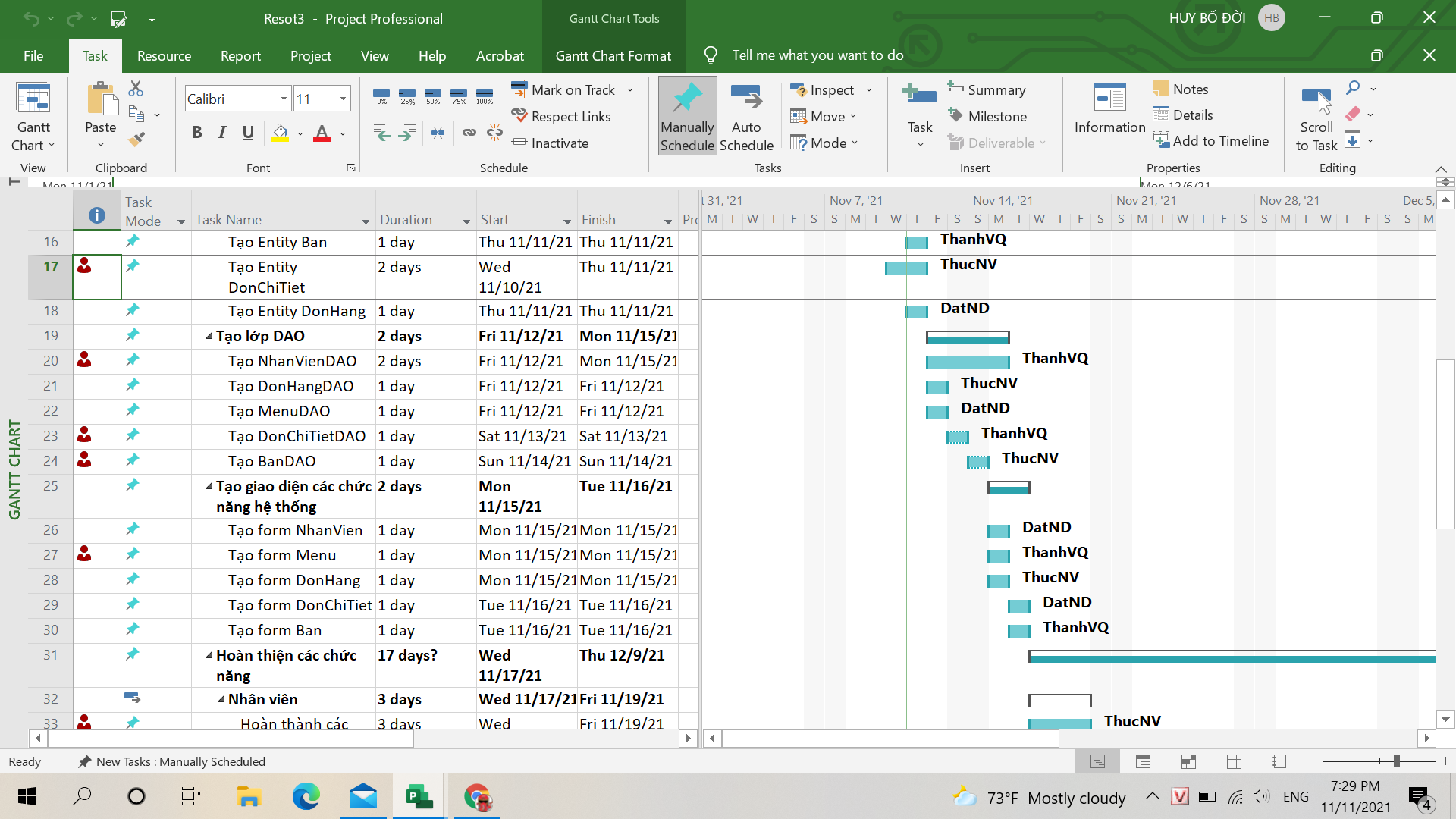
+ Tính toán bảng lương

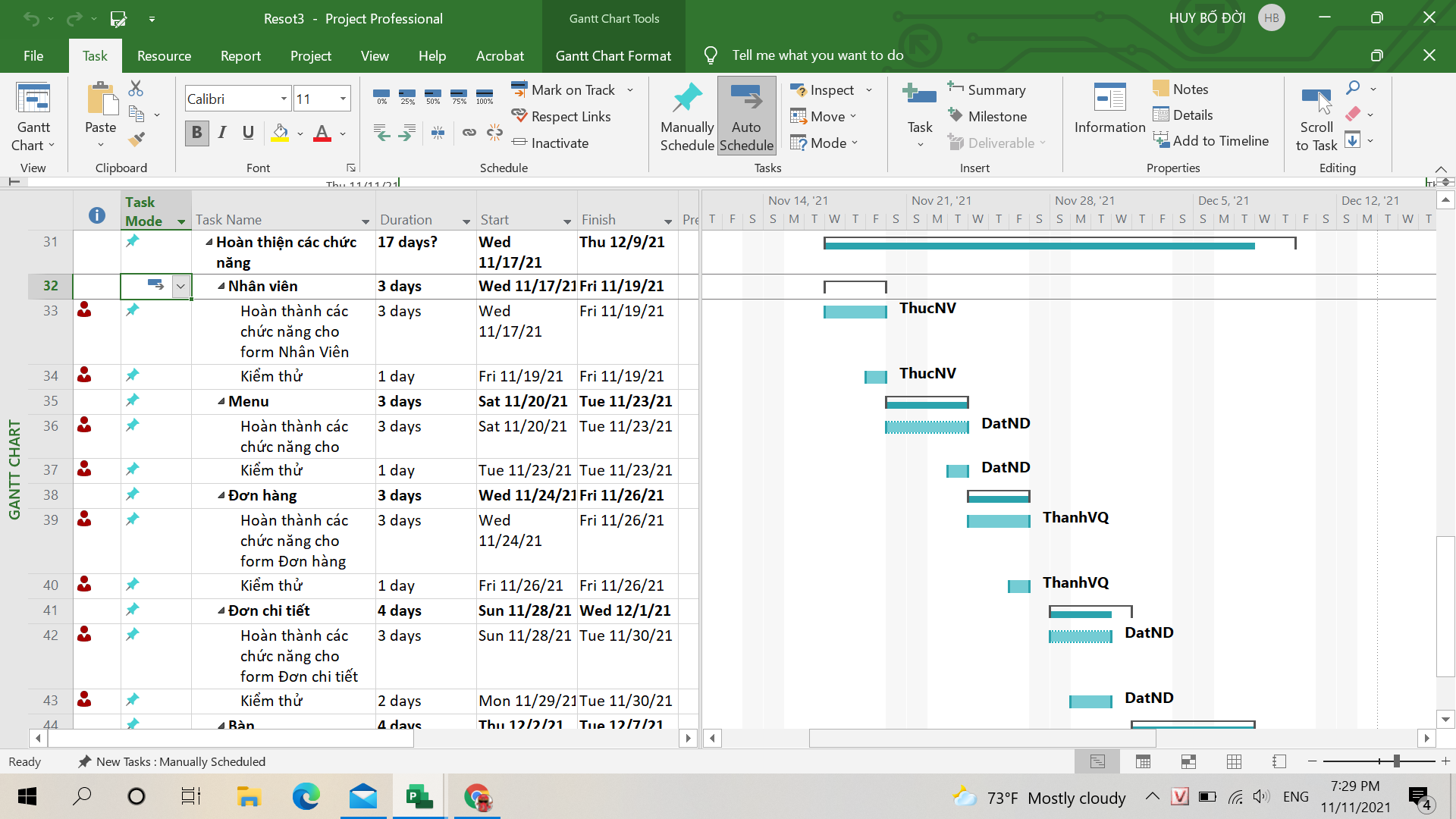
+ Phân công công việc cho nhan vien

