

CAO ĐẲNG THỰC HÀNH FPT POLYTECHNIC
ĐÀ NẴNG



FPT POLYTECHNIC
Assignment



Đề tài:

Xây dựng và thiết kế phần mềm quản lý Quán Ăn

GIẢNG VIÊN HƯỚNG DẪN: Nguyễn Văn Định

SINH VIÊN THỰC HIỆN: Lê Văn Hưng

MÃ SỐ SINH VIÊN: PD05050

Huế ngày 3 tháng 11 năm 2021

Mục Lục

I.	Phân tích	4
1.	Hiện trạng	4
2.	Yêu cầu hệ thống	4
3.	Use case	5
a.	Usecase hệ thống Quản Lý Quán Ăn	5
b.	Usecase Quản lý nhân viên	Error! Bookmark not defined.
c.	Usecase Quản lý Món Ăn	Error! Bookmark not defined.
d.	Usecase Quản lý bán hàng	Error! Bookmark not defined.
e.	Usecase Thống kê	Error! Bookmark not defined.
f.	Đặc tả Usecase	Error! Bookmark not defined.
II.	Thiết kế	10
1.	Mô hình triển khai	10
2.	Thiết kế CSDL	10
a.	Sơ đồ quan hệ thực thể	10
b.	Thiết kế chi tiết các thực thể	11
3.	Thiết kế giao diện	12
a.	Sơ đồ tổ chức giao diện	12
b.	Thiết kế giao diện	13
III.	Thực hiện viết mã	20
1.	Viết mã tạo CSDL	20
a.	Tạo CSDL	20
b.	SQL truy vấn và thao tác	21
c.	Các thủ tục lưu tổng hợp thống kê	22
2.	Lập trình JDBC	24
a.	Lớp hỗ trợ	24
b.	Model class-Các lớp mô tả dữ liệu	26
c.	DAO class-Các lớp truy xuất dữ liệu	30
3.	Viết mã cho ứng dụng	33
a.	Xử lý Form	34
b.	Xử lý Form	Error! Bookmark not defined.
IV.	Kiểm thử	45
1.	Kiểm thử form	45

2. Kiểm thử form	47
V. Đóng gói và triển khai.....	49
1. Hướng dẫn chuyển đổi jar thành exe	49
2. Hướng dẫn cài đặt triển khai	70
3. Hướng dẫn sử dụng phần mềm.....	71

I. Phân tích

1. Hiện trạng

Trong thời buổi công nghệ hiện nay, các hình thức quản lý thủ công trước kia bắt đầu giảm bớt do những sự cố sai sót đáng tiếc ảnh hưởng đến người tiêu dùng khách hàng lẫn các doanh nghiệp thương mại. Thay vào đó, những phần mềm công nghệ ngày càng xuất hiện đa dạng, hỗ trợ hầu hết các lĩnh vực. Những phần mềm đem lại những lợi ích vô cùng to lớn, hạn chế được những sai sót đáng tiếc khi dùng phương pháp thủ công trước kia.

Nhận thấy được nhu cầu trên cùng với qua tìm hiểu, em xin xây dựng hệ thống phần mềm “Quản lý quán ăn”.

a. Hệ Thống Cũ

1. Ưu điểm

- Quản lý các món ăn và tài liệu về các món ăn rõ ràng

2. Nhược điểm

- Bảo mật kém
- Mọi thống kê phải viết bằng tay không tự động
- Cần rất nhiều nhân viên để làm nhiều công việc khác nhau

b. Hệ Thống Mới

1. Ưu điểm

- Bảo mật tốt
- Tìm kiếm và thống kê nhanh chóng
- Không cần nhiều nhân viên
- Bán hàng một cách nhanh chóng

2. Nhược điểm

- Các nhân viên mới cần kiến thức cơ bản về việc sử dụng máy tính

2. Yêu cầu hệ thống

a. Yêu cầu hệ thống

1. Yêu cầu về phần cứng

- Ram: 4gb
- CPU: i3 trở lên

2. Yêu cầu về công nghệ

- JDK 1.8
- Ngôn ngữ lập trình: Java
- Phần mềm lập trình: Netbean và SQL server

b. Yêu cầu của người dùng

1. Yêu cầu lưu trữ

- Thông tin nhân viên
- Thông tin món ăn
- Thông tin bán hàng

2. Yêu cầu quản lý

- Quản lý nhân viên
- Quản lý món ăn
- Quản lý bán hàng

3. Yêu cầu thống kê

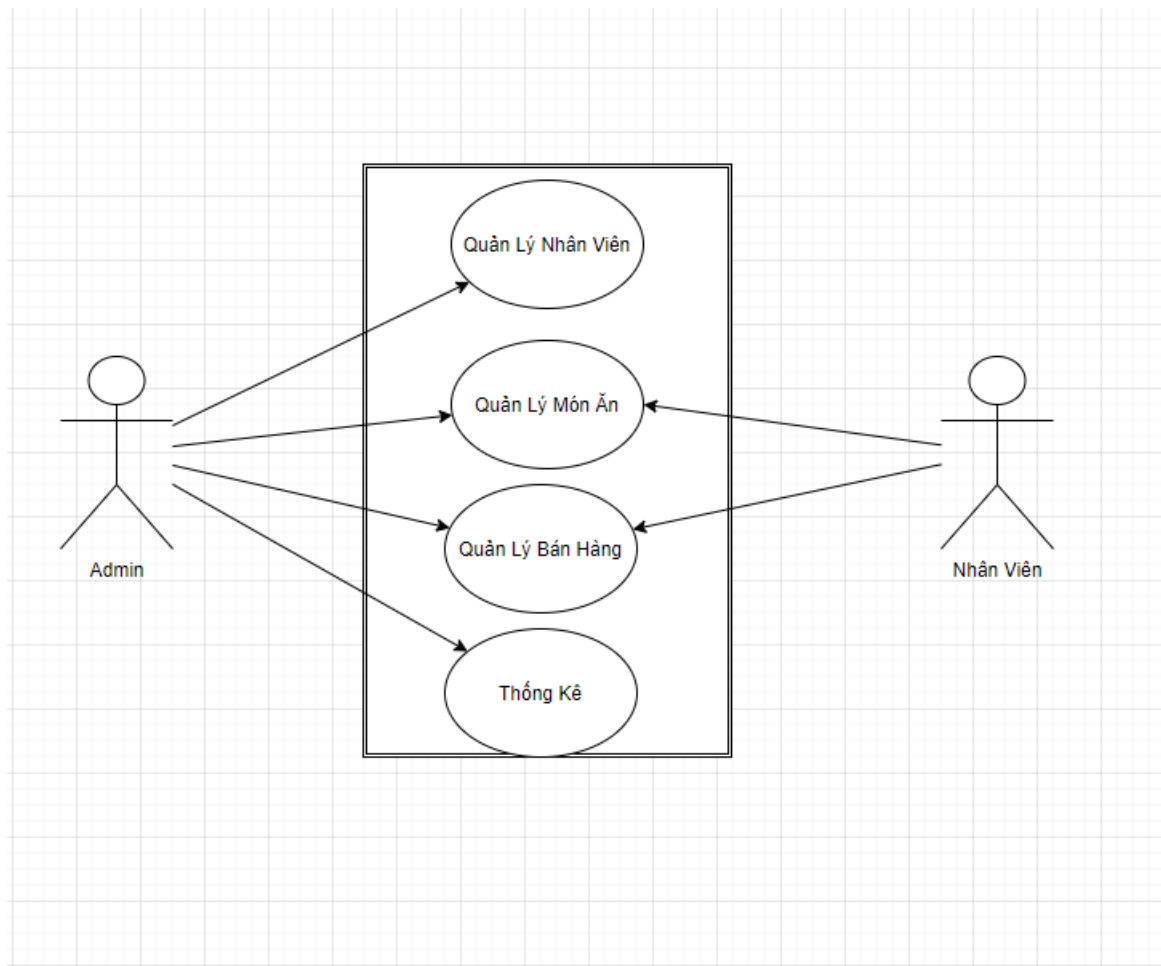
- Thống kê doanh thu tuần, tháng, năm

4. Yêu cầu bảo mật

- Tất cả thành viên phải đăng nhập mới được sử dụng phần mềm
- Admin mới có thể xem được thống kê Doanh thu
- Nhân viên không được phép thêm sửa xóa thông tin của nhân viên khác