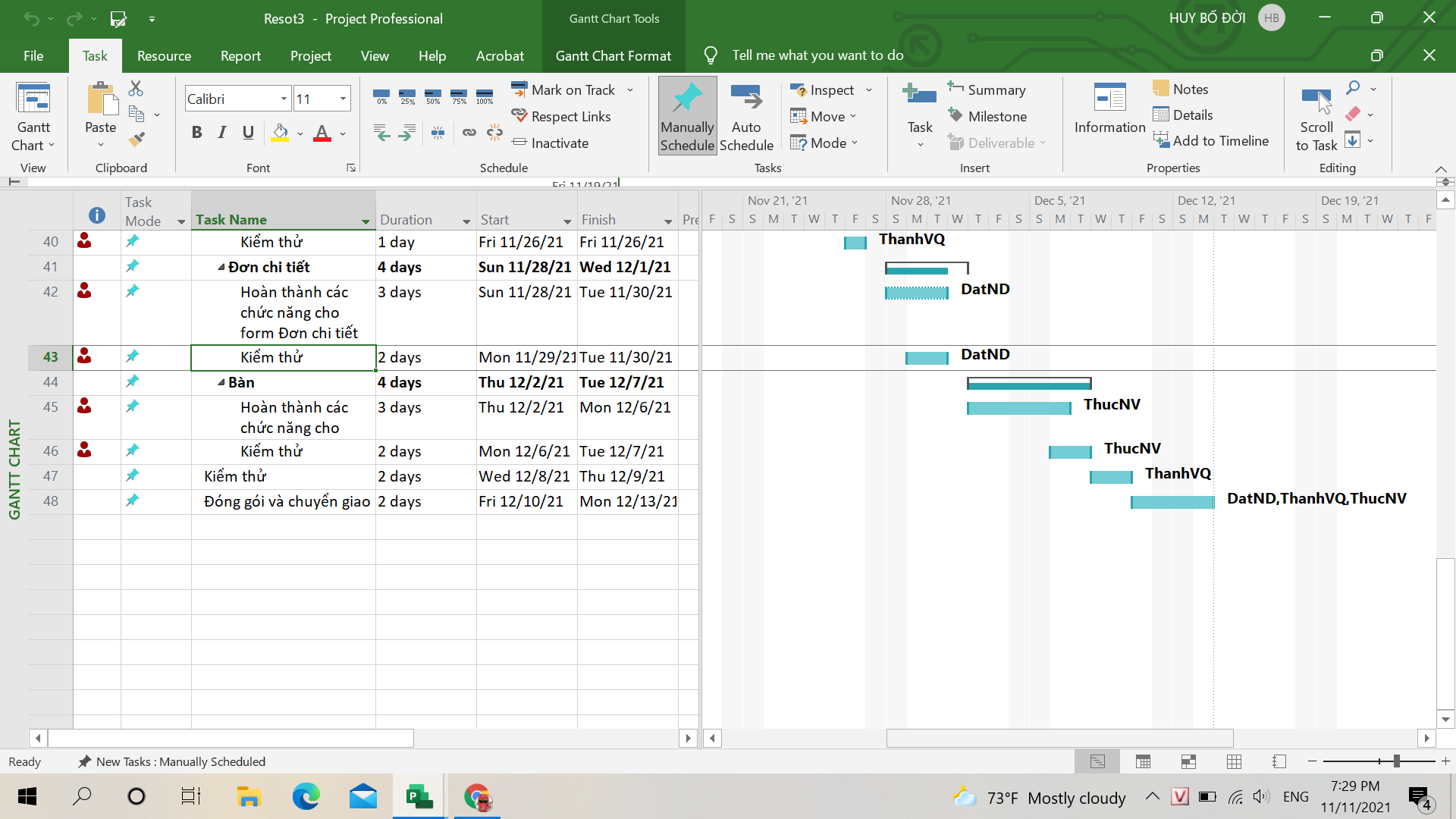
+ In báo cáo các bảng thống kê

II Quản Ly Thời Gian

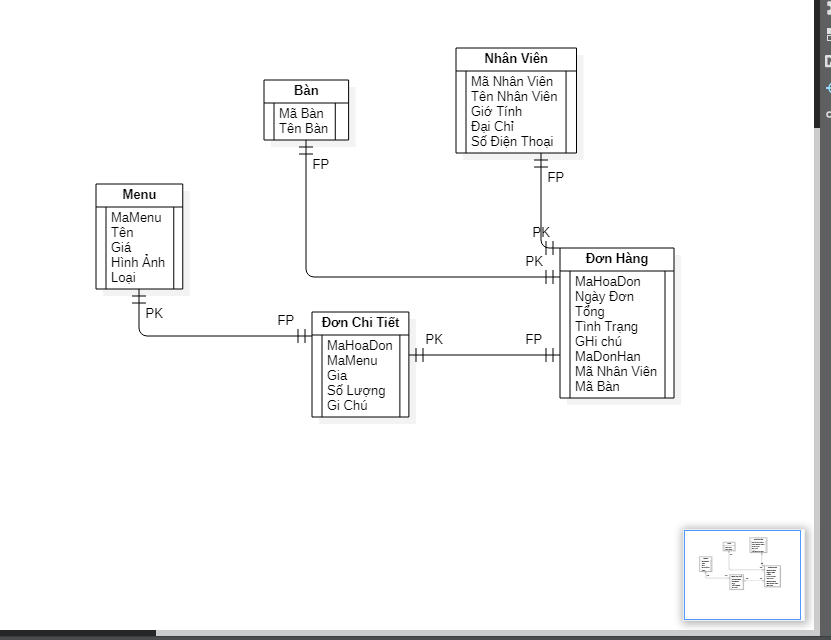