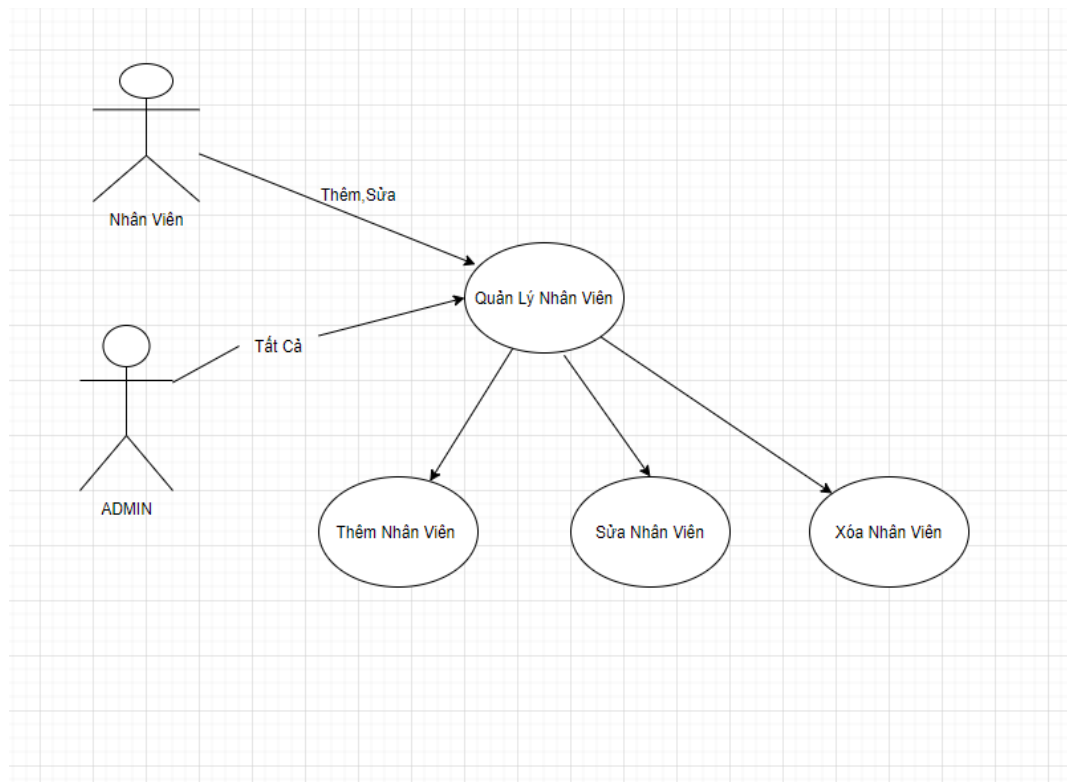
3. Use case

a. Usecase tổng quát hệ thống quản lý quán ăn



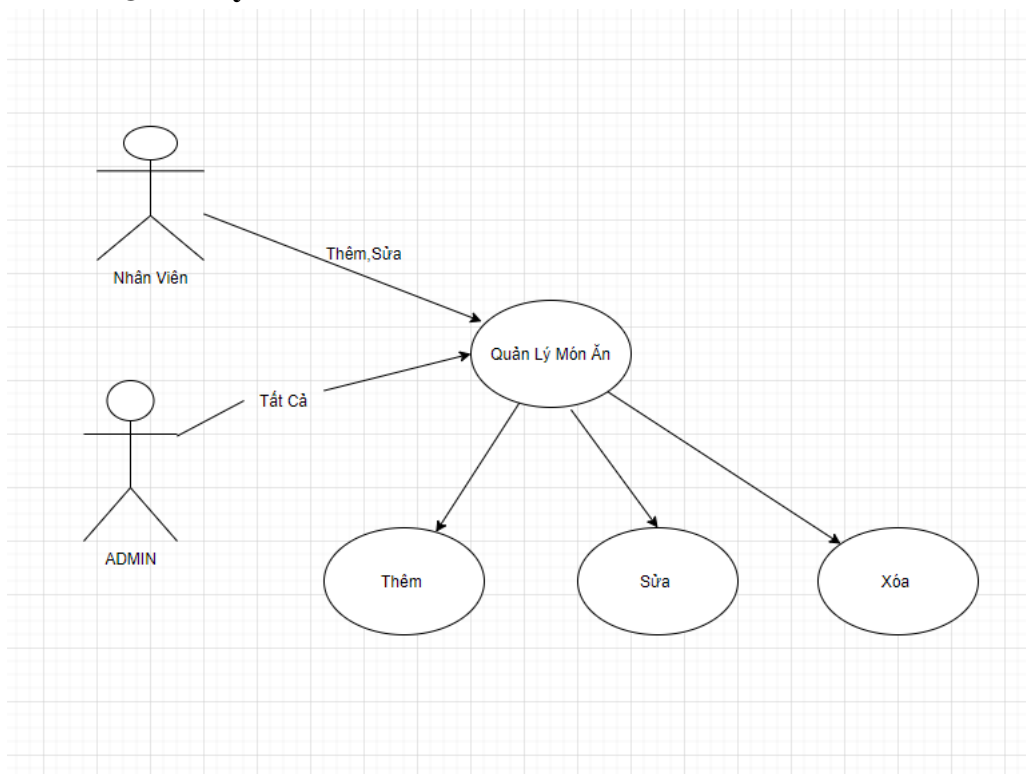
Hình 1

b. Usecase Quản Lý Nhân Viên



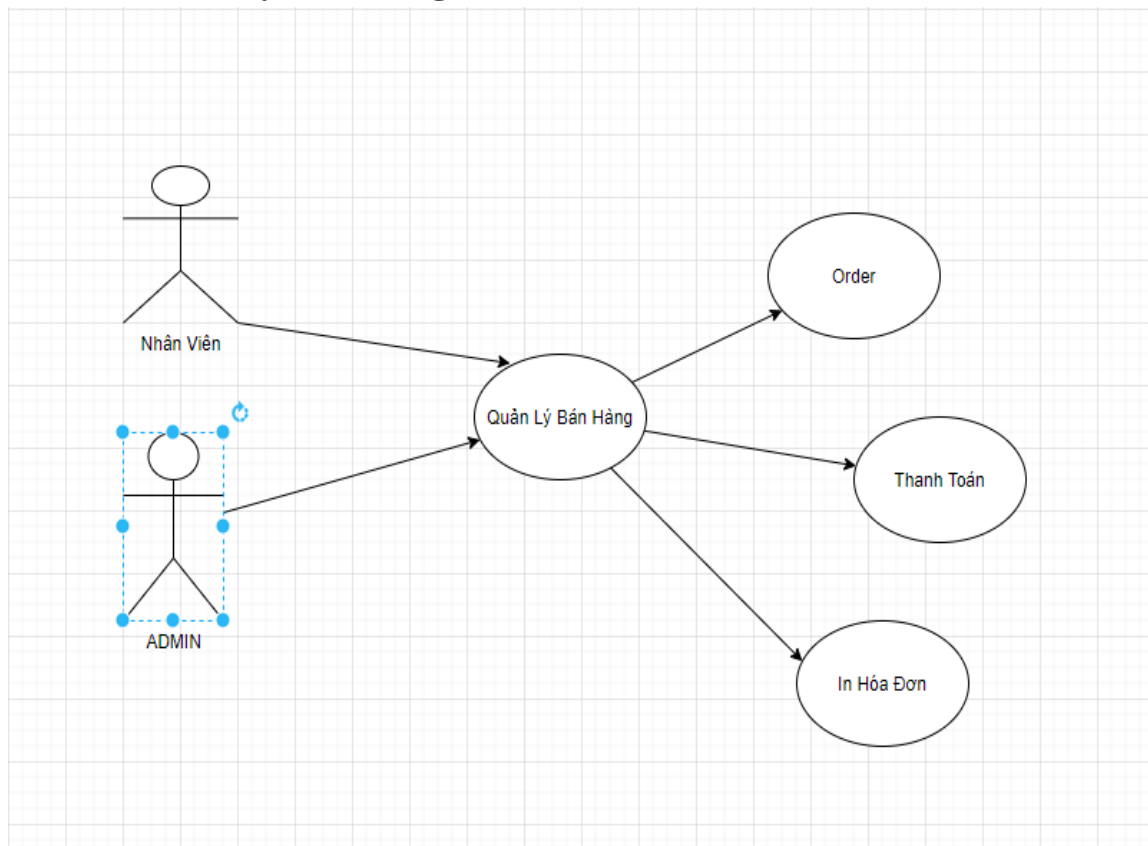
Hình 2

c. Usecase Quản Lý Món Ăn



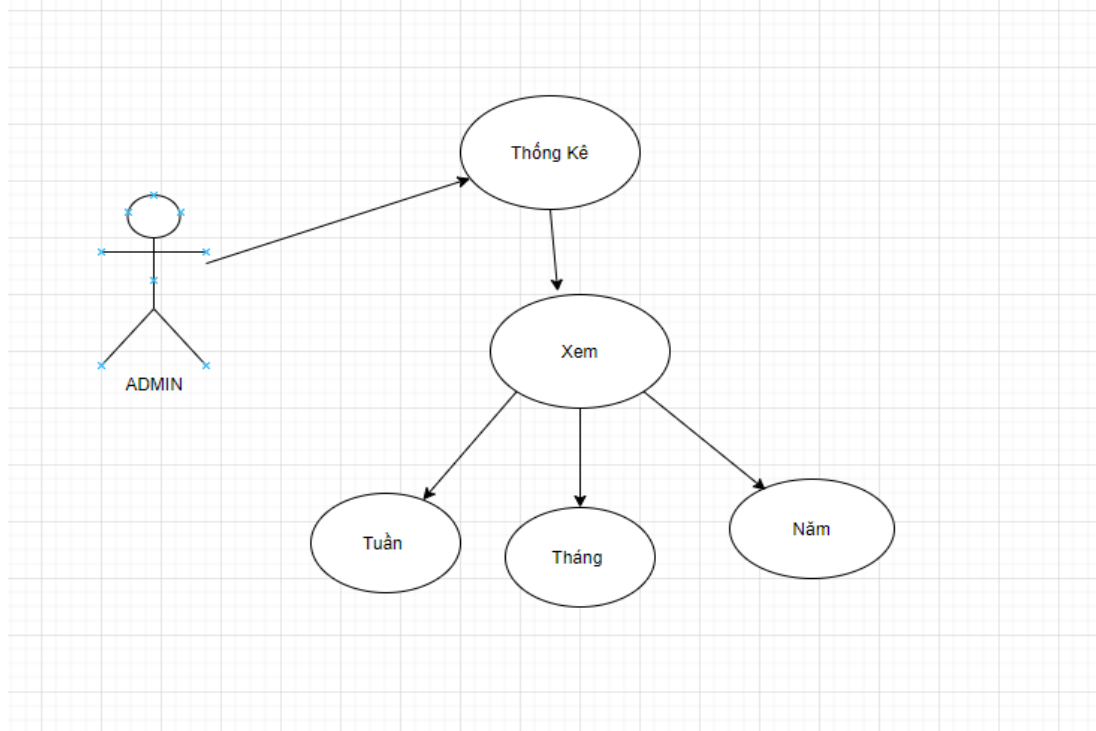
Hình 3

d. Usecase Quản Lý Bán Hàng



Hình 4

e. Usecase Thống Kê



Hình 5

CHI TIẾT CÁC CHỨC NĂNG

1. Các Usecase bảng Quản Lý(Nhân viên, Món Ăn,Bán Hàng)

- Các chức năng của các bảng Admin
 - Thêm: thêm các dữ liệu ghi trên bảng vào CSDL
 - Sửa: sửa các dữ liệu đang xem trên bảng
 - Xóa: xóa các dữ liệu đang xem trên bảng
 - Tìm kiếm: tìm kiếm các dữ liệu trong CSDL
 - Điều hướng: di chuyển trước và sau
 - Các chức năng(Thêm,sửa,xóa) cần được kiểm lỗi hợp lý với dữ liệu
 - Bán Hàng có chức năng order,Thanh Toán,In Hóa Đơn.

2. Các Usecase Thống kê(Doanh thu,)

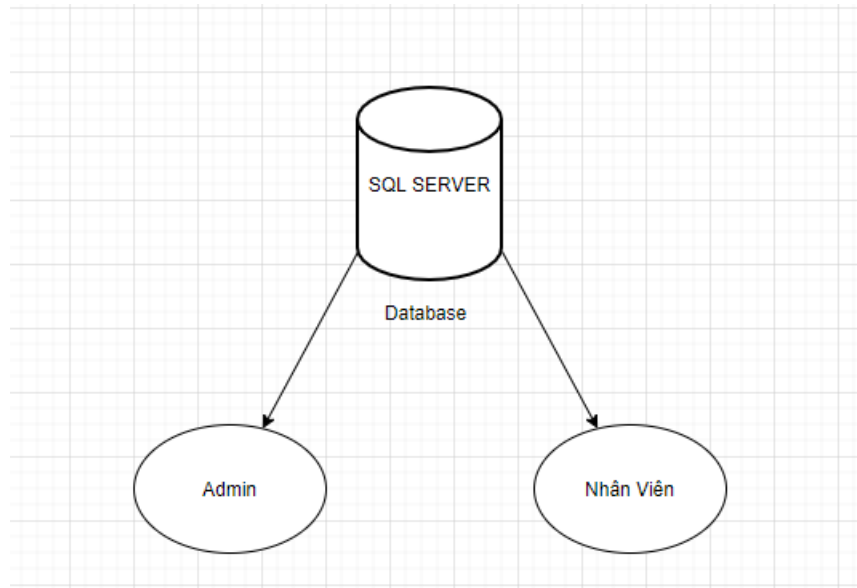
- Thống kê Doanh thu(chỉ có tài khoản Admin được xem)

3. Chi tiết về yêu cầu bảo mật

- Tất cả chức năng phải đăng nhập mới được sử dụng
- Admin: được phép sử dụng tất cả chức năng
- Nhân viên:
 - Không được xem Thống kê doanh thu
 - Không được Thêm sửa xóa thông tin của các nhân viên khác

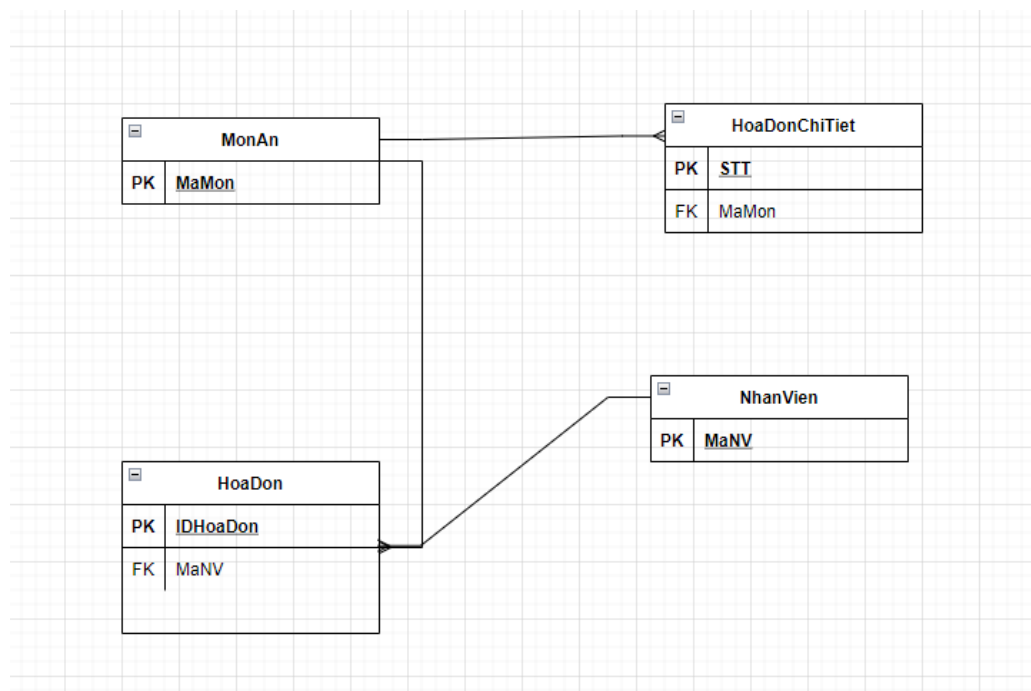
II. Thiết kế

1. Mô hình triển khai

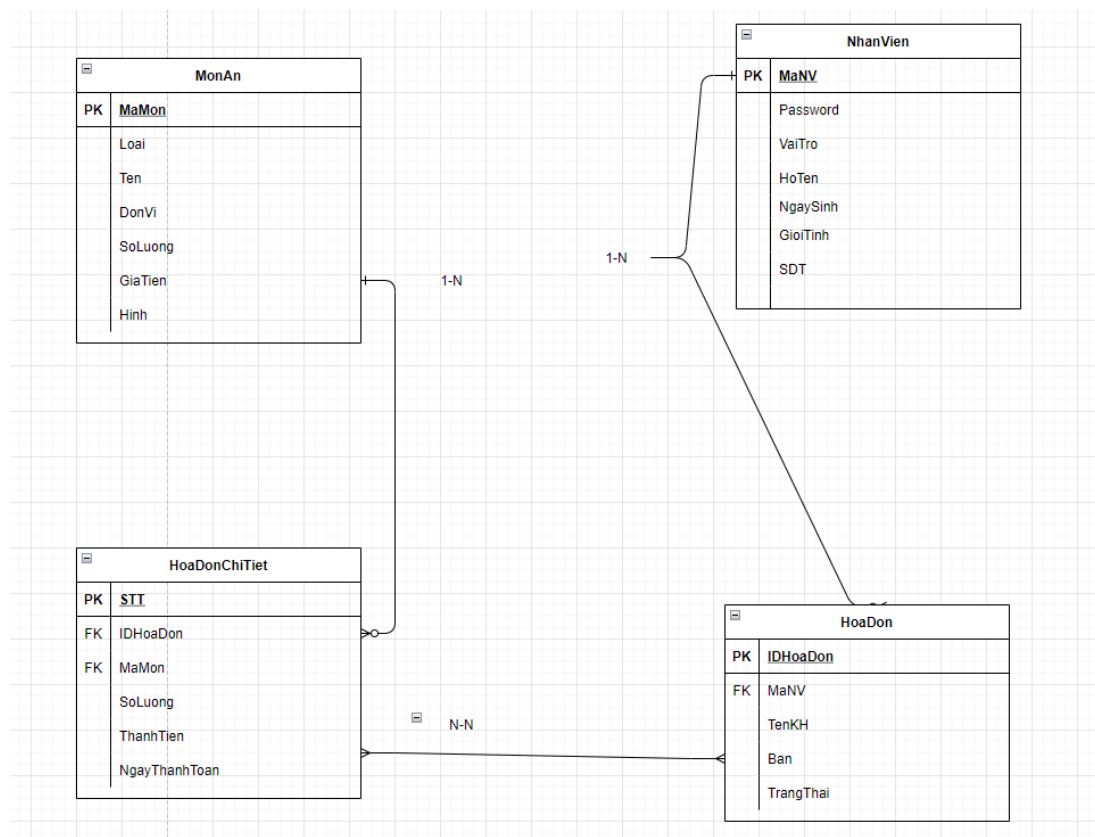


2. Thiết kế CSDL

a. Sơ đồ quan hệ thực thể(ERD)(cấp 1)



b. Sơ đồ quan hệ thực thể(ERD)(cấp 2)



c. Thiết kế chi tiết các thực thể

○ Nhân Viên:

STT	Tên Trường	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
1	MaNV	Nchar(10)	Not null	Mã Nhân Viên
2	Password	Nvarchar(30)	Not null	Mật Khẩu
3	VaiTro	Bit	Default 0	Vai Trò
4	HoTen	Nvarchar(30)	Not Null	Họ và Tên
5	NgaySinh	Date	Not Null	Ngày sinh
6	GioiTinh	Bit	Default 0	Giới tính
7	SDT	Nvarchar(10)	Not Null	Số điện thoại

○ Món Ăn

STT	Tên Trường	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
1	MaMon	Nchar(10)	Not Null	Mã Món Ăn
2	Loai	Bit	Default 0	Loại
3	TenMon	Nvarchar(50)	Not Null	Tên Món Ăn
4	DonVi	Nvarchar(20)	Null	Đơn Vị
5	SoLuong	Int	Not Null	Số Lượng
6	GiaTien	Float	Not Null	Gía Tiền
7	Hinh	Nvarchar(50)	null	Hình

○ Hóa Đơn

STT	Tên Trường	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
1	IDHoaDon	int	Not null	Mã Hóa Đơn
2	MaNV	Nchar(10)	Not null	Mã nhân viên
3	TenKH	Nvarchar(30)	Not null	Tên Khách Hàng
4	Ban	Nvarchar(15)	Null	Id bàn
5	TrangThai	Nvarchar(30)	Not Null	Trạng Thái

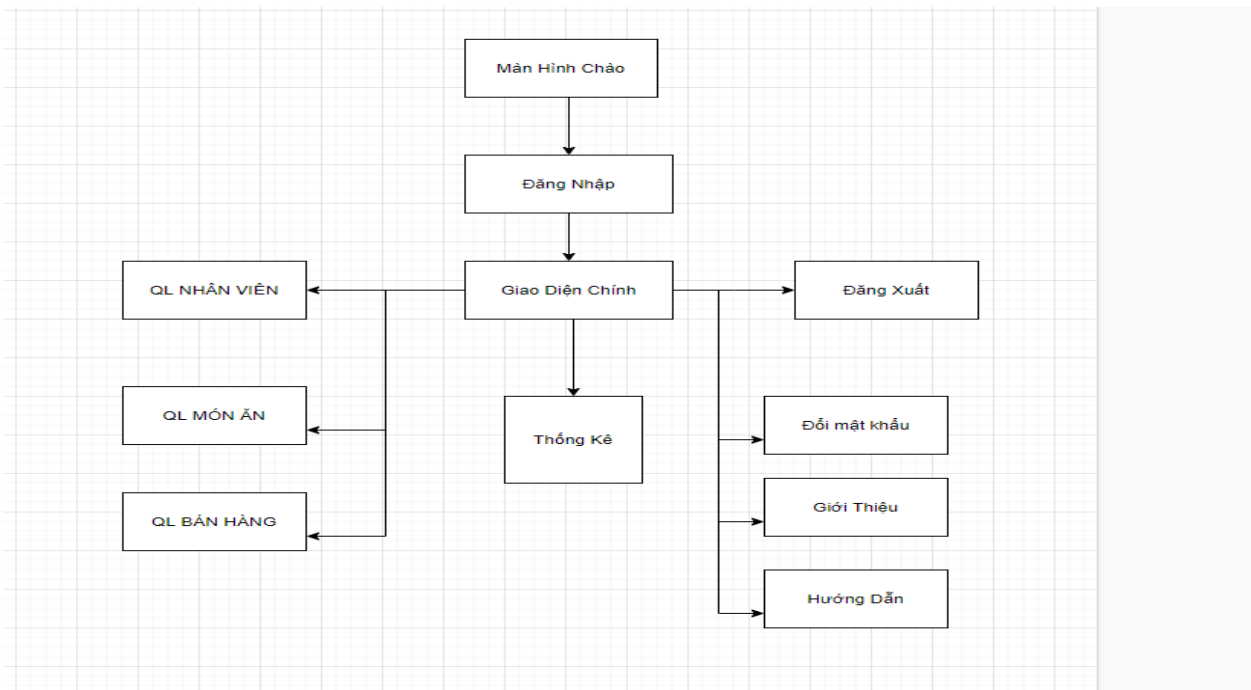
○ Hóa Đơn Chi Tiết

STT	Tên Trường	Kiểu Dữ Liệu	Ràng Buộc	Mô Tả
1	IDHoaDon	int	Not null	Id hóa đơn
2	MaMon	Nchar(10)	Not Null	Mã Món
3	SoLuong	Int	Not Null	Số Lượng
4	ThanhTien	Float	Not Null	Tổng Tiền
5	NgayThanhToan	Date	DEFAULT GET DATE	Ngày Thanh Toán