III Sơ Đồ ERD

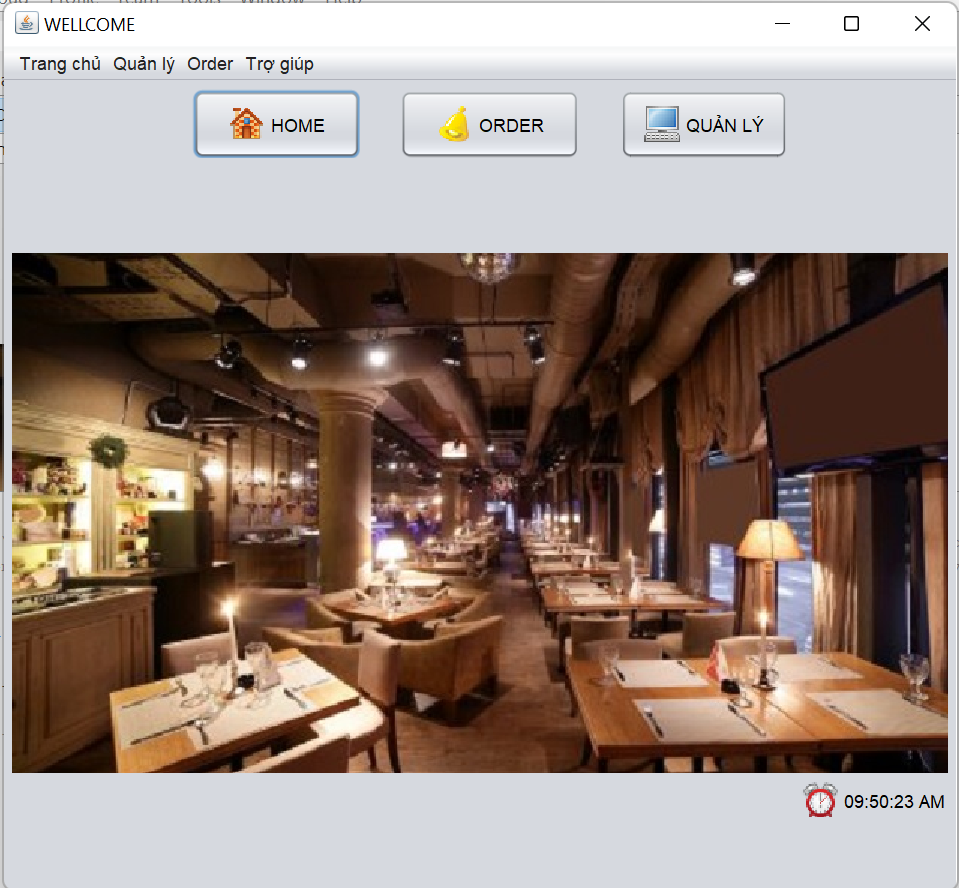


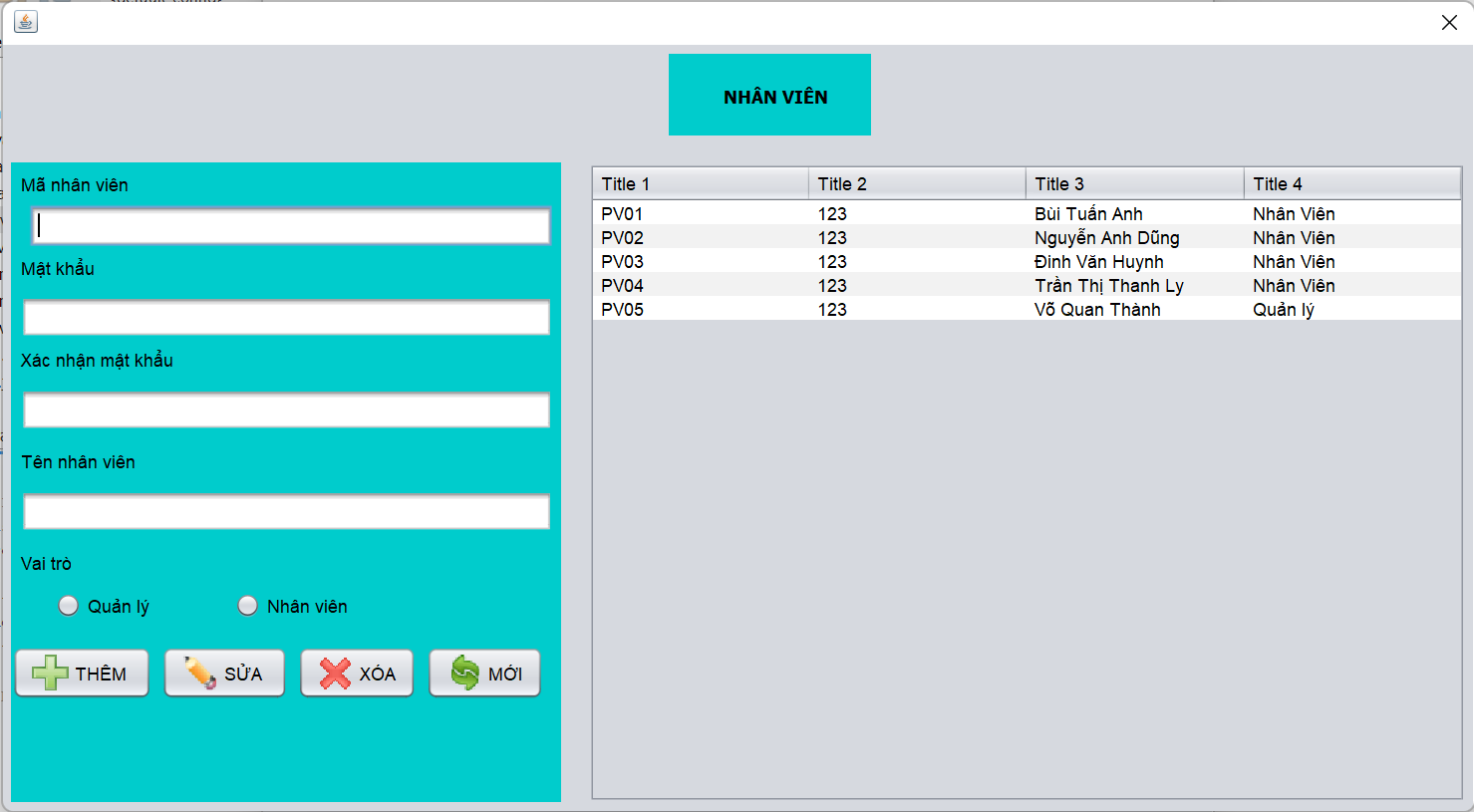
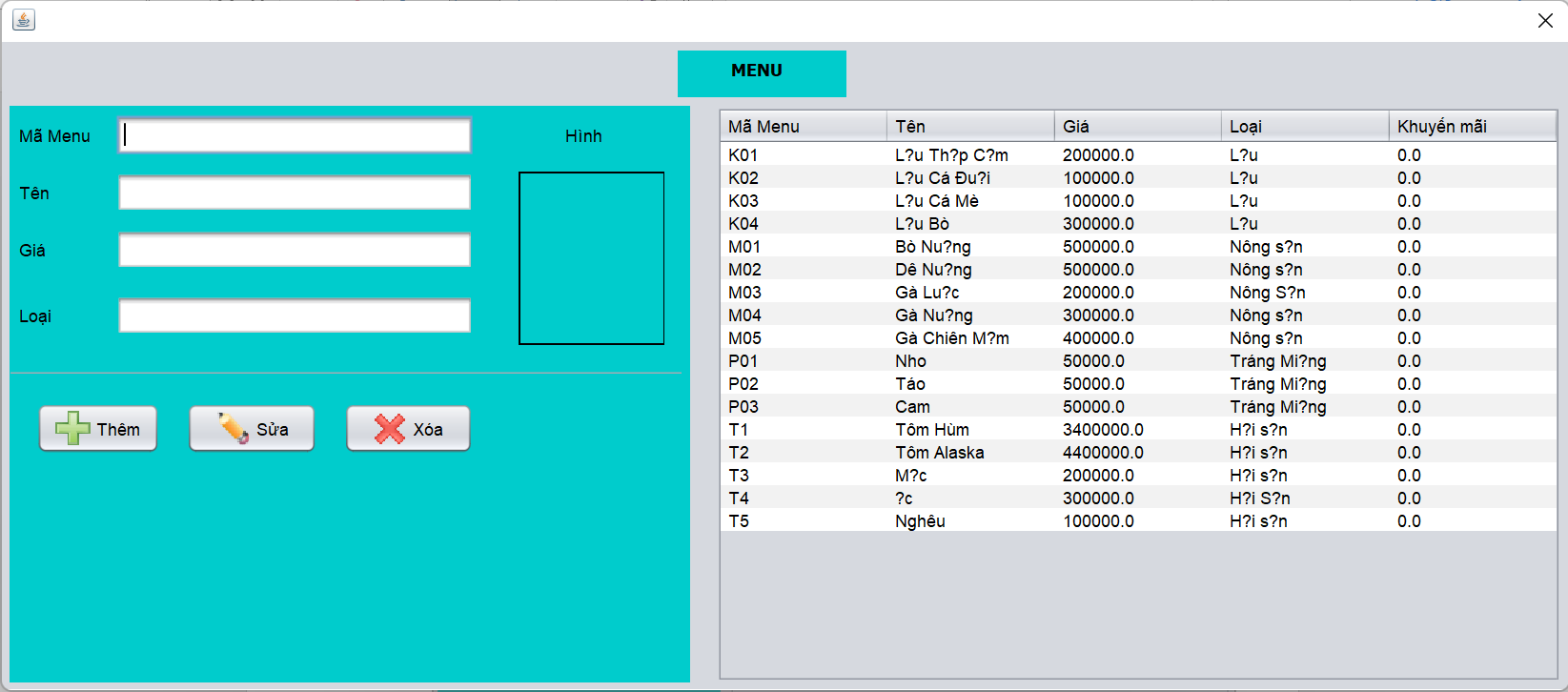