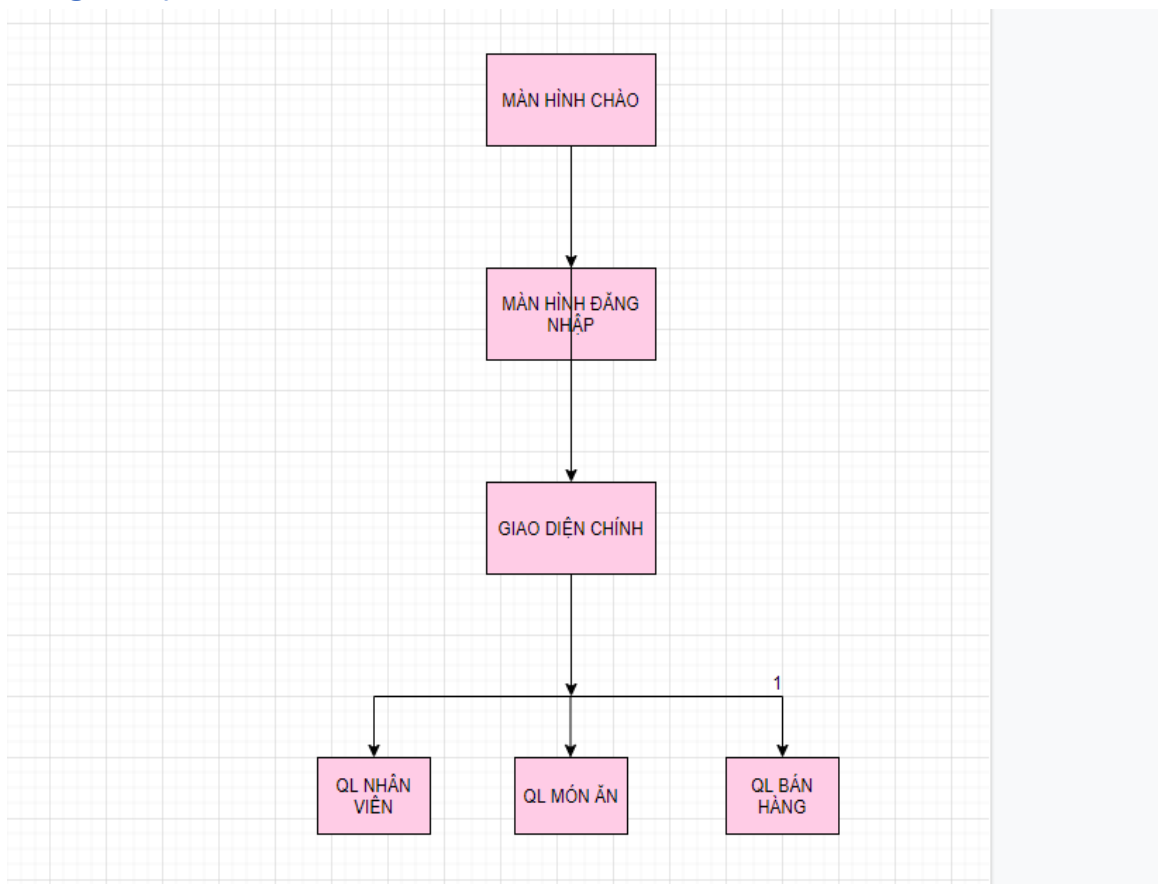
3. Thiết kế giao diện

a. Sơ đồ tổ chức giao diện

1.Sơ đồ tổ chức giao diện Quản Lý



2.Sơ đồ giao diện nhân viên




b. Thiết kế giao diện cho các chức năng nghiệp vụ


2.3.1:Thiết kế form màn hình chào




2.3.2:Thiết kế form đăng nhập

2.3.3: Thiết kế form đổi mật khẩu



Đổi Mật Khẩu


ĐỔI MẬT KHẨU


Tên đăng nhập




Mật khẩu hiện tại




Mật khẩu mới




Xác nhận mật khẩu mới








☐ Hiện Mật Khẩu?



Đồng ý


Hủy bỏ


2.3.4: Thiết kế form trang chính



HỆ THỐNG QUẢN LÝ QUẢN ĂN




Quản lý

Thống kê

Trợ giúp



Quản lý Quản Ăn


Xin Chào:



Admin



Lê Văn Hùng



Nhân Viên



Món Ăn



Bàn Hàng


Thống kê


Đăng Xuất




Thông Tin


10:39:32 PM





Hệ thống	Quản lý	Thống kê
	Đăng nhập	Ctrl+L
	Đăng xuất	Ctrl+O
	Đổi mật khẩu	
	Kết thúc	Ctrl+Q

Quản lý	Thống kê	Trợ giúp
	Nhân Viên	Ctrl+1
	Món Ăn	Ctrl+2
	Bán Hàng	Ctrl+3
Thống kê	Trợ giúp	
	Doanh Thu	Alt+1

Trợ giúp		
	Hướng dẫn sử dụng	F1
	Giới thiệu sản phẩm	F2

2.3.5: Thiết kế form quản lý nhân viên


QUẢN LÝ NHÂN VIÊN


QUẢN LÝ NHÂN VIÊN QUÁN ĂN

Tìm Kiếm:

CẬP NHẬT

Mã Nhân Viên:

Mật Khẩu:

Vai Trò:
☐ Admin
☐ Nhân Viên

Họ Và Tên:

Ngày Sinh:

Giới Tính

Số Điện Thoại:

Mã Nhân Viên	Mật Khẩu	Vai Trò	Họ Và Tên	Ngày Sinh	Giới Tính	Số Điện Th...
admin	admin	Admin	Lê Văn Hưng	14/02/2002	Nam	096264723
nv12	123123	Nhân Viên	le van hung	12/02/2002	Nam	0362772321

2.3.6: Thiết kế form quản lý món ăn

18

QUẢN LÝ BÁN HÀNG

QUẢN LÝ BÁN HÀNG

Hóa Đơn

Table

ID Hóa Đơn:

Mã NV:

admin

Tên Khách Hàng

Món:

Bò Xiên Nướng

Bàn:

Giá Tiền:

100000

Số Lượng:

Thành Tiền:

Ngày Thanh Toán:

30/11/2021

Trạng Thái:

In Hóa Đơn

Order

Thanh Toán

In Hóa Đơn

2.3.8: Thiết kế form thống kê

THỐNG KÊ				
THỐNG KÊ				
Năm: 2021				
Môn	NV Bán	Số Lượng	Title 4	Doanh Thu
BF1	admin	22	2021-11-30	2200000.0
BF1	admin	3	2021-11-30	300000.0
BF1	admin	2	2021-11-30	200000.0
BF1	admin	2	2021-11-30	200000.0
BF1	admin	3	2021-11-30	300000.0
BF1	admin	3	2021-11-30	300000.0
BF1	admin	3	2021-11-30	300000.0
BF1	admin	3	2021-11-30	600000.0
BF1	admin	5	2021-11-30	500000.0
BF1	admin	5	2021-11-30	500000.0
BF1	admin	3	2021-11-30	300000.0
BF1	admin	2	2021-11-30	200000.0
BF1	admin	2	2021-11-30	200000.0
BF3	admin	2	2021-11-30	400000.0

III. Thực hiện viết mã

1. Viết mã tạo CSDL

a. Tạo CSDL

```

CREATE DATABASE QuanLyQuanAn
GO
USE QuanLyQuanAn
GO
CREATE TABLE [dbo].[NhanVien](
    [MaNV] [nchar](10) NOT NULL,
    [Password] [nvarchar](30) NOT NULL,
    [VaiTro] [bit] NOT NULL default 0,
    [HoTen] [nvarchar](50) NOT NULL,
    [NgaySinh] [date] NOT NULL,
    [GioiTinh] [bit] NOT NULL default 0,
    [SDT] [nvarchar](10) NOT NULL,
    PRIMARY KEY(MaNV)
)
GO
CREATE TABLE [dbo].[MonAn](
    [MaMon] [nchar](10) NOT NULL,
    [Loai] [bit] NOT NULL,

```

```

[TenMon] [nvarchar](50) NOT NULL,
[DonVi] [nvarchar](20) NULL,
[SoLuong] [int] NOT NULL,
[GiaTien] [float] NOT NULL,
[Hinh] [nvarchar](50) NULL,
PRIMARY KEY(MaMon)
)
CREATE TABLE [dbo].[HoaDon](
[IDHoaDon] [nchar](10) NOT NULL,
[MaNV] [nchar](10) NOT NULL,
[TenKH] [nvarchar](30) NOT NULL,
[Ban] [nvarchar](15) NULL,
[TrangThai] [nvarchar](30) NOT NULL DEFAULT N'Chưa Thanh Toán',
PRIMARY KEY(IDHoaDon),
FOREIGN KEY (MaNV) REFERENCES NhanVien(MaNV)
)
CREATE TABLE [dbo].[HoaDonChiTiet](
[STT] [int] IDENTITY(1,1) NOT NULL,
[IDHoaDon] [nchar](10) NOT NULL,
[MaMon] [nchar](10) NOT NULL,
[SoLuong] [int] NOT NULL,
[ThanhTien] [float] NOT NULL,
[NgayThanhToan] [date] NOT NULL DEFAULT getdate(),
PRIMARY KEY(STT),
FOREIGN KEY (MaMon) REFERENCES MonAn(MaMon)
)

```

b. SQL truy vấn và thao tác

1. Nhân Viên

Hành Động	Câu Lệnh SQL
Thêm	<i>INSERT INTO NhanVien(MaNV,Password,VaiTro,HoTen,NgaySinh,GioiTinh,SDT)</i> <i>VALUES(?,?,?,?,?,?,?)</i>
Cập Nhật	<i>UPDATE NhanVien SET</i> <i>Password=?,VaiTro=?,HoTen=?,NgaySinh=?,GioiTinh=?,SDT=? WHERE MaNV=?</i>
Xóa	<i>DELETE FROM NhanVien Where MaNV=?</i>
Truy Vấn Tất Cả	<i>SELECT * FROM NhanVien</i>
Truy Vấn Theo Mã	<i>SELECT *FROM NhanVien where MaNV = ?</i>

2. Bán Hàng

Hành Động	Câu Lệnh SQL
-----------	--------------

Thêm	<code>Call sp_TrucSoLuong(?,?,?,?,?,?)</code>
Cập Nhật	<code>update HoaDon set TrangThai = ? where IDHoaDon = ?</code>
Truy Vấn Hóa Đơn Chưa Thanh Toán	<code>select hd.IDHoaDon , hd.MaNV , hd.TenKH ,ct.MaMon, hd.Ban , ct.SoLuong ,</code> <code>+ " ct.ThanhTien,ct.NgayThanhToan,hd.TrangThai from HoaDon as hd,HoaDonChiTiet as ct \n" +</code> <code>"where hd.IDHoaDon = ct.IDHoaDon and hd.TrangThai = N'Chưa Thanh Toán'</code>
Truy Vấn Mã Món theo id Hóa đơn	<code>select ma.TenMon from HoaDonChiTiet as ct,MonAn as ma where ct.MaMon = ma.MaMon and ct.idHoaDon = ?</code>
Truy vấn số lượng theo id hóa đơn	<code>select ct.SoLuong from HoaDonChiTiet as ct,MonAn as ma where ct.MaMon = ma.MaMon and ct.IDHoaDon = ?</code>
Truy vấn giá tiền theo id hóa đơn	<code>select ma.GiaTien from HoaDonChiTiet as ct,MonAn as ma where ct.MaMon = ma.MaMon and ct.idHoaDon = ?</code>
In Hóa đơn theo id hóa đơn	<code>select</code> <code>HoaDon.IDHoaDon,HoaDon.TenKH,HoaDonChiTiet.NgayThanhToan,Ban,TrangThai,sum(SoLuong)</code> <code>Soluong,sum(ThanhTien) from HoaDonChiTiet \n" +</code> <code>" Join HoaDon \n" +</code> <code>" on HoaDonChiTiet.IDHoaDon=HoaDon.IDHoaDon \n" +</code> <code>" where HoaDonChiTiet.IDHoaDon = ? \n" +</code> <code>" Group by HoaDon.IDHoaDon,NgayThanhToan,TenKH,Ban,TrangThai</code>

c. Các thủ tục lưu tổng hợp thống kê

1.Thống Kê

```
CREATE PROC [dbo].[sp_ThongKeDoanhThu](@Year INT)
AS BEGIN
SELECT
    COUNT(hd.STT) STT,
    MAX(hdct.MaMon) MaMon,
    MAX(hd.MaNV) MaNV ,
    SUM(DISTINCT hdct.SoLuong) SoLuong,
    MAX(hdct.NgayThanhToan) NgayThanhToan,
    SUM(hdct.ThanhTien) thanhtien
FROM HoaDonChiTiet hdct
    JOIN HoaDon hd ON hd.IDHoaDon=hdct.IDHoaDon
WHERE YEAR(NgayThanhToan) = @Year
GROUP BY hd.STT
END
GO
```

2.Thống Kê Theo Tháng

```
ALTER PROC [dbo].[sp_ThongKeThang](@Month INT)
AS BEGIN
SELECT
hdct.MaMon MaMon,
hd.MaNV MaNV,
SUM(DISTINCT hdct.SoLuong) SoLuong,
SUM(hdct.ThanhTien) ThanhTien
FROM HoaDonChiTiet hdct
JOIN HoaDon hd ON hd.IDHoaDon=hdct.IDHoaDon
WHERE MONTH(NgayThanhToan) = @Month
GROUP BY hdct.MaMon,hd.MaNV,hdct.NgayThanhToan
END
GO
```

3.Top 5 Món Bán Chạy

```
CREATE PROC [dbo].[sp_Top5]
AS BEGIN
SELECT TOP(5)
Max(hdct.MaMon) MaMon,
AVG(hdct.SoLuong)SoLuongTB,
SUM(hdct.SoLuong) Tong
FROM HoaDonChiTiet hdct
GROUP BY hdct.SoLuong
END
GO
```

4.Thêm Hóa Đơn Và Trừ Trực Tiếp Vào Số Lượng

```
CREATE proc [dbo].[sp_TrucSoLuong]
@id Nchar(10) , @manv nchar(10),@tenkh nvarchar(30) , @mamon nchar(10), @ban
Nvarchar(15),@soluong int,@thanhtien float
as
begin
    if
        exists (Select * From HoaDon where IDHoaDon = @id)
    begin
        insert into HoaDonChiTiet(IDHoaDon,MaMon,SoLuong,ThanhTien) values
        (@id,@mamon,@soluong,@thanhtien)
```

```

Update MonAn set SoLuong = SoLuong - @SoLuong Where MaMon = @MaMon
end
else
begin
insert into HoaDon(IDHoaDon, MaNV, TenKH, Ban) values (@id, @manv, @tenkh, @ban)
insert into HoaDonChiTiet(IDHoaDon, MaMon, SoLuong, ThanhTien) values
(@id, @mamon, @soluong, @thanhtien)
Update MonAn set SoLuong = SoLuong - @SoLuong Where MaMon = @MaMon
end
end
GO

```

2. Lập trình JDBC

a. Lớp hỗ trợ

2.1.1: Lớp Tiện Ích X-Date

```

public class XDate {
    static SimpleDateFormat formater = new SimpleDateFormat("dd/MM/yyyy");
    // chuyen doi string sang date
    // date là string cần chuyển
    //pattern là định dạng thời gian
    // return date kết quả

    public static Date now(){
        return new Date();
    }

    public static Date addDays(Date date, long days){
        date.setTime(date.getTime()+days*24*60*60*1000);
        return date;
    }

    public static Date add(int days){
        Date now = XDate.now();
        now.setTime(now.getTime()+ days*24*60*60*1000);
        return now;
    }

    public static String toString1(Date date, String...pattern){
        if(pattern.length > 0){
            formater.applyPattern(pattern[0]);
        }
        if(date == null){
            date = XDate.now();
        }
        return formater.format(date);
    }

    public static Date toDate(String date, String...pattern){

```



```

try{
    if(pattern.length > 0){
        formater.applyPattern(pattern[0]);
    }
    if(date == null){
        return XDate.now();
    }
    return formater.parse(date);
}
catch(ParseException ex){
    throw new RuntimeException(ex);
}
}

```

2.1.2: Lớp Tiện Ích DB

```

public class DB {
    private static String driver="com.microsoft.sqlserver.jdbc.SQLServerDriver";
    private static String dburl="jdbc:sqlserver://localhost;database=QuanLyQuanAn";
    private static String username="sa";
    private static String password="123456";
    static{
        try {
            Class.forName(driver);
        }
        catch (ClassNotFoundException ex) {
            throw new RuntimeException(ex);
        }
    }
    //xay dựng PreparedStatement

    public static PreparedStatement getStmt(String sql, Object...args) throws SQLException{
        Connection connection = DriverManager.getConnection(dburl, username, password);
        PreparedStatement pstmt = null;
        if(sql.trim().startsWith("{")){
            pstmt = connection.prepareCall(sql);// store procedure
        }
        else{
            pstmt = connection.prepareStatement(sql);//
        }
        for(int i=0;i<args.length;i++){
            pstmt.setObject(i + 1, args[i]);// ps.setString(1,hv.getHoten());
        }
        return pstmt;
    }
    public static ResultSet query(String sql, Object...args) {
        try {