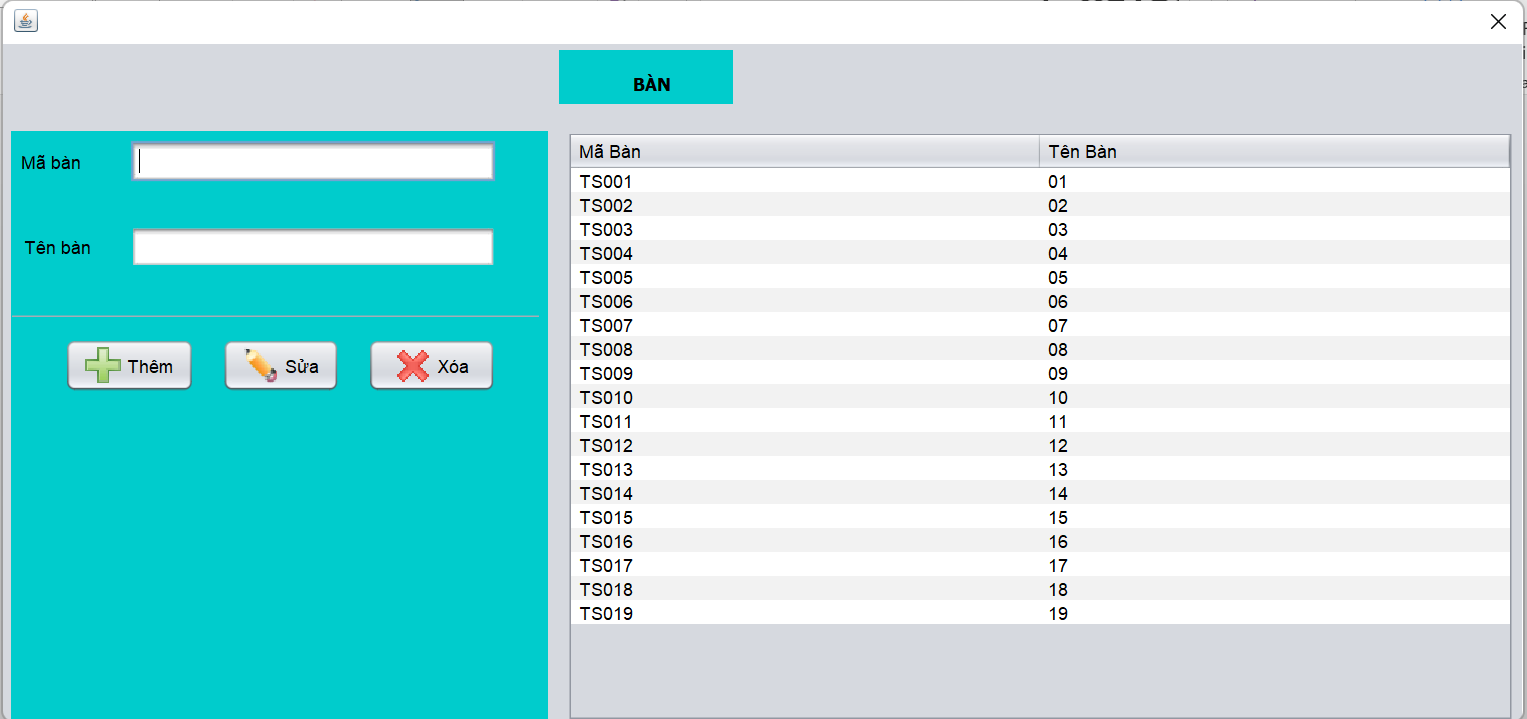
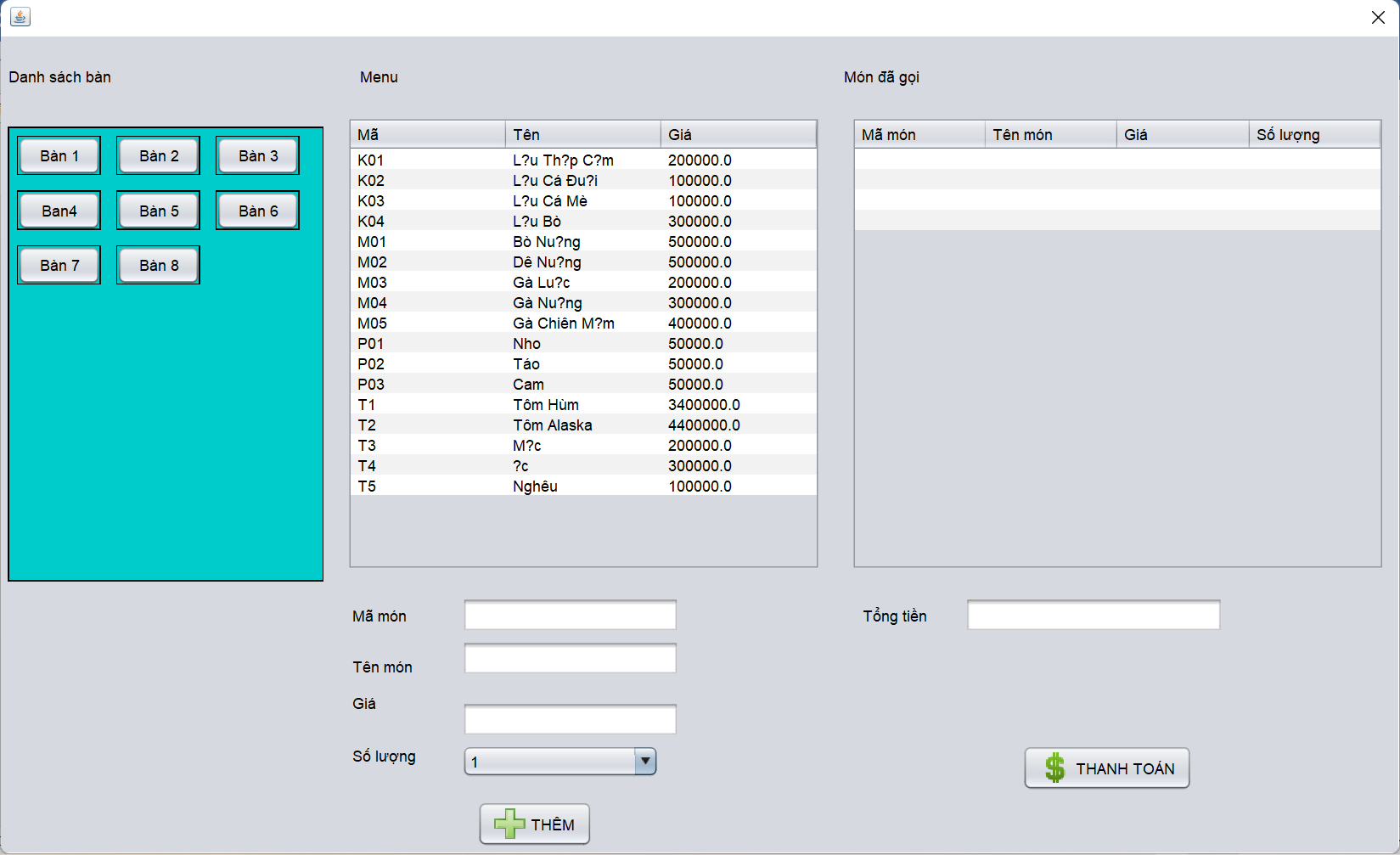
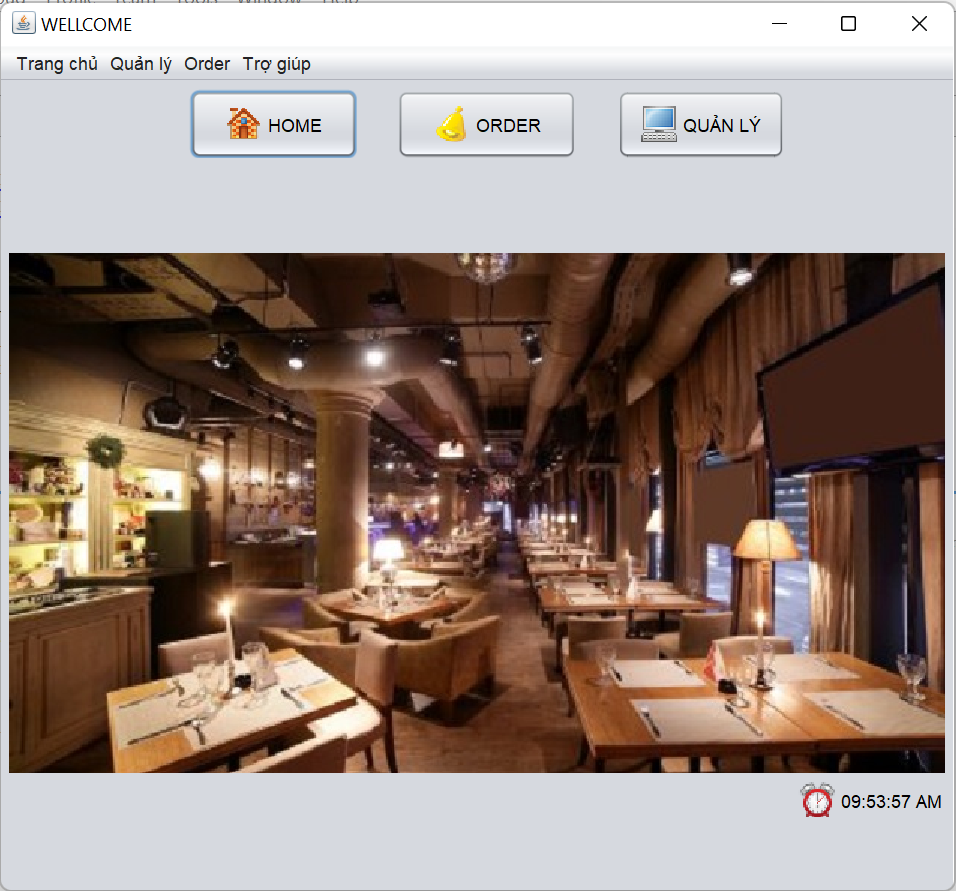
IV.Vẽ Sơ Đồ Phần Mền

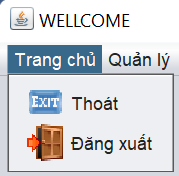
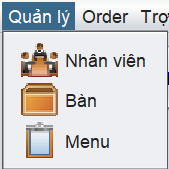
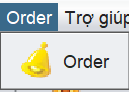
1. Đăng nhập



1. Giao diện main

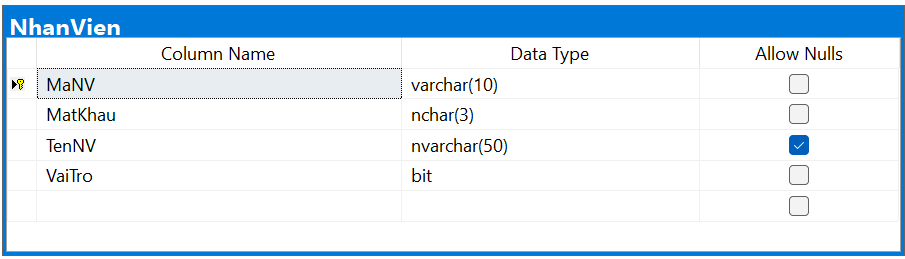


1. Form Nhân viên
2. Form Menu
3. Form Ban
4. Form Order
5. Form Thống Kê
6. Trợ giúp
7. Giao diện chính
8. Các thanh Menu

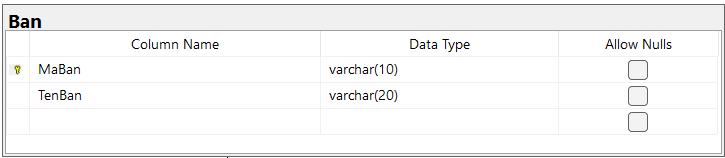
   

V. Cấu trúc cơ sở dữ liệu

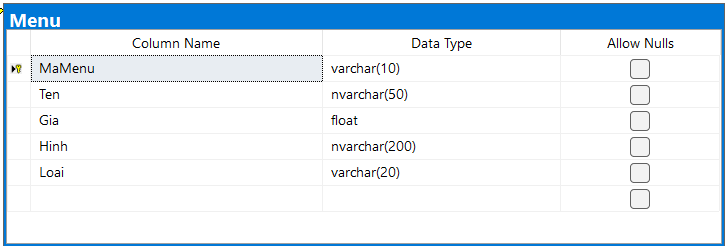
1.Thực thể nhân viên



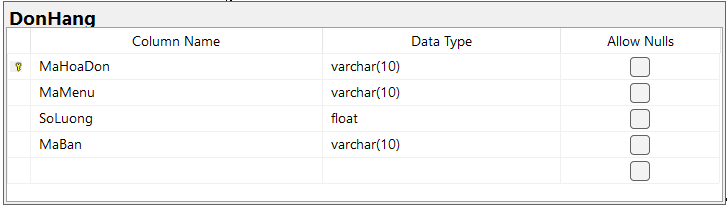
2.Thực thể Bàn



3.Thực thể Menu



4.Thực thể Đơn hàng



VI. Tạo cơ sở dữ liệu

1.Nhân viên

GO

CREATE TABLE [dbo].[NhanVien](

[MaNV] [varchar](10) NOT NULL,

[MatKhau] [nchar](3) NOT NULL,

[TenNV] [nvarchar](50) NULL,

[VaiTro] [bit] NOT NULL,

CONSTRAINT [PK\_NhanVien] PRIMARY KEY CLUSTERED

2.Menu

GO

CREATE TABLE [dbo].[Menu](

[MaMenu] [varchar](10) NOT NULL,

[Ten] [nvarchar](50) NOT NULL,

[Gia] [float] NOT NULL,

[Hinh] [nvarchar](200) NOT NULL,

[Loai] [varchar](20) NOT NULL,

CONSTRAINT [PK\_Menu] PRIMARY KEY CLUSTERED

3.Bàn

GO

CREATE TABLE [dbo].[Ban](

[MaBan] [varchar](10) NOT NULL,

[TenBan] [varchar](20) NOT NULL,

CONSTRAINT [PK\_Ban] PRIMARY KEY CLUSTERED

4.Đơn hàng

GO

CREATE TABLE [dbo].[DonHang](

[MaHoaDon] [varchar](10) NOT NULL,

[MaMenu] [varchar](10) NOT NULL,

[SoLuong] [float] NOT NULL,

[MaBan] [varchar](10) NOT NULL,

CONSTRAINT [PK\_DonHang] PRIMARY KEY CLUSTERED

XII. SQL TRUY VẤN VÀ THAO TÁC

1.Nhân viên

|  |  |
| --- | --- |
|  |  |
| **Hành động** | **Câu lệnh SQL** |
|  |  |
| Thêm mới | INSERT INTO NhanVien(MaNV,MatKhau,TenNV,VaiTro) values(?,?,?,?) |
|  |  |
| Cập nhật theo mã | UPDATE NhanVien set MatKhau=?,TenNV=?,VaiTro=? WHERE MaNV =? |
|  |  |
| Xóa theo mã | DELETE FROM NhanVien WHERE MaNV=? |
|  |  |
| Truy vấn tất cả | SELECT \* FROM NhanVien |
|  |  |
| Truy vấn theo mã | SELECT \* FROM NhanVien WHERE MaNV=? |
|  |  |

2.Bàn

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
|  |  |
| Thêm mới | INSERT INTO Ban(MaBan,TenBan) values(?,?) |
|  |  |
|  |  |
| Cập nhật theo mã | UPDATE Ban set TenBan=? WHERE MaBan =? |
|  |  |
| Xóa theo mã | DELETE FROM Ban WHERE MaBan=? |
|  |  |
| Truy vấn tất cả | SELECT \* FROM Ban |
|  |  |
| Truy vấn theo mã | SELECT \* FROM Ban WHERE MaBan=? |
|  |  |