```

```

        PreparedStatement stmt = DB.getStmt(sql, args);
        return stmt.executeQuery();
    }
    catch (SQLException e) {
        e.printStackTrace();
        throw new RuntimeException(e);
    }
}

public static Object value(String sql, Object...args) {
    try {
        ResultSet rs = DB.query(sql, args);
        if(rs.next()){
            return rs.getObject(0);
        }
        rs.getStatement().getConnection().close();
        return null;
    } catch (Exception e) {
        throw new RuntimeException(e);
    }
}

public static int update(String sql, Object...args){
    try{
        PreparedStatement stmt = DB.getStmt(sql, args);
        try{
            if(stmt.executeUpdate()>0){
                return 1;
            }
        }
        finally{
            stmt.getConnection().close();
        }
    }
    catch(SQLException e){
        throw new RuntimeException(e);
    }
    return 0;
}
}

```

b. Model class-Các lớp mô tả dữ liệu

2.2.3: Lớp Entity Nhân Viên

```

public class NhanVien {
    private String MaNV;
    private String Password;
    private boolean VaiTro;
    private String HoTen;
}

```

```
private Date NgaySinh;
private boolean GioiTinh;
private String SDT;
public NhanVien() {
}
public NhanVien(String MaNV, String Password, boolean VaiTro, String HoTen, Date NgaySinh,
boolean GioiTinh, String SDT) {
    this.MaNV = MaNV;
    this.Password = Password;
    this.VaiTro = VaiTro;
    this.HoTen = HoTen;
    this.NgaySinh = NgaySinh;
    this.GioiTinh = GioiTinh;
    this.SDT = SDT;
}
@Override
public String toString(){
    return this.getMaNV();
}
public String getMaNV() {
    return MaNV;
}
public void setMaNV(String MaNV) {
    this.MaNV = MaNV;
}
public String getPassword() {
    return Password;
}
public void setPassword(String Password) {
    this.Password = Password;
}
public boolean isVaiTro() {
    return VaiTro;
}

public void setVaiTro(boolean VaiTro) {
    this.VaiTro = VaiTro;
}
public String getHoTen() {
    return HoTen;
}
public void setHoTen(String HoTen) {
    this.HoTen = HoTen;
}
public Date getNgaySinh() {
    return NgaySinh;
}
public void setNgaySinh(Date NgaySinh) {
```

```
this.NgaySinh = NgaySinh;
}
public boolean getGioiTinh() {
    return GioiTinh;
}
public void setGioiTinh(boolean GioiTinh) {
    this.GioiTinh = GioiTinh;
}
public String getSDT() {
    return SDT;
}
public void setSDT(String SDT) {
    this.SDT = SDT;
}
```

2.1.4:Lớp Entity Hóa Đơn

```
public class HoaDon {
    private String IDHoaDon;
    private String MaNV;
    private String TenKH;
    private String MaMon;
    private String Ban;
    private int SoLuong;
    private Double ThanhTien;
    private Date NgayThanhToan;
    private String TrangThai;

    public HoaDon() {
    }

    public HoaDon(String IDHoaDon, String MaNV, String TenKH, String Ban, String TrangThai) {
        this.IDHoaDon = IDHoaDon;
        this.MaNV = MaNV;
        this.TenKH = TenKH;
        this.Ban = Ban;
        this.TrangThai = TrangThai;
    }

    public HoaDon(String IDHoaDon, String MaMon, int SoLuong, Double ThanhTien, Date
    NgayThanhToan) {
        this.IDHoaDon = IDHoaDon;
        this.MaMon = MaMon;
        this.SoLuong = SoLuong;
        this.ThanhTien = ThanhTien;
    }
}
```

```
        this.NgayThanhToan = NgayThanhToan;
    }

    public HoaDon(String IDHoaDon, String MaNV, String TenKH, String MaMon, String Ban, int
    SoLuong, Double ThanhTien, Date NgayThanhToan, String TrangThai) {
        this.IDHoaDon = IDHoaDon;
        this.MaNV = MaNV;
        this.TenKH = TenKH;
        this.MaMon = MaMon;
        this.Ban = Ban;
        this.SoLuong = SoLuong;
        this.ThanhTien = ThanhTien;
        this.NgayThanhToan = NgayThanhToan;
        this.TrangThai = TrangThai;
    }

    public String toString(){
        return IDHoaDon;
    }
    public String getIDHoaDon() {
        return IDHoaDon;
    }
    public void setIDHoaDon(String IDHoaDon) {
        this.IDHoaDon = IDHoaDon;
    }
    public String getMaNV() {
        return MaNV;
    }
    public void setMaNV(String MaNV) {
        this.MaNV = MaNV;
    }
    public String getTenKH() {
        return TenKH;
    }
    public void setTenKH(String TenKH) {
        this.TenKH = TenKH;
    }
    public String getMaMon() {
        return MaMon;
    }
    public void setMaMon(String MaMon) {
        this.MaMon = MaMon;
    }
    public String getBan() {
        return Ban;
    }
    public void setBan(String Ban) {
        this.Ban = Ban;
    }
}
```

```

    }
    public int getSoLuong() {
        return SoLuong;
    }
    public void setSoLuong(int SoLuong) {
        this.SoLuong = SoLuong;
    }
    public Double getThanhTien() {
        return ThanhTien;
    }
    public void setThanhTien(Double ThanhTien) {
        this.ThanhTien = ThanhTien;
    }
    public Date getNgayThanhToan() {
        return NgayThanhToan;
    }
    public void setNgayThanhToan(Date NgayThanhToan) {
        this.NgayThanhToan = NgayThanhToan;
    }
    public String getTrangThai() {
        return TrangThai;
    }
    public void setTrangThai(String TrangThai) {
        this.TrangThai = TrangThai;
    }
}

```

c. DAO class-Các lớp truy xuất dữ liệu

2.1.1: Lớp NhanVienDAO

```

public class NhanVienDAO extends SystemDAO<NhanVien,String> {
    String INSERT_SQL = "INSERT INTO
    NhanVien(MaNV>Password,VaiTro,HoTen,NgaySinh,GioiTinh,SDT)VALUES(?,?,?,?,?,?,?)";
    String UPDATE_SQL="UPDATE NhanVien SET
    Password=?,VaiTro=?,HoTen=?,NgaySinh=?,GioiTinh=?,SDT=? WHERE MaNV=?";
    String UPDATEPASS_SQL="UPDATE NhanVien SET Password=?,VaiTro=?,HoTen=? WHERE
    MaNV=?";
    String DELETE_SQL="DELETE FROM NhanVien Where MaNV=?";
    String SELECT_ALL_SQL = "SELECT * FROM NhanVien";
    String SELECT_BY_ID_SQL = "SELECT *FROM NhanVien where MaNV = ?";

    @Override
    public int insert(NhanVien entity) {
        return DB.update(INSERT_SQL, entity.getMaNV(),entity.getPassword(),entity.isVaiTro(),
        entity.getHoTen(),entity.getNgaySinh(),entity.getGioiTinh(),entity.getSDT());
    }
    @Override

```

```

public int update(NhanVien entity) {
    return DB.update(UPDATE_SQL, entity.getPassword(),entity.isVaiTro(),entity.getHoTen(),
        entity.getNgaySinh(),entity.getGioiTinh(),entity.getSDT(),entity.getMaNV());
}
public int updatePass(NhanVien entity){
    return DB.update(UPDATEPASS_SQL,
entity.getPassword(),entity.isVaiTro(),entity.getHoTen(),entity.getMaNV());
}
@Override
public void delete(String id) {
    DB.update(DELETE_SQL, id);
}
@Override
public NhanVien selectById(String id) {
    List<NhanVien> list = selectBySql(SELECT_BY_ID_SQL,id);
    if(list.isEmpty()){
        return null;
    }
    return list.get(0);
}

@Override
public List<NhanVien> selectAll() {
    return this.selectBySql(SELECT_ALL_SQL);
}

@Override
protected List<NhanVien> selectBySql(String sql, Object... args) {
    List<NhanVien>list = new ArrayList<NhanVien>();
    try{
        ResultSet rs = DB.query(sql, args);
        while(rs.next()){
            NhanVien model = new NhanVien();
            model.setMaNV(rs.getString("MaNV"));
            model.setPassword(rs.getString("Password"));
            model.setVaiTro(rs.getBoolean("VaiTro"));
            model.setHoTen(rs.getString("HoTen"));
            model.setNgaySinh(rs.getDate("NgaySinh"));
            model.setGioiTinh(rs.getBoolean("GioiTinh"));
            model.setSDT(rs.getString("SDT"));
            list.add(model);
        }
        rs.getStatement().getConnection().close();
        return list;
    }catch(Exception e){
        throw new RuntimeException(e);
    }
}
}

```

```

public List<NhanVien> selectByKeyword(String keyword){
    String sql="SELECT * FROM NhanVien WHERE HoTen LIKE ?";
    return this.selectBySql(sql, "%" + keyword + "%");
}
}

```

2.3.2: Lớp MonAnDAO

```

public class HoaDonDAO extends SystemDAO<HoaDon,String> {

    String SELECT_BY_ID_SQL = "SELECT
HoaDon.IDHoaDon,HoaDon.MaNV,HoaDon.TenKH,HoaDonChiTiet.MaMon,HoaDon.Ban,\n" +
"HoaDonChiTiet.SoLuong,HoaDonChiTiet.ThanhTien,HoaDonChiTiet.NgayThanhToan,HoaDon.Tra
ngThai \n" +
"FROM HoaDon join HoaDonChiTiet \n" +
"on HoaDon.IDHoaDon = HoaDonChiTiet.IDHoaDon and HoaDonChiTiet.IDHoaDon = ?";
    public int insertHoaDon(HoaDon entity) {
        String SQL = "{Call sp_TrucSoLuong(?,?,?,?,?,?)";
        return DB.update(SQL,
            entity.getIDHoaDon(),
            entity.getMaNV(),
            entity.getTenKH(),
            entity.getMaMon(),
            entity.getBan(),
            entity.getSoLuong(),
            entity.getThanhTien()
        );
    }
    @Override
    public int update(HoaDon entity) {
        String SQL = "update HoaDon set TrangThai = ? where IDHoaDon = ?";
        return DB.update(SQL,
            entity.getTrangThai(),
            entity.getIDHoaDon()
        );
    }
    @Override
    public HoaDon selectById(String id) {
        List<HoaDon> list = selectBySql(SELECT_BY_ID_SQL,id);
        if(list.isEmpty()){
            return null;
        }
        return list.get(0);
    }

    public String inhoadon(int hd){
        String ad = "";
        try {
            String sql = "select monan.TenMon from MonAn as monan,HoaDonChiTiet as hdct "

```



```

        + "where monan.MaMon = hdct.MaMon and monan.MaMon = ?";
        ResultSet rs = DB.query(sql, hd);
        while (rs.next()) {
            ad += rs.getString(1)+",";
        }
    } catch (Exception e) {
    }
    return ad;
}

public List<HoaDon> ChuaThanhToan() {
    String sql = "select hd.IDHoaDon , hd.MaNV , hd.TenKH ,ct.MaMon, hd.Ban , ct.SoLuong , "
        + " ct.ThanhTien,ct.NgayThanhToan,hd.TrangThai from HoaDon as hd,HoaDonChiTiet as
ct \n" +
"where hd.IDHoaDon = ct.IDHoaDon and hd.TrangThai = N'Chưa Thanh Toán'";
    List<HoaDon> hd = new ArrayList<>();
    ResultSet rs = DB.query(sql);
    try {
        while (rs.next()) {
            HoaDon h = new HoaDon();
            h.setIDHoaDon(rs.getString("IDHoaDon"));
            h.setMaNV(rs.getString("MaNV"));
            h.setTenKH(rs.getString("TenKH"));
            h.setMaMon(rs.getString("MaMon"));
            h.setBan(rs.getString("Ban"));
            h.setSoLuong(rs.getInt("SoLuong"));
            h.setThanhTien(rs.getDouble("ThanhTien"));
            h.setNgayThanhToan(rs.getDate("NgayThanhToan"));
            h.setTrangThai(rs.getString("TrangThai"));
            hd.add(h);
        }
    } catch (Exception e) {
        System.out.println(e.getMessage());
    }
    return hd;
}

```

3. Viết mã cho ứng dụng

a. Xử lý Form Nhân Viên

```

public class NhanVienJDialog extends javax.swing.JDialog {
    NhanVienDAO dao = new NhanVienDAO();
    int row = -1;

    /**
     * Creates new form NhanVienJDialog
     */
    public NhanVienJDialog(java.awt.Frame parent, boolean modal) {
        super(parent, modal);
        initComponents();
        fillTable();
        setLocationRelativeTo(null);
        setTitle("QUẢN LÝ NHÂN VIÊN");
        init();
    }

    void init(){
        setLocationRelativeTo(null);
        fillTable();
        this.row = -1;
        updateStatus();
    }

    void setForm(NhanVien model){
        txtMaNV.setText(model.getMaNV());
        txtPass.setText(model.getPassword());
        rdoAdmin.setSelected(model.isVaiTro());
        rdoNV.setSelected(!model.isVaiTro());
        txtHoTen.setText(model.getHoTen());
        txtNgaySinh.setText(XDate.toString1(model.getNgaySinh()));
        cboGioiTinh.setSelectedIndex(model.getGioiTinh()?0:1);
        txtSDT.setText(model.getSDT());
    }

    NhanVien getForm(){
        NhanVien nv = new NhanVien();
        nv.setMaNV(txtMaNV.getText());
        nv.setPassword(new String(txtPass.getPassword()));
        nv.setVaiTro(rdoAdmin.isSelected());
        nv.setHoTen(txtHoTen.getText());
        nv.setNgaySinh(XDate.toDate(txtNgaySinh.getText()));
        nv.setGioiTinh(cboGioiTinh.getSelectedIndex() == 0);
        nv.setSDT(txtSDT.getText());
        return nv;
    }
}

```

```

private void btnFirstActionPerformed(java.awt.event.ActionEvent evt) {
    first();
}

private void btnPrevActionPerformed(java.awt.event.ActionEvent evt) {
    prev();
}

private void btnNextActionPerformed(java.awt.event.ActionEvent evt) {
    next();
}

private void btnLastActionPerformed(java.awt.event.ActionEvent evt) {
    last();
}

private void tbNhanVienMousePressed(java.awt.event.MouseEvent evt) {
    if(evt.getClickCount() == 2){
        this.row = tbNhanVien.rowAtPoint(evt.getPoint());
        edit();
    }
}

public boolean check16tuoi(JTextField txt) {
    txt.setBackground(white);
    Date date = XDate.toDate(txt.getText());
    Calendar c1 = Calendar.getInstance();
    Calendar c2 = Calendar.getInstance();
    c1.setTime(date);
    c2.setTime(new Date());
    long a = (c2.getTime().getTime() - c1.getTime().getTime()) / (24 * 3600 * 1000);
    if (a >= 5844) {
        return true;
    } else {
        txt.setBackground(pink);
        MsgBox.alert(this, txt.getName() + " phải trên 16 tuổi.");
        return false;
    }
}

public boolean checkTrungMa(JTextField txt) {
    txt.setBackground(white);
    if (dao.selectById(txt.getText()) == null) {
        return true;
    } else {
        txt.setBackground(pink);
        MsgBox.alert(this, txt.getName() + " đã bị tồn tại.");
        return false;
    }
}
}

```

```

private void btnThemActionPerformed(java.awt.event.ActionEvent evt) {
    if (check.checkNotNullText(txtMaNV)
        && check.checkNotNullPass(txtPass)
        && check.checkNotNullText(txtHoTen)
        && check.checkNotNullText(txtNgaySinh)
        && check.checkNotNullText(txtSDT)){
        if (check.checkMaNV(txtMaNV)
            && check.checkPass(txtPass)
            && check.checkName(txtHoTen)
            && check.checkDate(txtNgaySinh)
            && check.checkSDT(txtSDT)) {
            if (checkTrungMa(txtMaNV)) {
                if (check16tuoi(txtNgaySinh)) {
                    insert();
                }
            }
        }
    }
}

private void btnSuaActionPerformed(java.awt.event.ActionEvent evt) {
    if (check.checkNotNullText(txtMaNV)
        && check.checkNotNullPass(txtPass)
        && check.checkNotNullText(txtHoTen)
        && check.checkNotNullText(txtNgaySinh)
        && check.checkNotNullText(txtSDT)){
        if (
            check.checkPass(txtPass)
            && check.checkName(txtHoTen)
            && check.checkDate(txtNgaySinh)
            && check.checkSDT(txtSDT)) {
            if (check16tuoi(txtNgaySinh)) {
                update();
            }
        }
    }
}

private void btnXoaActionPerformed(java.awt.event.ActionEvent evt) {
    delete();
}

private void btnResetActionPerformed(java.awt.event.ActionEvent evt) {
    clearForm();
}

private void btnTimActionPerformed(java.awt.event.ActionEvent evt) {
    this.fillTable();
    this.clearForm();
}

```

```

    }
    void fillTable(){
        DefaultTableModel model = (DefaultTableModel) tbNhanVien.getModel();
        model.setRowCount(0);
        try{
//            List<NhanVien>list = dao.selectAll();
            String keyword = txtTim.getText();
            List<NhanVien> list = dao.selectByKeyword(keyword);
            for(NhanVien nv:list){
                Object[]row = {nv.getMaNV(),nv.getPassword(),
                    nv.isVaiTro()?"Admin":"Nhân Viên",
                    nv.getHoTen(),
                    XDate.toString1(nv.getNgaySinh()),
                    nv.getGioiTinh()?"Nam":"Nữ",
                    nv.getSDT()
                };
                model.addRow(row);
            }
        }
        catch(Exception e){
            MsgBox.alert(this, e.getMessage());
        }
    }
    void insert(){
        NhanVien model = getForm();
        try{
            dao.insert(model);
            this.fillTable();
            this.clearForm();
            MsgBox.alert(this, "Thêm mới thành công");
        }catch (Exception e){
            MsgBox.alert(this, e.getMessage());
        }
    }

    void update(){
        NhanVien model = getForm();
        try{
            dao.update(model);
            this.fillTable();
            this.clearForm();
            MsgBox.alert(this, "Cập nhật thành công");
        }
        catch(Exception e){
            MsgBox.alert(this, e.getMessage());
        }
    }
}

```

```

void delete(){
    if(!Auth.isManager()){
        MsgBox.alert(this, "Chỉ admin mới được xóa nhân viên!");
    }
    else{
        String manv = txtMaNV.getText();
        if(manv.equals(Auth.user.getMaNV())){
            MsgBox.alert(this, "Bạn không được xóa chính mình!");
        }
        else if(MsgBox.confirm(this, "Bạn thực sự muốn xóa nhân viên này?")){
            try {
                dao.delete(manv);
                this.fillTable();
                this.clearForm();
                MsgBox.alert(this, "Xóa thành công!");
            }
            catch (Exception e) {
                MsgBox.alert(this, "Xóa thất bại!");
            }
        }
    }
}

void edit(){
    try{
        String maNV = (String)tbNhanVien.getValueAt(this.row, 0);
        NhanVien model = dao.selectById(maNV);
        if(model !=null){
            setForm(model);
            Tabs.setSelectedIndex(0);
        }

        }catch(Exception e){
            MsgBox.alert(this, e.getMessage());
        }
    }

void first(){
    row =0;
    edit();
}

void prev(){
    if(row>0){
        row--;
        edit();
    }
}

```

```

void next() {
    if(row < tbNhanVien.getRowCount() - 1){
        row++;
        edit();
    }
}
void last(){
    row = tbNhanVien.getRowCount() - 1;
    edit();
}
void updateStatus(){
    boolean edit = this.row >= 0;
    boolean first = this.row == 0;
    boolean last = this.row == tbNhanVien.getRowCount() - 1;
    btnThem.setEnabled(!edit);
    btnSua.setEnabled(!edit);
    btnXoa.setEnabled(!edit);

    btnFirst.setEnabled(!edit &&!first);
    btnPrev.setEnabled(!edit &&!first);
    btnNext.setEnabled(!edit &&!first);
    btnLast.setEnabled(!edit &&!first);
}
void clearForm(){
    txtMaNV.setText("");
    txtPass.setText("");
    rdoAdmin.setSelected(true);
    txtHoTen.setText("");
    txtNgaySinh.setText("");
    cboGioiTinh.setSelectedIndex(0);
    txtSDT.setText("");
}

```

b.Xử lý Form Bán Hàng

```

public class BanHangJDialog extends javax.swing.JDialog {

    HoaDonDAO hddao = new HoaDonDAO();
    DecimalFormat df = new DecimalFormat("#####");
    MonAnDAO madao = new MonAnDAO();
    int row = -1;

```

```

int index =0;
/**
 * Creates new form BanHangJDialog
 */
public BanHangJDialog(java.awt.Frame parent, boolean modal) {
    super(parent, modal);
    initComponents();
    setLocationRelativeTo(null);
    init();
    HienThi();
}

void init() {
    setTitle("QUẢN LÝ BÁN HÀNG");
    setLocationRelativeTo(null);
    setTotal();
    FillTable(hddao.ChuaThanhToan());
    fillcombobox();
}

void HienThi() {
    txtMaNV.setText(Auth.user.getMaNV());
    txtNgayThanhToan.setText(XDate.toString1(new Date(), "dd/MM/yyyy"));
}

//lấy thông tin từ đt HoaDon đưa lên form
void setForm(HoaDon model) {
    txtHoaDon.setText(String.valueOf(model.getIDHoaDon()));
    txtMaNV.setText(String.valueOf(Auth.user.getMaNV()));
    txtTenKH.setText(String.valueOf(model.getTenKH()));
    cboTenMon.setToolTipText(String.valueOf(model.getMaMon()));
    cboTenMon.setSelectedItem(madao.selectById(model.getMaMon()));
    txtBan.setText(String.valueOf(model.getBan()));
    txtSoLuong.setText(String.valueOf(model.getSoLuong()));
    txtThanhTien.setText(String.valueOf(model.getThanhTien()));
    //    txtNgayThanhToan.setText(XDate.toString1(model.getNgayThanhToan()));

    lblTrangThai.setText(String.valueOf(model.getTrangThai()));
}

//lấy thông tin trên form điền vào đt HoaDon
//return chuyenDe
HoaDon getForm() {
    HoaDon model = new HoaDon();
    model.setIDHoaDon(String.valueOf(txtHoaDon.getText()));
    model.setMaNV(Auth.user.getMaNV());
    model.setTenKH(String.valueOf(txtTenKH.getText()));
    MonAn monan = (MonAn) cboTenMon.getSelectedItem();

```



```

        model.setMaMon(monan.getMaMon());
        model.setBan(String.valueOf(txtBan.getText()));
        model.setSoLuong(Integer.valueOf(txtSoLuong.getText()));
        model.setThanhTien(Double.valueOf(txtThanhTien.getText()));
//        model.setNgayThanhToan(XDate.toDate(lblNgayThanhToan.getText()));
        return model;
    }
    void selectComboBox() {
//        madao.selectAll().forEach(monan -> {
//            if (monan.getTenMon().equalsIgnoreCase((String) cboTenMon.getSelectedItem())) {
//                txtGiaTien.setText(String.valueOf(monan.getGiaTien()));
//            }
//        });
        MonAn monan = (MonAn) cboTenMon.getSelectedItem();
        lblGiaTien.setText(df.format(monan.getGiaTien()));
    }
    void fillcombobox() {
        DefaultComboBoxModel model = (DefaultComboBoxModel) cboTenMon.getModel();
        model.removeAllElements();
        try {
            List<MonAn> list = madao.selectAll();
            if(!list.isEmpty()){
                for (MonAn ma : list) {
                    model.addElement(ma);
                }
            }
        } catch (Exception e) {
            MsgBox.alert(this, e.getMessage());
        }
    }
    void FillTable(List<HoaDon> list) {
        DefaultTableModel model = (DefaultTableModel) tbBanHang.getModel();
        model.setRowCount(0);
        try {
            for (HoaDon hd : list) {
                Object[] row = {
                    hd.getIDHoaDon(),
                    hd.getMaNV(),
                    hd.getTenKH(),
                    hd.getMaMon(),
                    hd.getBan(),
                    hd.getSoLuong(),
                    hd.getThanhTien(),
                    XDate.toString1(hd.getNgayThanhToan(), "dd/MM/yyyy"),
                    hd.getTrangThai();
                };
                model.addRow(row);
            }
        }
    }

```

```

    }
    } catch (Exception e) {
        MsgBox.alert(this, e.getMessage());
    }
}

void setTotal() {
    new Thread() {
        @Override
        public void run() {
            while (true) {
                try {
                    MonAn monan = (MonAn) cboTenMon.getSelectedItem();
                    if (!txtSoLuong.getText().isEmpty()) {
                        int count = Integer.parseInt(txtSoLuong.getText());

                        double total = (count * monan.getGiaTien());
                        txtThanhTien.setText(df.format(total));
                    } else {
                        txtThanhTien.setText("");
                    }
                    Thread.sleep(1000);
                } catch (InterruptedException ex) {
                    Logger.getLogger(BanHangJDialog.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        }
    }.start();
}

void insert(){
    HoaDon model = getForm();
    try{
        hddao.insertHoaDon(model);
        FillTable(hddao.ChuaThanhToan());
        clearForm();
        MsgBox.alert(this, "Order thành công");
    }catch (Exception e){
        MsgBox.alert(this, e.getMessage());
    }
}

HoaDon UpdateHoaDon(){
    HoaDon model = new HoaDon();
    model.setIDHoaDon(txtHoaDon.getText());
    model.setTrangThai("Đã Thanh Toán");
    return model;
}

void update() {
    HoaDon model = UpdateHoaDon();

```

```

        hddao.update(model);
        lblTrangThai.setText("Đã Thanh Toán");
        this.FillTable(hddao.ChuaThanhToan());
    }
    void edit(){
        try{
            String idhoaddon = (String)tbBanHang.getValueAt(this.index, 0);
            HoaDon model = hddao.selectById(idhoaddon);
            if(model !=null){
                setForm(model);
                Tabs.setSelectedIndex(0);
            }

        }catch(Exception e){
            JOptionPane.showMessageDialog(this, e.initCause(e));
        }
    }
    void clearForm(){
        txtHoaDon.setText("");
        txtTenKH.setText("");
        cboTenMon.setSelectedIndex(0);
        txtBan.setText("");
        txtSoLuong.setText("");
        txtThanhTien.setText("");
        lblTrangThai.setText("");
    }
    private void btnOrderActionPerformed(java.awt.event.ActionEvent evt) {
        // TODO add your handling code here:
        if (
            check.checkNotNullText(txtHoaDon)
            && check.checkNotNullText(txtTenKH)
            && check.checkNotNullText(txtBan)
            && check.checkNotNullText(txtSoLuong)){
            if (
                check.checkName(txtTenKH)
                && check.checksoluong(txtSoLuong)){
                insert();
            }
        }
    }

    void printHoaDon(){
        HoaDon hd = hddao.printHoaDon(txtHoaDon.getText());
        txtInHoaDon.append("\t\t QUÁN ĂN RICE FRESH \n\n "
            + "        \t\t HÓA ĐƠN BÁN HÀNG \n\n"
            +
            "=====
            =====\n\n"

```

```

+ "ĐC:222 Trần Duy Hưng,Cầu Giấy,Hà Nội \n\n"
+ "SDT: 0359623642 \n\n"
+ "Hotline: 18005555 \n\n"
+
"=====
===== \n\n"
+ "Mã Hóa Đơn:      \t\t\t" + txtHoaDon.getText()      + "\n\n"
+ "Mã Nhân Viên:    \t\t\t" + txtMaNV.getText()      + "\n\n"
+ "Tên Khách Hàng:  \t\t\t" + txtTenKH.getText()      + "\n\n"
+ "Món:             \t\t\t" + hddao.SanPham(txtHoaDon.getText()) + "\n\n"
+ "Bàn:             \t\t\t" + hddao.printHoaDon(txtHoaDon.getText()) + "\n\n"
+ "Thành Tiền:      \t\t\t" + hddao.GiaTien(txtHoaDon.getText()) + "\n\n"
+ "Số Lượng:        \t\t\t" + hddao.soluong(txtHoaDon.getText()) + "\n\n"
+ "Ngày Thanh Toán: \t\t\t" + hd.getNgayThanhToan()      + "\n\n"
+ "Trạng Thái:      \t\t\t" + "Đã Thanh Toán"          + "\n\n"
+
"=====
===== \n\n"
+ "Thành Tiền:      \t\t\t" + XDecimal.ad.format(hd.getThanhTien()) + "\n\n"
+
"=====
===== \n\n"
+ "\t\t\t CẢM ƠN QUÝ KHÁCH VÀ HẸN GẶP LẠI \n\n"
+ "Mọi thắc mắc xin liên hệ hotline: 18005555 "
);
}
private void btnThanhToanActionPerformed(java.awt.event.ActionEvent evt) {
    printHoaDon();
    update();
}
private void btnHoaDonActionPerformed(java.awt.event.ActionEvent evt) {
    try {
        txtInHoaDon.print();
    } catch (PrinterException ex) {
        Logger.getLogger(BanHangJDialog.class.getName()).log(Level.SEVERE, null, ex);
    }
}
private void cboTenMonActionPerformed(java.awt.event.ActionEvent evt) {
    selectComboBox();
}
private void tbBanHangMouseClicked(java.awt.event.MouseEvent evt) {
    if (evt.getClickCount() == 1) {
        this.index = tbBanHang.rowAtPoint(evt.getPoint());
        if (this.index >= 0) {
            this.edit();
            Tabs.setSelectedIndex(0);
        }
    }
}

```

IV. Kiểm thử

1. Kiểm thử Giao Diện

V.	#	Hoạt động	Kết quả mong đợi	OK or NA
A		Kiểm tra bố cục		OK

1	Thuộc tính trang		
1.1	Kiểm tra bố cục phù hợp với màn hình độ phân giải	<p>có thanh menu ngang , có thanh menu dọc,</p> <p>-Hệ Thống : đăng nhập, đăng xuất, đổi mật khẩu, kết thúc</p> <p>-Quản Lý: Nhân Viên, Món Ăn, Thanh Toán</p> <p>-Thống kê: tổng hợp thống kê</p>	ok
1.2	Kiểm tra chính tả cho tiêu đề và cửa sổ giao diện	Tiêu đề và cửa sổ giao diện không sai chính tả	ok
1.3	Kiểm tra tình trạng đăng nhập	Phải có tính nhất quán trong khâu đăng nhập	ok
1.4	Hình ảnh button là cụ thể	<p>Một hình ảnh liên kết tới giao diện chính</p> <p>Quản Lý Nhân Viên</p> <p>Quản Lý Món Ăn</p> <p>Quản Lý Thanh Toán</p> <p>Báo Cáo Thống Kê</p>	ok
1.5	Bố trí hình ảnh	Một hình ảnh là một ứng dụng, có tất cả 4 ứng dụng	ok
1.6	Kiểm tra phiên bản ứng dụng	Xuất hiện bên dưới của form và trong phần giới thiệu	ok
1.7	Kiểm tra phần giới thiệu ứng dụng	Phần giới thiệu ứng dụng có thể xuất hiện ở phần đầu.	ok
2	Kích thước của mặt hàng		
2.1	Kiểm tra kích thước của các mục: phong chữ, nút, điều khiển văn bản, điều khiển danh sách, khung, ...	Kích thước của: Phong chữ -> Cỡ 13, Phong chữ tiêu đề đậm -> Cỡ 18 Khung -> Phù hợp với kích thước màn hình và độ phân giải của máy tính Nút -> Cỡ 15 Hình ảnh -> Cỡ 15	ok
3	Căn chỉnh của mặt hàng		ok
3.1	Kiểm tra quản lý nhân viên, quản lý món ăn, quản lý bán hàng, Thống Kê ,Thông Tin, đồng hồ	<p>Button quản lý Nhân Viên -> bên trái</p> <p>Button quản lý Món Ăn > bên trái</p> <p>Button quản lý Bán Hàng -> bên trái</p> <p>Button Thống Kê-> bên trái</p> <p>Button Thông Tin -> bên trái</p>	ok

		Đồng hồ -> Bên Trái	
3.2	Kiểm tra sự liên kết của các trường bố cục	Tất cả các trường được phải được căn chỉnh với nhau	ok
4	Các trường chỉ đọc		ok
4.1	Kiểm tra màu sắc	Cùng màu với nền	ok
4.2	Kiểm tra bàn phím có thể điều hướng được	bàn phím có thể điều hướng được	ok
5	Định dạng tiêu chuẩn		ok
5.1	Kiểm tra các định dạng tiêu chuẩn: Ngày	dd/MM/yyyy	ok