3.Menu

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
|  |  |
| Thêm mới | INSERT INTO Menu(MaMenu,Ten,Gia,Hinh,Loai) values(?,?,?,?,?) |
|  |  |
|  |  |
| Cập nhật theo mã | UPDATE Menu set Ten=?, Gia=?, Hinh=?,Loai=? WHERE MaMenu =? |
|  |  |
| Xóa theo mã | DELETE FROM Menu WHERE MaMenu=? |
|  |  |
| Truy vấn tất cả | SELECT \* FROM ChuyenDe |
|  |  |
| Truy vấn theo mã | SELECT \* FROM Menu WHERE MaMenu =? |
|  |  |

4.Order

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
|  |  |
| Thêm mới | INSERT INTO DonHang(MaHoaDon,MaMenu,SoLuong,MaBan) values (?,?,?,?,?) |
|  |  |
|  |  |
| Cập nhật theo mã | UPDATE Order set MaMenu=?,SoLuong=?,MaBan=? where MaHoaDon=? |
|  |  |
| Xóa theo mã | DELETE FROM Order WHERE MaHoaDon=? |
|  |  |
| Truy vấn tất cả | SELECT \* FROM Order |
|  |  |
| Truy vấn theo mã | SELECT \* FROM Order WHERE MaHoaDon=? |
|  |  |

**XIII. CÁC LỚP HỖ TRỢ**

**1.Nhân viên DAO**

package DAO;

import Entity.NV;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import uitil.jdbcHelper;

/\*\*

\*

\* @author PC

\*/

public class NhanVienDAO extends HomeDAO<NV, String>{

final String INSERT\_SQL="INSERT INTO NhanVien(MaNV,MatKhau,TenNV,VaiTro) values(?,?,?,?)";

final String UPDATE\_SQL= "UPDATE NhanVien set MatKhau=?,TenNV=?,VaiTro=? WHERE MaNV =?";

final String DELETE\_SQL= "DELETE FROM NhanVien WHERE MaNV=?";

final String SELECT\_ALL\_SQL= "SELECT \* FROM NhanVien";

final String SELECT\_BY\_ID\_SQL= "SELECT \* FROM NhanVien WHERE MaNV =?";

@Override

public void insert(NV entity) {

jdbcHelper.update(INSERT\_SQL, entity.getMaNV(),entity.getMatKhau(),entity.getTenNV(),entity.isVaiTro());

}

@Override

public void update(NV entity) {

jdbcHelper.update(UPDATE\_SQL,entity.getMatKhau(),entity.getTenNV(),entity.isVaiTro(),entity.getMaNV());

}

@Override

public void delete(String id) {

jdbcHelper.update(DELETE\_SQL,id);

}

@Override

public List<NV> selectAll() {

return selectBySql(SELECT\_ALL\_SQL);

}

@Override

public NV selectById(String id) {

List<NV> list = selectBySql(SELECT\_BY\_ID\_SQL, id);

if(list.isEmpty()){

return null;

}

return list.get(0);

}

@Override

public List<NV> selectBySql(String sql, Object... args) {

List<NV> list = new ArrayList<>();

try {

ResultSet rs = jdbcHelper.query(sql, args);

while(rs.next()){

NV entity = new NV();

entity.setMaNV(rs.getString("MaNV"));

entity.setMatKhau(rs.getString("MatKhau"));

entity.setTenNV(rs.getString("TenNV"));

entity.setVaiTro(rs.getBoolean("VaiTro"));

list.add(entity);

}

} catch (Exception e) {

throw new RuntimeException(e);

}

return list;

}

}

**2.Menu DAO**

package DAO;

import Entity.MN;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import uitil.jdbcHelper;

import java.util.List;

/\*\*

\*

\* @author PC

\*/

public class MenuDAO extends HomeDAO<MN, String>{

final String INSERT\_SQL= "INSERT INTO Menu(MaMenu,Ten,Gia,Hinh,Loai) values(?,?,?,?,?)";

final String UPDATE\_SQL= "UPDATE Menu set Ten=?, Gia=?, Hinh=?,Loai=? WHERE MaMenu =?";

final String DELETE\_SQL= "DELETE FROM Menu WHERE MaMenu=?";

final String SELECT\_ALL\_SQL= "SELECT \* FROM Menu";

final String SELECT\_BY\_ID\_SQL= "SELECT \* FROM Menu WHERE MaMenu =?";

@Override

public void insert(MN entity) {

jdbcHelper.update(INSERT\_SQL, entity.getMaMenu(),entity.getTenMenu()

,entity.getGia(),entity.getHinh(),entity.getLoai());

}

@Override

public void update(MN entity) {

jdbcHelper.update(UPDATE\_SQL,entity.getTenMenu()

,entity.getGia(),entity.getHinh(),entity.getLoai(),entity.getMaMenu());

}

@Override

public void delete(String id) {

jdbcHelper.update(DELETE\_SQL,id);

}

@Override

public List<MN> selectAll() {

return selectBySql(SELECT\_ALL\_SQL);

}

@Override

public MN selectById(String id) {

List<MN> list = selectBySql(SELECT\_BY\_ID\_SQL, id);

if(list.isEmpty()){

return null;

}

return list.get(0);

}

@Override

public List<MN> selectBySql(String sql, Object... args) {

List<MN> list = new ArrayList<>();

try {

ResultSet rs = jdbcHelper.query(sql, args);

while(rs.next()){

MN entity = new MN();

entity.setMaMenu(rs.getString("MaMenu"));

entity.setTenMenu(rs.getString("Ten"));

entity.setGia(rs.getDouble("Gia"));

entity.setHinh(rs.getString("Hinh"));

entity.setLoai(rs.getString("Loai"));

list.add(entity);

}

} catch (Exception e) {

throw new RuntimeException(e);

}

return list;

}

}

**3.Bàn DAO**

package DAO;

import Entity.Table;

import Entity.Table;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import uitil.jdbcHelper;

/\*\*

\*

\* @author PC

\*/

public class TableDAO extends HomeDAO<Table, String>{

final String INSERT\_SQL="INSERT INTO Ban(MaBan,TenBan) values(?,?)";

final String UPDATE\_SQL= "UPDATE Ban set TenBan=? WHERE MaBan =?";

final String DELETE\_SQL= "DELETE FROM Ban WHERE MaBan=?";

final String SELECT\_ALL\_SQL= "SELECT \* FROM Ban";

final String SELECT\_BY\_ID\_SQL= "SELECT \* FROM Ban WHERE MaBan=?";

@Override

public void insert(Table entity) {

jdbcHelper.update(INSERT\_SQL,entity.getMaBan(),entity.getTenBan());

}

@Override

public void update(Table entity) {

jdbcHelper.update(UPDATE\_SQL,entity.getTenBan(),entity.getMaBan());

}

@Override

public void delete(String id) {

jdbcHelper.update(DELETE\_SQL,id);

}

@Override

public List<Table> selectAll() {

return selectBySql(SELECT\_ALL\_SQL);

}

@Override

public Table selectById(String id) {

List<Table> list = selectBySql(SELECT\_BY\_ID\_SQL, id);

if(list.isEmpty()){

return null;

}

return list.get(0);

}

@Override

public List<Table> selectBySql(String sql, Object... args) {

List<Table> list = new ArrayList<>();

try {

ResultSet rs = jdbcHelper.query(sql, args);

while(rs.next()){

Table entity = new Table();

entity.setMaBan(rs.getString("MaBan"));

entity.setTenBan(rs.getString("TenBan"));

list.add(entity);

}

} catch (Exception e) {

throw new RuntimeException(e);

}

return list;

}

}

**4.Order DAO**

package DAO;

import Entity.GoiMon;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import uitil.jdbcHelper;