1. Kiểm thử form Nhân Viên

Check Tính Hợp Lệ

Hành Động	Kết Quả Mong Đợi	Kết Quả Thực Tế	Ok hoặc Lỗi
Không nhập dữ liệu	Không được để trống	Không được để trống	OK
Nhập dữ liệu sai	<p>Thông báo và chuyển màu sang màu hồng :</p> <p>mã nhân viên phải có 4 -10 kí tự chữ thường ,chữ hoa hoặc số</p> <p>Mật Khẩu phải có từ 3 – 16 kí tự</p> <p>Tên phải là tên tiếng việt hoặc không dấu từ 3-25 kí tự</p> <p>Ngày Sinh phải đúng định dạng dd/yy/mm</p> <p>Số Điện Thoại phải gồm 10 số đúng các đầu số của nhà mạng</p>	<p>Thông báo và chuyển màu sang màu hồng :</p> <p>mã nhân viên phải có 4 -10 kí tự chữ thường ,chữ hoa hoặc số</p> <p>Mật Khẩu phải có từ 3 – 16 kí tự</p> <p>Tên phải là tên tiếng việt hoặc không dấu từ 3-25 kí tự</p> <p>Ngày Sinh phải đúng định dạng dd/yy/mm</p> <p>Số Điện Thoại phải gồm 10 số đúng các đầu số của nhà mạng</p>	OK

Kiểm Thử Form

Hành Động	Kết Quả Mong Đợi	Kết Quả Thực Tế	Ok hoặc Lỗi
Nhập dữ liệu đầy đủ và ấn nút Thêm	Hiện Thị thông báo thêm mới thành công và đổ dữ liệu lên bảng	Hiện Thị thông báo thêm mới thành công và đổ dữ liệu lên bảng	OK
Không nhập dữ liệu và ấn nút thêm	Thông báo không được để trống và chuyển màu ô sang màu hồng	Thông báo không được để trống và chuyển màu ô sang màu hồng	OK
Không nhập và ấn nút sửa	Hiện thị thông báo không được để trống	Hiện thị thông báo không được để trống	ok
Click vào Xóa Chính mình	Hiện thị thông báo bạn không được xóa chính mình	Hiện thị thông báo bạn không được xóa chính mình	ok
Click Vào Xóa	Hiện thị thông báo Bạn thực sự muốn xóa nhân viên này	Bạn thực sự muốn xóa nhân viên này	ok
Click vào nút yes trong thông báo	Hiện thị thông báo xóa thành công	Hiện thị thông báo xóa thành công	ok
Click vào nút làm mới xem có reset form hay không	Reset form quản lý nhân viên về ban đầu	Reset form quản lý nhân viên về ban đầu	ok
Nhập tên và click vào nút tìm	Tìm theo tên và hiển thị trên table	Tìm theo tên và hiển thị trên table	ok

2. Kiểm Thử Form Bán Hàng

Check Tính Hợp Lệ

Hành Động	Kết Quả Mong Đợi	Kết Quả Thực Tế	Ok hoặc
-----------	------------------	-----------------	---------

			Lỗi
Không nhập dữ liệu	Không được để trống	Không được để trống	OK
Nhập dữ liệu sai	<p>Hiện Thị thông báo theo từng ô và đổi màu các ô sang màu hồng</p> <p>Tên phải là tên tiếng việt hoặc không dấu từ 3-25 kí tự</p> <p>Số Lượng Phải là số</p>	<p>Hiện Thị thông báo từng ô và đổi màu các ô sang màu hồng</p> <p>Tên phải là tên tiếng việt hoặc không dấu từ 3-25 kí tự</p> <p>Số Lượng Phải là số</p>	OK

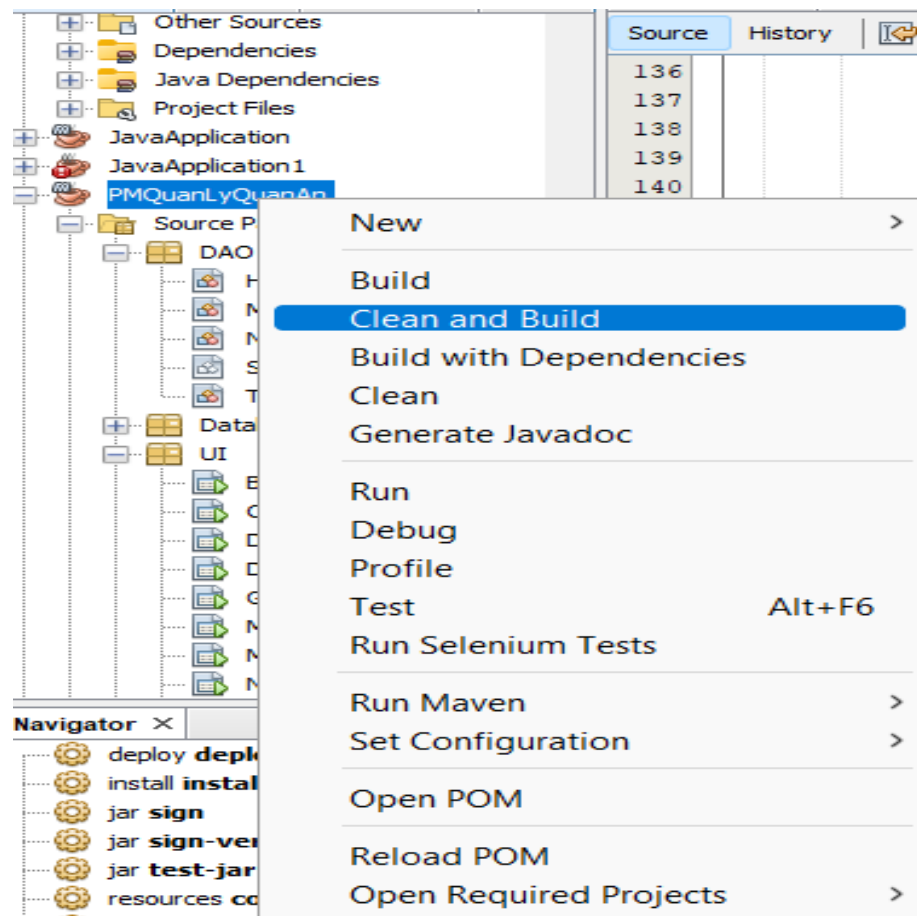
Check Form

Hành Động	Kết Quả Mong Đợi	Kết Quả Thực Tế	Ok hoặc Lỗi
Nhập dữ liệu đầy đủ và ấn nút Order	Hiện Thị thông báo thêm mới thành công và đổ dữ liệu lên bảng	Hiện Thị thông báo thêm mới thành công và đổ dữ liệu lên bảng	OK
Không nhập dữ liệu và ấn nút Order	Thông báo không được để trống và chuyển màu ô sang màu hồng	Thông báo không được để trống và chuyển màu ô sang màu hồng	OK
Nhập Id Hóa Đơn và ấn thanh toán	Hiện thị thông báo thanh toán thành công và in thông tin ra ô hóa đơn	Hiện thị thông báo thanh toán thành công và in thông tin ra ô hóa đơn	OK

VI. Đóng gói và triển khai

- ✓ Sử dụng phần mềm EXE4J để chuyển đổi jar sang exe
- ✓ Sử dụng phần mềm InnoSetup để đóng gói

1. Hướng dẫn chuyển đổi jar thành exe



exe4j

1. Welcome
2. Project type
3. Application info
4. Executable info
5. Java invocation
6. JRE
7. Splash screen
8. Messages
9. Compile executable
10. Finished

Welcome to exe4j!

With exe4j, you can produce windows executables to start up your Java applications in a secure, professional and attractive way.

This wizard leads you step by step to collect all required information to compile the executable. Press the help button at any time to get more information and detailed instructions.

Open Configuration

Click on [Open] to load an existing configuration. To try out exe4j, there are sample configuration files in the demo subdirectory of the exe4j installation.

Open

Click on [Reopen] to open a previously saved or loaded configuration.

Reopen ▼

License Information

This copy of exe4j 6.0.2 is unlicensed and may be used for evaluation purposes only. Nag screens and evaluation warnings are removed when purchasing exe4j at

<https://www.ej-technologies.com/download/exe4j/files>

Enter License

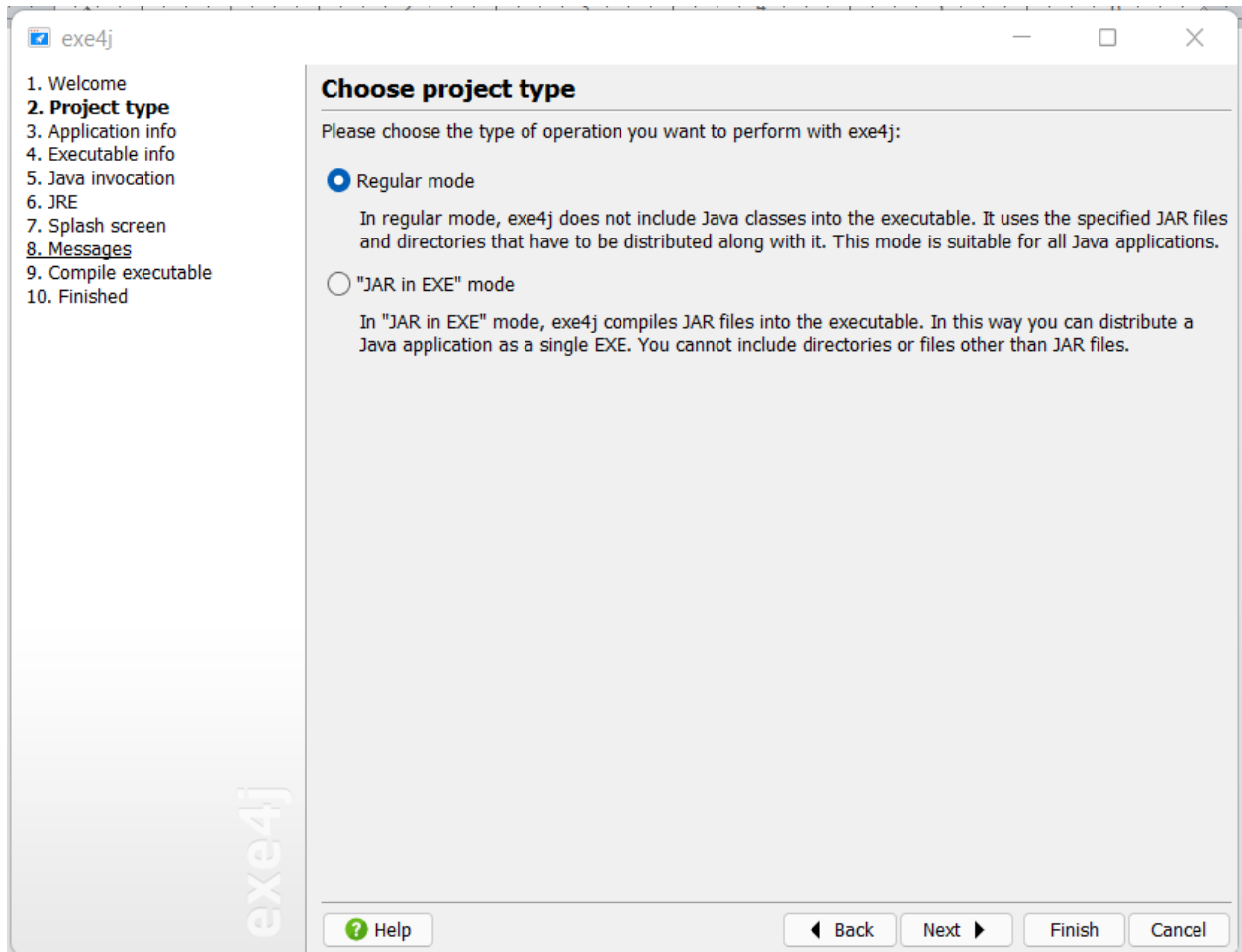
Help

Next ►

Finish

Cancel

exe4j



exe4j

1. Welcome

2. Project type

3. **Application info**

4. Executable info

5. Java invocation

6. JRE

7. Splash screen

8. Messages

9. Compile executable

10. Finished

exe4j

Configure application

General

Short name of your application: QuanLyQuanAn

Directories

Distribution source directory: D:\caijava\PMQuanLyQuanAn\target

The distribution source directory is the reference point for relative directories. If necessary, directories selected in a file chooser will be converted to relative directories.

Executable directory:

The generated executable will be copied to the executable directory. The executable directory must be below the distribution source directory. If the executable directory is left empty, it will be the same as the distribution source directory.

Help

Back

Next

Finish

Cancel

exe4j

1. Welcome

2. Project type

3. Application info

4. Executable info

Advanced options:

- Redirection
- Service options
- Version info
- 32-bit or 64-bit
- Manifest options

5. Java invocation

6. JRE

7. Splash screen

8. Messages

9. Compile executable

10. Finished

exe4j

Configure executable

Executable type: ☒ GUI application
☐ Allow -console parameter
☐ Console application
☐ Service

Executable name: .exe

☐ Icon File:

☐ Allow only a single running instance of the application

☒ Fail if an exception in the main thread is thrown

☒ Change working directory to: (relative to executable)

▼ Advanced Options

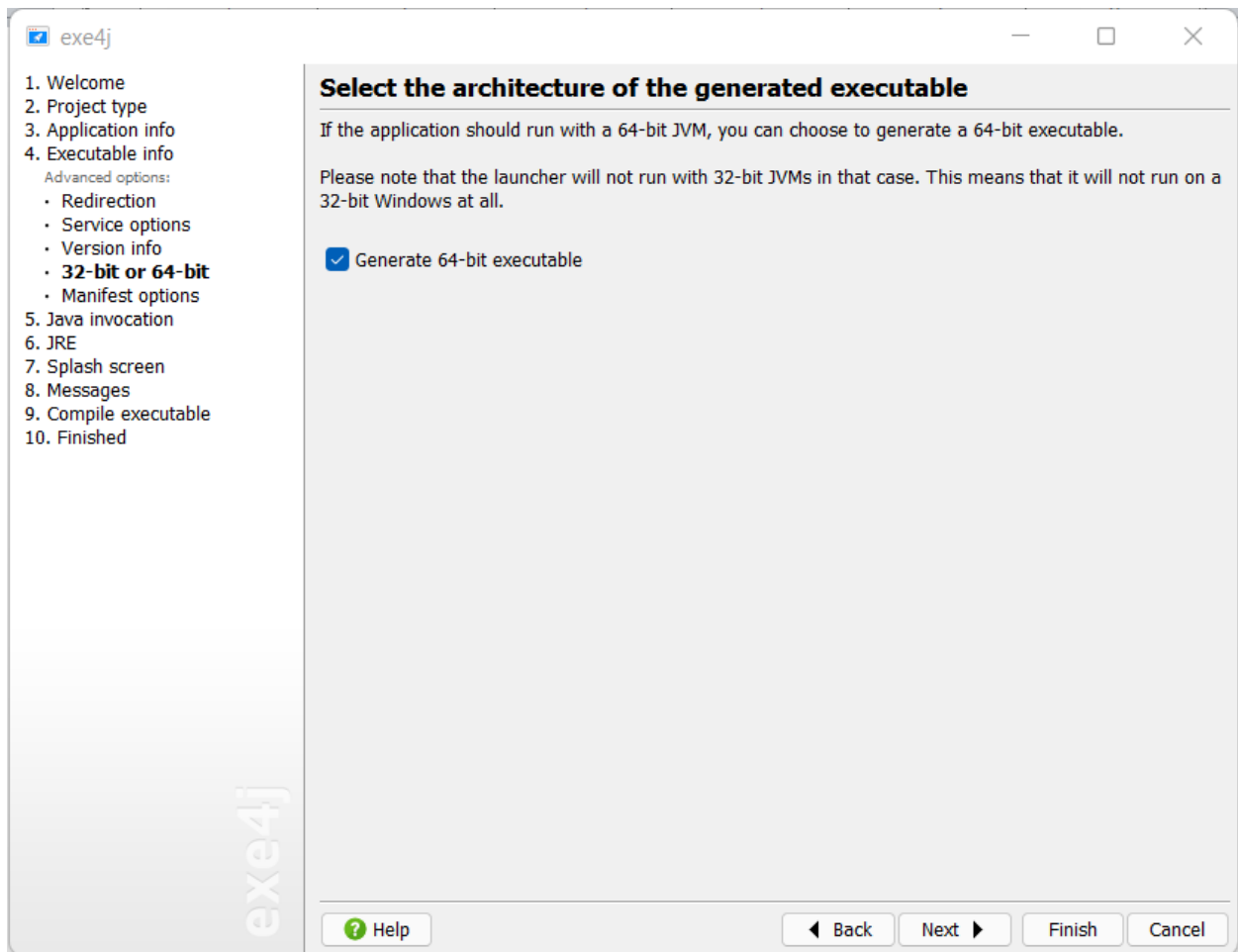
? Help

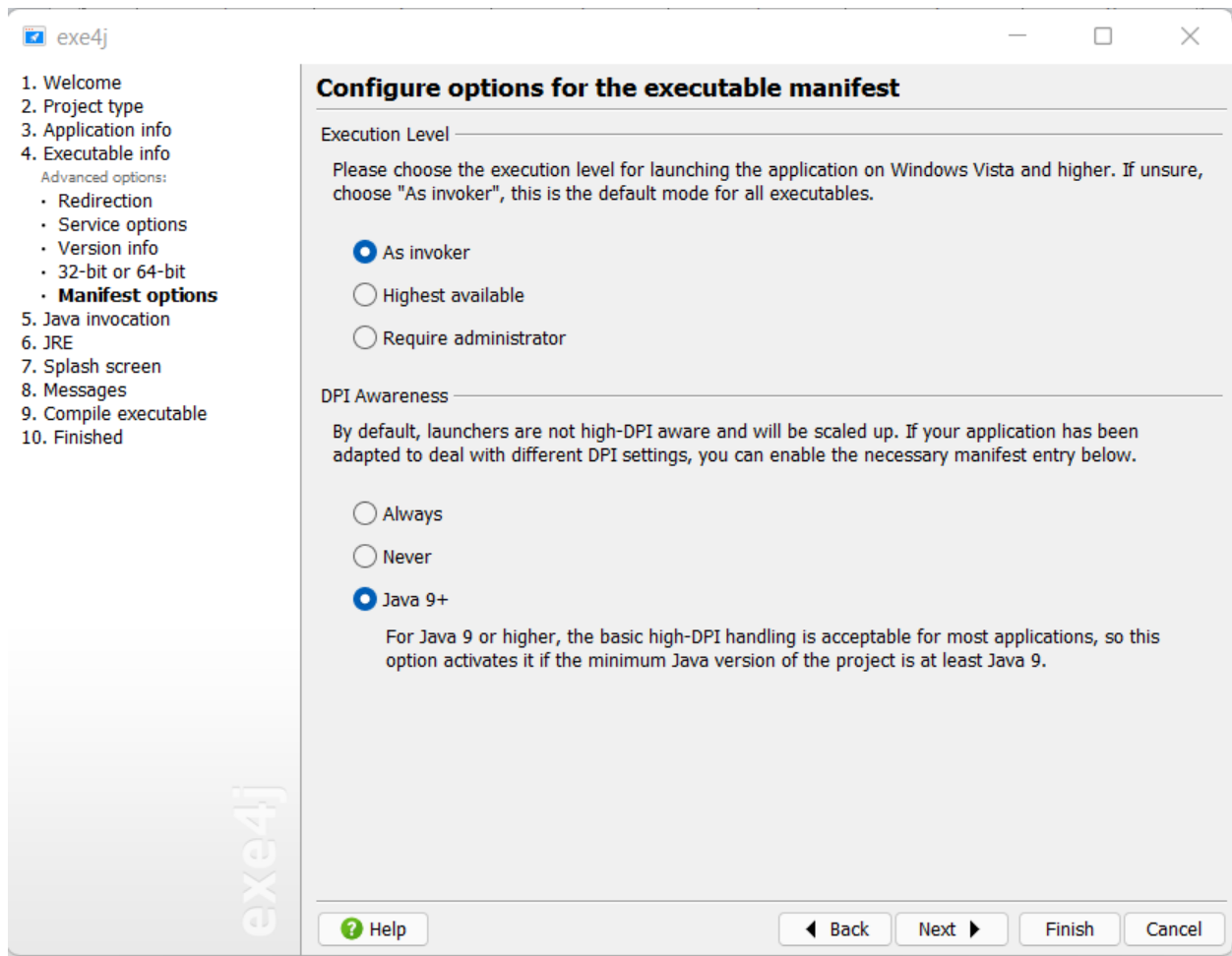
◀ Back

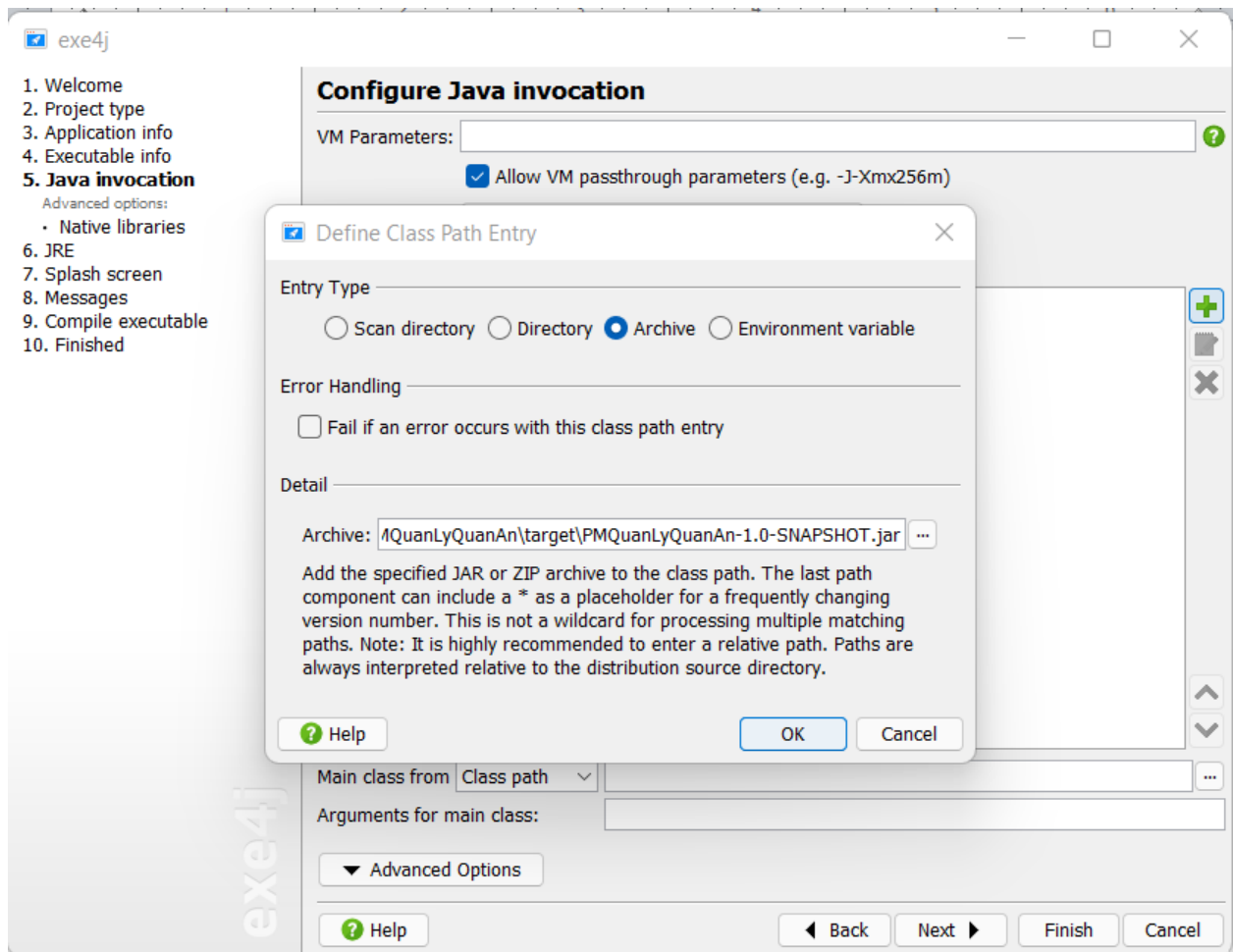
Next ▶

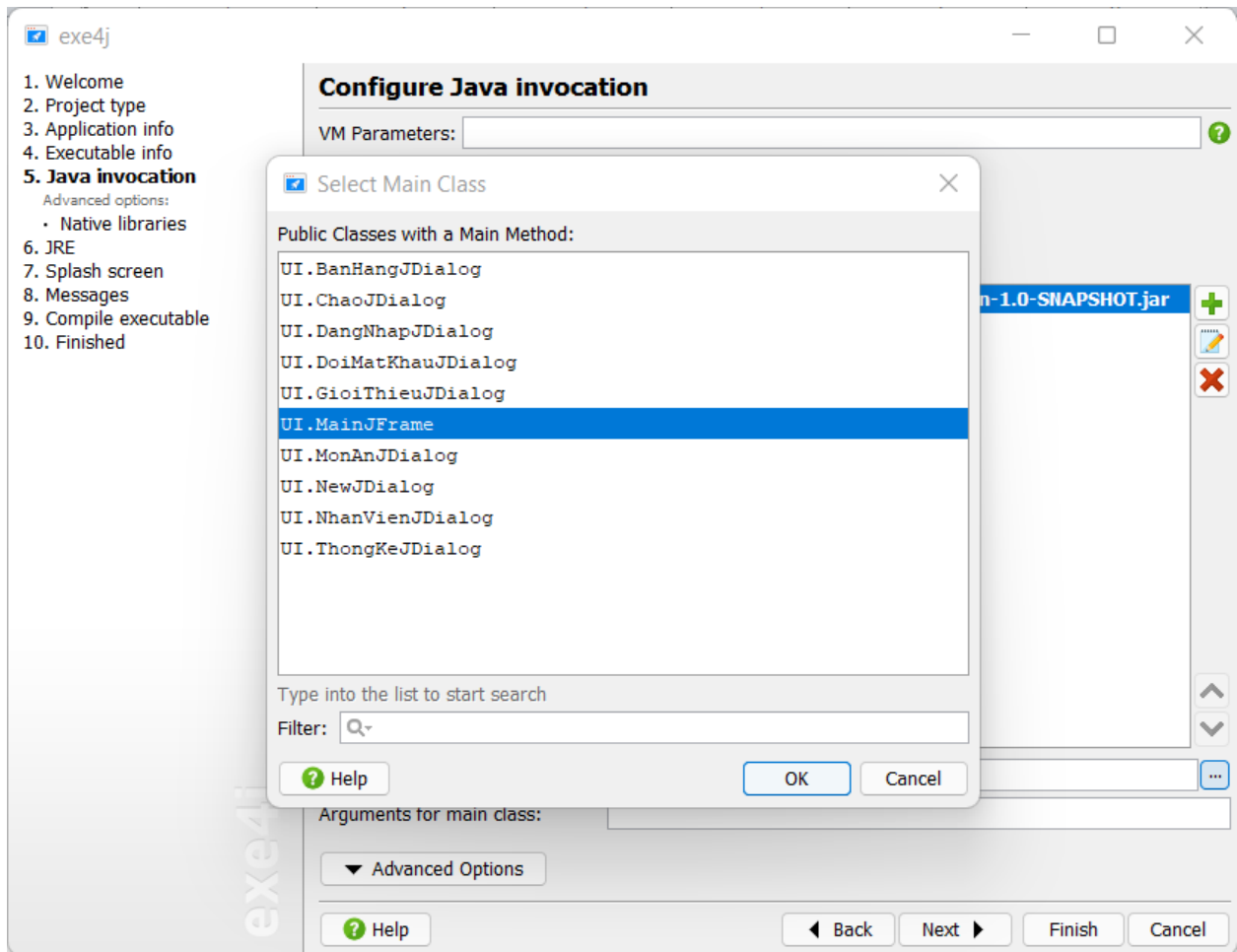
Finish

Cancel









exe4j

1. Welcome

2. Project type

3. Application info

4. Executable info

5. Java invocation

6. JRE

Advanced options:

Search sequence

Preferred VM

7. Splash screen

8. Messages

9. Compile executable

10. Finished

Configure JRE

Please define the minimum version requirements for the JRE that your application will be started with.

Minimum version:

1.7

(e.g. 1.7)

Maximum version:

1.8

(e.g. 1.8 or empty)

☐ Allow JREs with a beta version number

☐ Only allow JDKs and no JREs

▼ Advanced Options

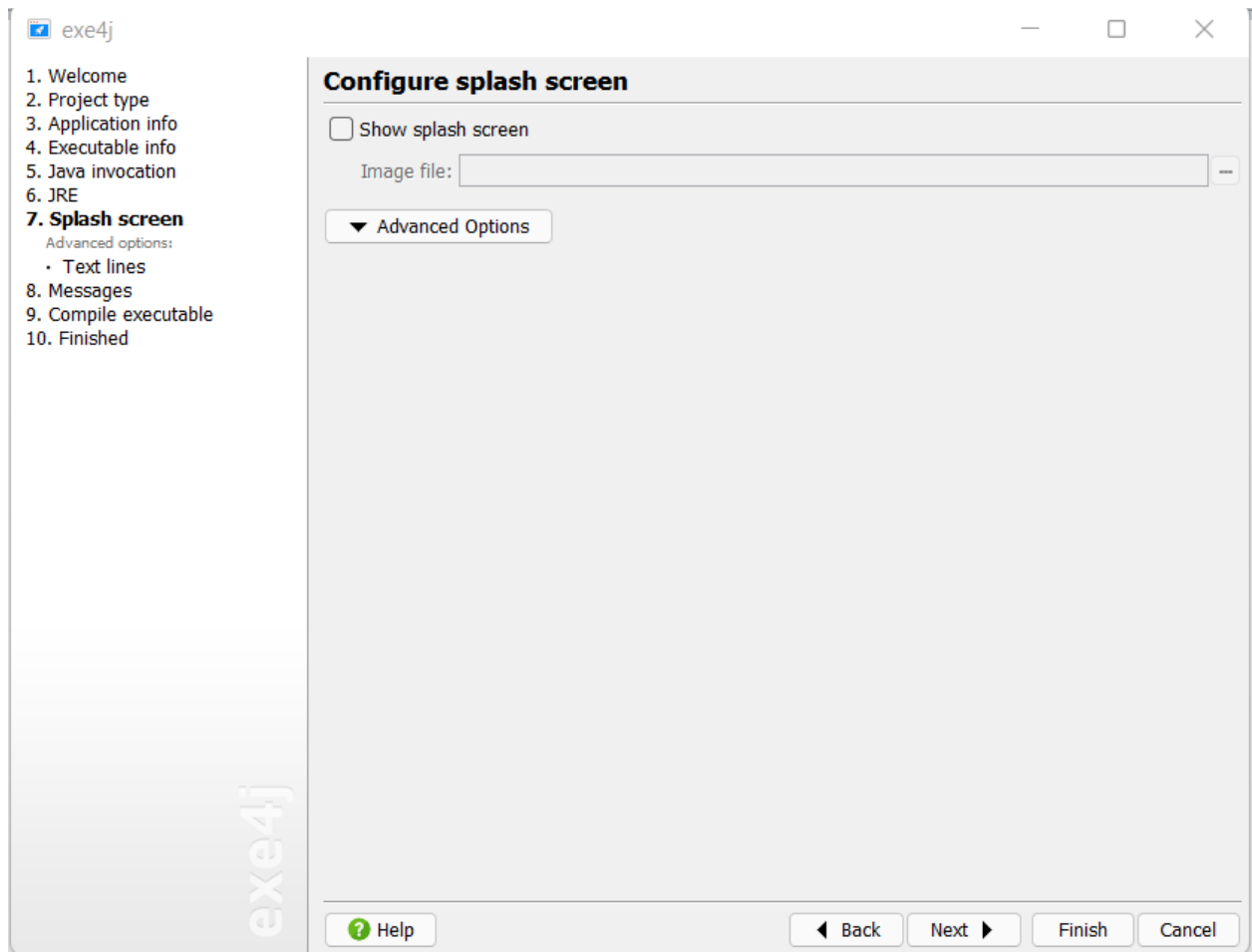
Help

◀ Back

Next ▶

Finish

Cancel



exe4j

1. Welcome
2. Project type
3. Application info
4. Executable info
5. Java invocation
6. JRE
7. Splash screen
- 8. Messages**
9. Compile executable
10. Finished

Configure messages of the executable

Language: English [en]

!	Id	Message
	MessageBoxTitle	Error
	NoJvmFound	No JVM could be found on your system.\nPlease defin...
	JvmInPath	The JVM found in your path is damaged.\nPlease rein...
	JvmEnvVar	The {0} environment variable does not\npoint to a w...
	JvmDefinedPath	The JVM found at {0} is damaged.\nPlease reinstall o...
	JvmStart	The JVM could not be started. The maximum heap si...
	JvmStart2	The JVM could not be started. The maximum heap si...
	FileNotFound	File or directory {0} not found or empty\n
	WrongParameters	Wrong Parameter Usage
	EnvVarNotDefined	The environment variable {0} has to be defined
	InternalError	An internal error occurred (error code: {0})
	MainClassNotFound	Couldn't load main class.
	MainMethodNotFound	Main method not found.
	ErrorDialogCaption	Error
	ErrorDialogOk	Ok
	ErrorDialogText	An error occurred while starting up {0}:
	PowerUserRequired	You must be at least Poweruser to run this program.
	NoJvmFound3264	No JVM could be found on your system.\nPlease defin...
	JvmInPath3264	The JVM found in your path is damaged.\nPlease rein...
	JvmEnvVar32	The {0} environment variable does not\npoint to a w...
	JvmEnvVar64	The {0} environment variable does not\npoint to a w...
	JvmDefinedPath32	The JVM found at {0} is damaged.\nPlease reinstall o...