/\*\*

\*

\* @author PC

\*/

public class OrderDAO extends HomeDAO<GoiMon, String>{

final String INSERT\_SQL= "INSERT INTO DonHang(MaHoaDon,MaMenu,SoLuong,MaBan) values (?,?,?,?,?)";

final String UPDATE\_SQL= "UPDATE Order set MaMenu=?,SoLuong=?,MaBan=? where MaHoaDon=?";

final String DELETE\_SQL= "DELETE FROM Order where MaHoaDon=?";

final String SELECT\_ALL\_SQL= "SELECT \* FROM Order";

final String SELECT\_BY\_ID\_SQL= "SELECT \* FROM Order WHERE MaHoaDon =?";

final String SELECT\_KH\_SQL= "SELECT \* FROM Order WHERE MaMon =?";

@Override

public void insert(GoiMon entity) {

jdbcHelper.update(INSERT\_SQL,entity.getMaMenu(),entity.getSoLuong(),entity.getMaBan());

}

@Override

public void update(GoiMon entity) {

jdbcHelper.update(UPDATE\_SQL,entity.getMaMenu(),entity.getSoLuong(),entity.getMaBan(),entity.getMaHoaDon());

}

@Override

public void delete(String id) {

jdbcHelper.update(DELETE\_SQL,id);

}

@Override

public List<GoiMon> selectAll() {

return selectBySql(SELECT\_ALL\_SQL);

}

@Override

public GoiMon selectById(String id) {

List<GoiMon> list = selectBySql(SELECT\_BY\_ID\_SQL,id);

if(list.isEmpty()){

return null;

}

return list.get(0);

}

@Override

public List<GoiMon> selectBySql(String sql, Object... args) {

List<GoiMon> list = new ArrayList<>();

try {

ResultSet rs = jdbcHelper.query(sql, args);

while(rs.next()){

GoiMon entity = new GoiMon();

entity.setMaHoaDon(rs.getString("MaHoaDon"));

entity.setMaMenu(rs.getString("MaMenu"));

entity.setSoLuong(rs.getDouble("SoLuong"));

entity.setMaBan(rs.getString("MaBan"));

list.add(entity);

}

} catch (Exception e) {

throw new RuntimeException(e);

}

return list;

}

}

**5.Nhân viên Entity**

package Entity;

/\*\*

\*

\* @author PC

\*/

public class NV {

private String maNV;

private String TenNV;

private String matKhau;

private boolean vaiTro;

public NV() {

}

public NV(String maNV, String TenNV, String matKhau, boolean vaiTro) {

this.maNV = maNV;

this.matKhau = matKhau;

this.TenNV = TenNV;

this.vaiTro = vaiTro;

}

public String getMaNV() {

return maNV;

}

public void setMaNV(String maNV) {

this.maNV = maNV;

}

public String getMatKhau() {

return matKhau;

}

public void setMatKhau(String matKhau) {

this.matKhau = matKhau;

}

public String getTenNV() {

return TenNV;

}

public void setTenNV(String TenNV) {

this.TenNV = TenNV;

}

public boolean isVaiTro() {

return vaiTro;

}

public void setVaiTro(boolean vaiTro) {

this.vaiTro = vaiTro;

}

}

**6.Bàn Entity**

package Entity;

/\*\*

\*

\* @author PC

\*/

public class Table {

private String MaBan;

private String TenBan;

public Table(String MaBan, String TenBan) {

this.MaBan = MaBan;

this.TenBan = TenBan;

}

public Table() {

}

public String getMaBan() {

return MaBan;

}

public void setMaBan(String MaBan) {

this.MaBan = MaBan;

}

public String getTenBan() {

return TenBan;

}

public void setTenBan(String TenBan) {

this.TenBan = TenBan;

}

}

**7.Menu Entity**

package Entity;

/\*\*

\*

\* @author PC

\*/

public class MN {

private String MaMenu;

private String TenMenu;

private double Gia;

private String Hinh;

private String Loai;

private double khuyenmai;

public MN() {

}

public MN(String MaMenu, String TenMenu, double Gia, String Hinh, String Loai, double khuyenmai) {

this.MaMenu = MaMenu;

this.TenMenu = TenMenu;

this.Gia = Gia;

this.Hinh = Hinh;

this.Loai = Loai;

this.khuyenmai = khuyenmai;

}

public String getMaMenu() {

return MaMenu;

}

public void setMaMenu(String MaMenu) {

this.MaMenu = MaMenu;

}

public String getTenMenu() {

return TenMenu;

}

public void setTenMenu(String TenMenu) {

this.TenMenu = TenMenu;

}

public double getGia() {

return Gia;

}

public void setGia(double Gia) {

this.Gia = Gia;

}

public String getHinh() {

return Hinh;

}

public void setHinh(String Hinh) {

this.Hinh = Hinh;

}

public String getLoai() {

return Loai;

}

public void setLoai(String Loai) {

this.Loai = Loai;

}

public double getKhuyenmai() {

return khuyenmai;

}

public void setKhuyenmai(double khuyenmai) {

this.khuyenmai = khuyenmai;

}

}

**8.Order entity**

package Entity;

/\*\*

\*

\* @author PC

\*/

public class GoiMon {

private String MaHoaDon;

private String MaMenu;

private double Gia;

private double SoLuong;

private String MaBan;

private String TenMon;

public GoiMon(String MaHoaDon, String MaMenu, double Gia, double SoLuong, String MaBan, String TenMon) {

this.MaHoaDon = MaHoaDon;

this.MaMenu = MaMenu;

this.Gia = Gia;

this.SoLuong = SoLuong;

this.MaBan = MaBan;

this.TenMon = TenMon;

}

public GoiMon() {

}

public String getMaHoaDon() {

return MaHoaDon;

}

public void setMaHoaDon(String MaHoaDon) {

this.MaHoaDon = MaHoaDon;

}

public String getMaMenu() {

return MaMenu;

}

public void setMaMenu(String MaMenu) {

this.MaMenu = MaMenu;

}

public double getGia() {

return Gia;

}

public void setGia(double Gia) {

this.Gia = Gia;

}

public double getSoLuong() {

return SoLuong;

}

public void setSoLuong(double SoLuong) {

this.SoLuong = SoLuong;

}

public String getMaBan() {

return MaBan;

}

public void setMaBan(String MaBan) {

this.MaBan = MaBan;

}

public String getTenMon() {

return TenMon;

}

public void setTenMon(String TenMon) {

this.TenMon = TenMon;

}

}

**II.CÁC THƯ VIỆN TIỆN ÍCH**

1. **Auth**

package uitil;

import Entity.NV;

import static java.awt.Color.pink;

import static java.awt.Color.white;

import java.awt.Component;

import javax.swing.JOptionPane;

import javax.swing.JPasswordField;

import javax.swing.JTextField;

/\*\*

\*

\* @author PC

\*/

public class Auth {

/\*\*

\* Đối tượng này chứa thông tin người sử dụng khi đăng nhập

\*/

public static NV user = null;

/\*\*

\* Xóa thông tin người sử dụng khi có yêu cầu đăng xuất

\*/

public static void clear(){

Auth.user = null;

}

/\*\*

\* Kiểm tra xem đăng nhập hay chưa

\* @return đăng nhập hay chưa

\*/

public static boolean isLogin(){

return Auth.user !=null;

}

public static boolean isManager(){

return Auth.isLogin() && user.isVaiTro();

}

/\*\*

\* Hiển thị thông báo cho người dùng

\* @param parent là cửa sổ chứa thông báo

\* @param message là thông báo

\*/

public static void alert(Component parent, String message) {

JOptionPane.showMessageDialog(parent, message,

"Hệ thống quản lý đào tạo", JOptionPane.INFORMATION\_MESSAGE);

}

public static boolean checkNullText(JTextField txt) {

txt.setBackground(white);

if (txt.getText().trim().length() > 0) {

return true;

} else {

txt.setBackground(pink);

Auth.alert(txt.getRootPane(), "Không được để trống " + txt.getName());

return false;

}

}

public static boolean checkNullPass(JPasswordField txt) {

txt.setBackground(white);

if (txt.getPassword().length > 0) {

return true;

} else {

txt.setBackground(pink);

Auth.alert(txt.getRootPane(), "Không được để trống " + txt.getName());

return false;

}

}

}

1. **MsgBox**

package uitil;

import java.awt.Component;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author PC

\*/

public class MsgBox {

/\*\*

\* Hien thi thong bao cho nguoi dung

\* @param parent la cua so chua thong bao

\* @param message la thong bao

\*/

public static void alert(Component parent, String message){

JOptionPane.showMessageDialog(parent, message,"Hệ thống quản lý đào tạo ", JOptionPane.INFORMATION\_MESSAGE);

}

/\*\*

\* Hien thi thong bao va yeu cau nguoi dung xac nhan

\* @param parent la cua so chua thong bao

\* @param message la cau hoi yes/no

\* return la ket qua nhan dc

\*/

public static boolean confirm(Component parent, String message){

int result = JOptionPane.showConfirmDialog(parent, message,

"Hệ thống quản lý đào tạo",

JOptionPane.YES\_NO\_OPTION,JOptionPane.QUESTION\_MESSAGE);

return result == JOptionPane.YES\_OPTION;

}

/\*\*

\* Hiển thị thông báo yêu cầu nhập dữ liệu

\* @param parent la cửa sổ thông báo

\* @param message la thong bao nhac nhở đăng nhập

\* @return la ket qua nhan được tu người sử dụng nhập vào

\*/

public static String prompt(Component parent, String message){

return JOptionPane.showInputDialog(parent, message,"Hệ thống quản lý đào tạo",

JOptionPane.INFORMATION\_MESSAGE);

}

}

1. **Validator**

package uitil;

import java.awt.Color;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.swing.JTextField;

/\*\*

\*

\* @author PC

\*/

public class Validator {

public static boolean checkEmpty(JTextField field, StringBuilder sb, String msg){

boolean ok = true;

if (field.getText().equals ("")){

sb.append(msg).append("\n");

field.setBackground(Color.red);

ok = true;

}else{

field.setBackground(Color.white);

}

return ok;

}

public static boolean checkEmail(JTextField field, StringBuilder sb){

boolean ok = true;

if(!checkEmpty(field, sb,"Email chưa nhập ")){

return false;

}

Pattern pattern = Pattern.compile("\\w+@\\w+\\.\\w+");

Matcher matcher = pattern.matcher(field.getText());

if(!matcher.find()){

sb.append("Email không hợp lệ\n");

field.setBackground(Color.red);

ok = false;

}

if(ok){

field.setBackground(Color.white);

}

return ok;

}

}

1. **Ximage**

package uitil;

import java.awt.Image;

import java.io.File;

import java.net.URL;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.StandardCopyOption;

import javax.swing.ImageIcon;

/\*\*

\*

\* @author PC

\*/

public class XImage {

public static Image getAppIcon(){

URL url = XImage.class.getResource("/img/Left.png");

return new ImageIcon(url).getImage();

}

public static boolean save(File src){

File dst = new File("src\\main\\resources\\logo", src.getName());

if(!dst.getParentFile().exists()){

dst.getParentFile().mkdirs();//tạo thư mục

}

try {

Path from = Paths.get(src.getAbsolutePath());

Path to = Paths.get(dst.getAbsolutePath());

Files.copy(from, to, StandardCopyOption.REPLACE\_EXISTING);

return true;

} catch (Exception e) {

return false;

}

}

public static ImageIcon read(String fileName){

File path = new File("src\\main\\resources\\logo",fileName);

return new ImageIcon(path.getAbsolutePath());

}

}

1. **jdbcHelper**

package uitil;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

/\*\*

\*

\* @author PC

\*/

public class jdbcHelper {

private static String driver ="com.microsoft.sqlserver.jdbc.SQLServerDriver";

private static String dburl = "jdbc:sqlserver://localhost;database=QL\_Nhahang";

private static String user ="sa";

private static String pass ="123";

/\*

\*Nạp diver

\*/

static {

try {

Class.forName(driver);

} catch (ClassNotFoundException ex) {

throw new RuntimeException(ex);

}

}

public static PreparedStatement getStmt(String sql, Object... args) throws SQLException{

Connection connection = DriverManager.getConnection(dburl, user, pass);

PreparedStatement pstmt = null;

if(sql.trim().startsWith("{")){

pstmt = connection.prepareCall(sql);

}else{

pstmt=connection.prepareStatement(sql);

}

for(int i = 0; i < args.length; i++){

pstmt.setObject(i + 1, args[i]);

}

return pstmt;

}

public static int update(String sql, Object... args){

try {

PreparedStatement stmt = getStmt(sql, args);

try {

return stmt.executeUpdate();

}finally {

stmt.getConnection().close();

}

} catch (Exception e) {

throw new RuntimeException(e);

}

}

public static ResultSet query(String sql, Object... args){

try {

PreparedStatement stmt= getStmt(sql, args);

return stmt.executeQuery();

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

public static Object value(String sql,Object... args){

try {

ResultSet rs = query(sql, args);

if(rs.next()){

return rs.getObject(0);

}

rs.getStatement().getConnection().close();

} catch (Exception e) {

throw new RuntimeException(e);

}

return null;

}

}

**VIII.KIỂM THỬ**

**1.ĐĂNG NHÂP**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Đăng nhập để trống tên và mật khẩu |  | Yêu cầu nhập | ok |
| 2 | Đăng nhập sai tên | abc, 123 | Sai tên đăng nhập | ok |
| 3 | Đăng nhập sai mật khẩu | PV01, 12333 | Sai mật khẩu | ok |
| 4 | Đăng nhập đúng | PV01,123 | Mở form main | ok |

**2.ĐỔI MẬT KHẨU**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Sai mật khẩu cũ | ABC | Thông báo lỗi | ok |
| 2 | Sai xác nhận mật khẩu | 24363 | Thông báo lỗi | ok |
| 3 | Bấm thoát | btnThaot | Uot trương trình | ok |
| 4 | Sai mã NV | abc | Thông báo lỗi |  |
| 5 | Đổi thành công | PV01,123,456,456 | Đổi thành công |  |

**3.Form NhanVien**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Để trống ô nhập | Hiển thị thông báo và hiện ô nhập thành màu xanh | OK |  |
| 2 | Sửa thông tin | Họ Tên : Ngô Bình | OK |  |
| 3 | Nhập mã nhân viên đã có | Thông báo | Not |  |
| 4 | Xóa nhân viên | MaNV: PV06 | OK |  |
| 5 | Đổ dữ liệu từ bảng lên form | Double click | OK |  |

**4.Form Ban**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Để trống ô nhập | Hiển thị thông báo và hiện ô nhập thành màu xanh | OK |  |
| 2 | Thêm bàn mới | MaBan: 01,Tên 1 | OK |  |
| 3 | Sửa bàn | MaBan 01, Tên : Ban Mới | OK |  |
| 4 | Xóa bàn | MaBan 01 | OK |  |
| 5 | Đổ dữ liệu từ bảng lên form | click | OK |  |

**5.Form menu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Để trống ô nhập | Hiển thị thông báo và hiện ô nhập thành màu xanh | OK |  |
| 2 | Thêm mới món |  | OK |  |
| 3 | Nhập mã nhân viên đã có | Thông báo | Not |  |
| 4 | Sửa món | Tên : Bánh mì | OK |  |
| 5 | Xóa món | Mã món : K02 | OK |  |
| 6 | Đổ dữ liệu từ bảng lên form | Double click | OK |  |

**6.Form Order**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TH** | **Mô tả tình huống** | **Dữ liệu mẫu** | **Kết quả trông đợi** | **Fix** |
| 1 | Để trống ô nhập | Hiển thị thông báo và hiện ô nhập thành màu xanh | OK |  |
| 2 | Thêm mới món |  | OK |  |
| 3 | Nhập mã nhân viên đã có | Thông báo | Not |  |
| 4 | Sửa món | Tên : Bánh mì | OK |  |
| 5 | Xóa món | Mã món : K02 | OK |  |
| 6 | Đổ dữ liệu từ bảng lên form | Double click | OK |  |