Double-click on message column to edit a message.

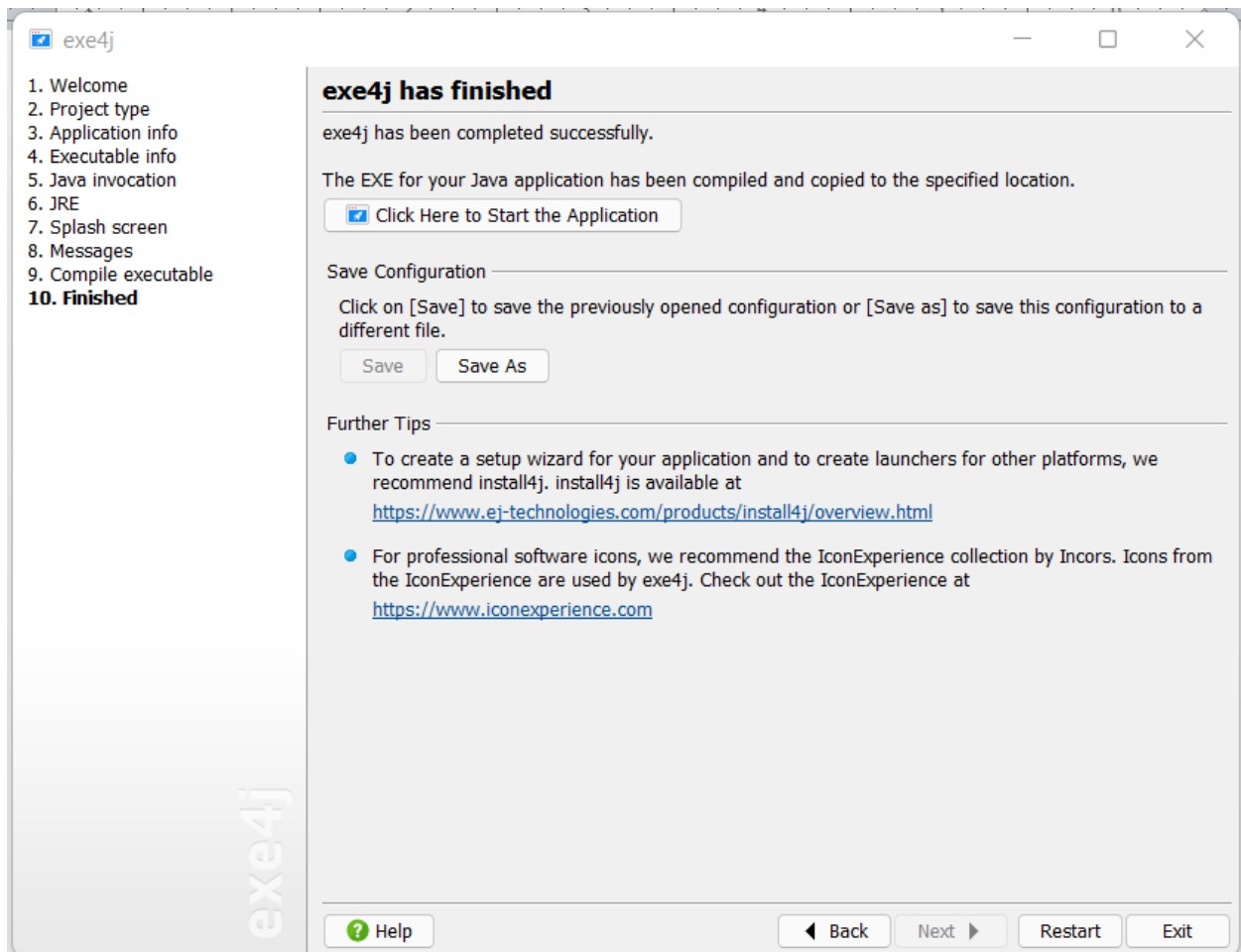
Help

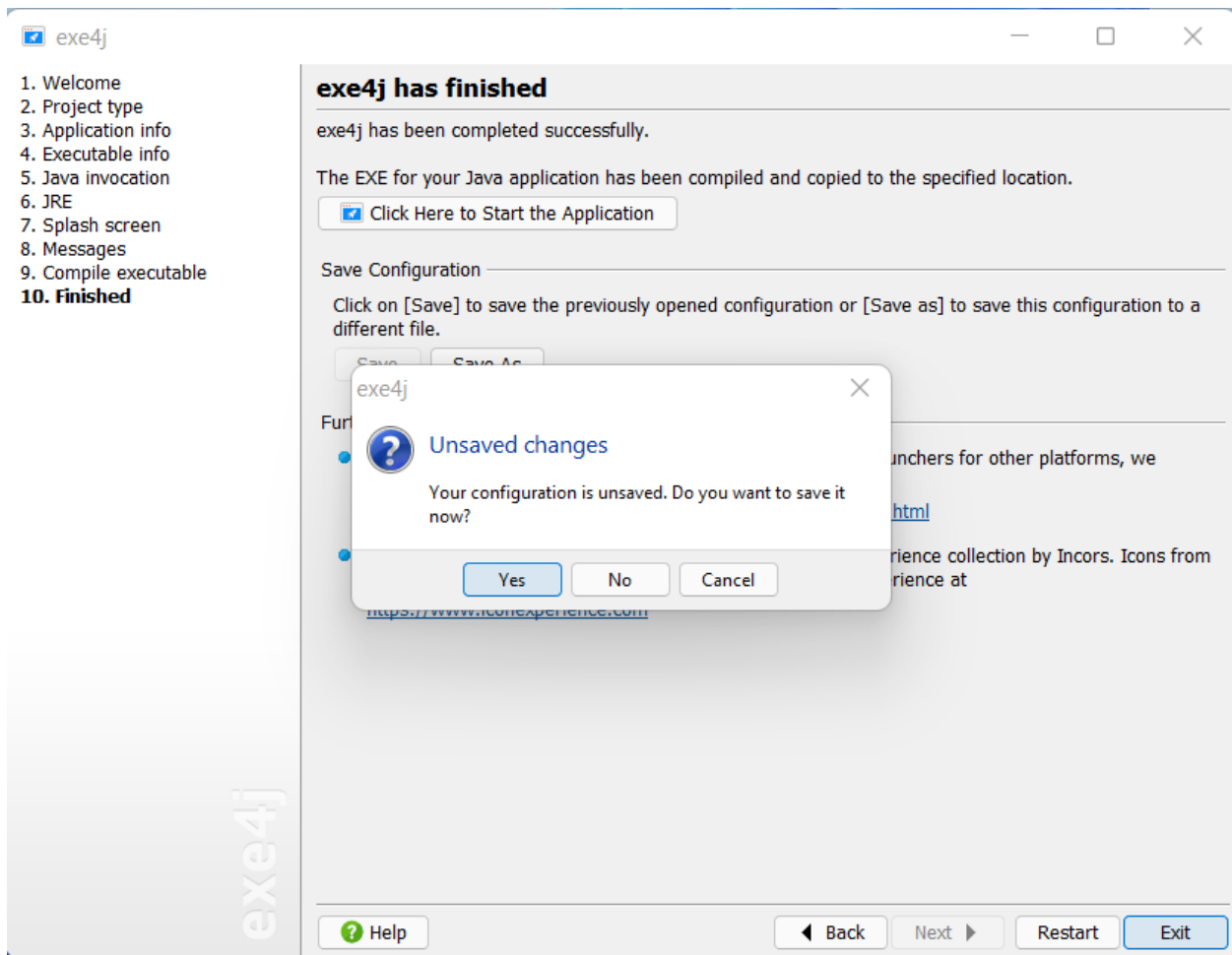
Back

Next

Finish

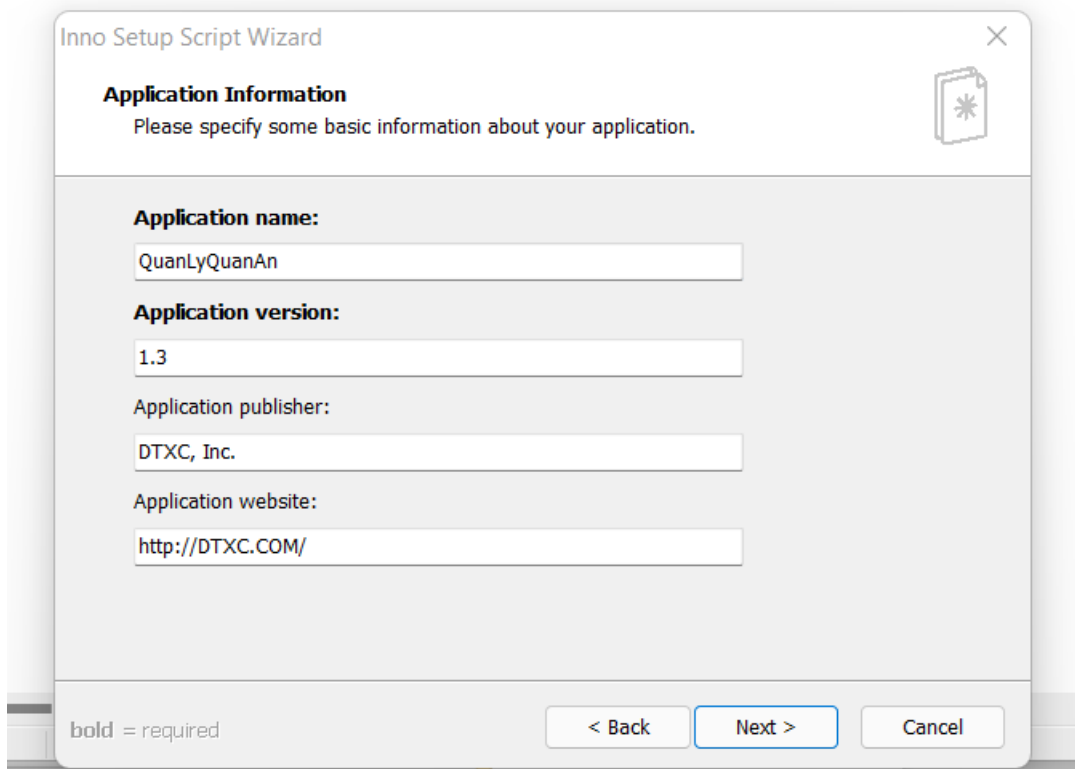
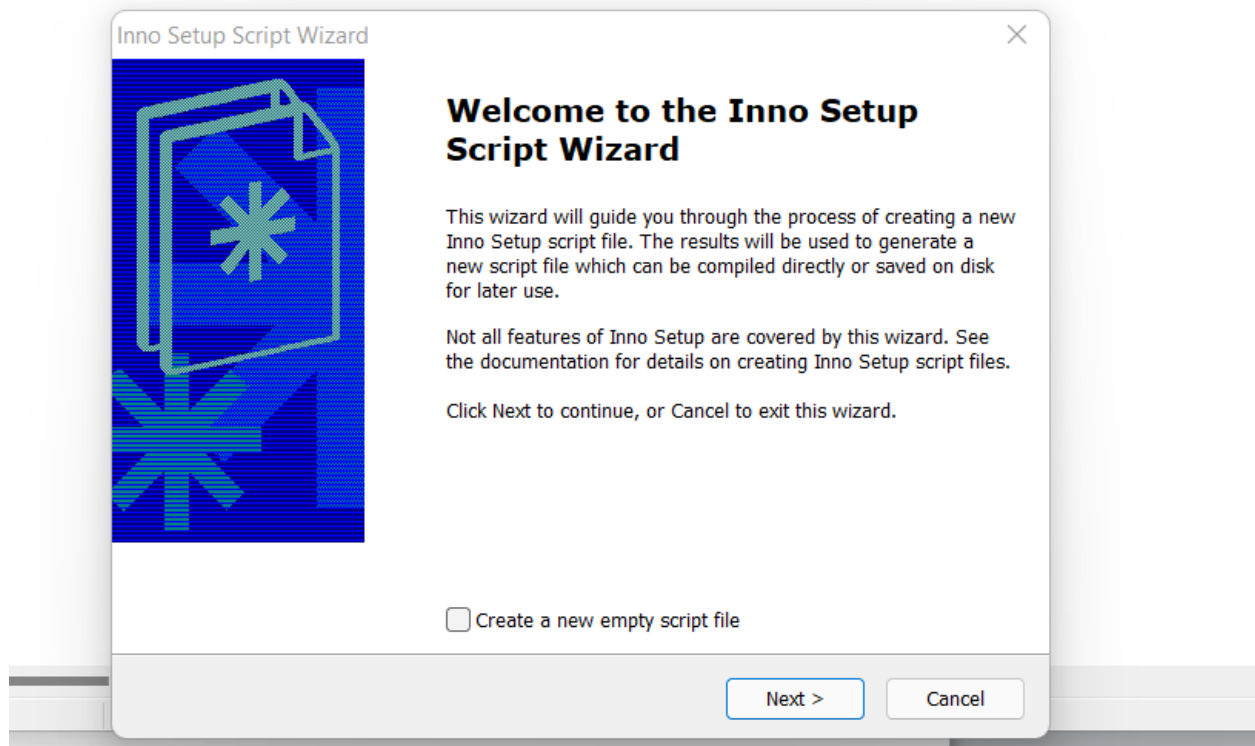
Cancel

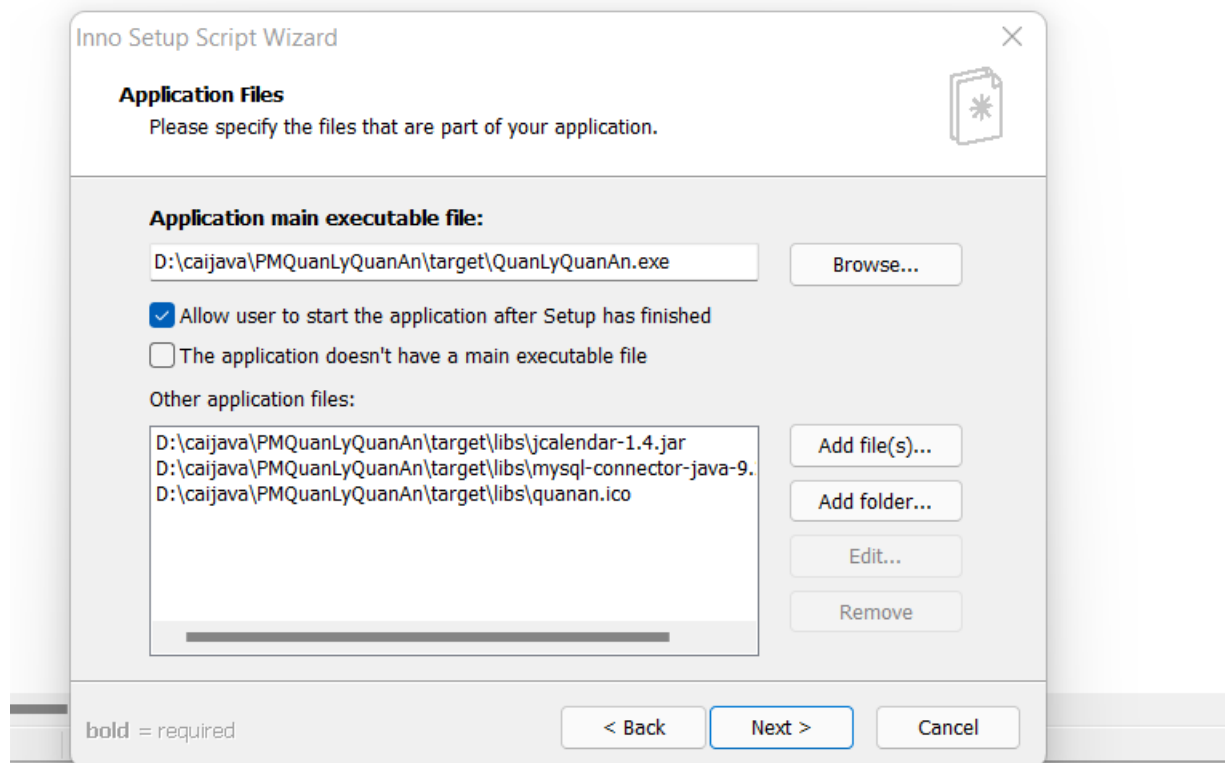
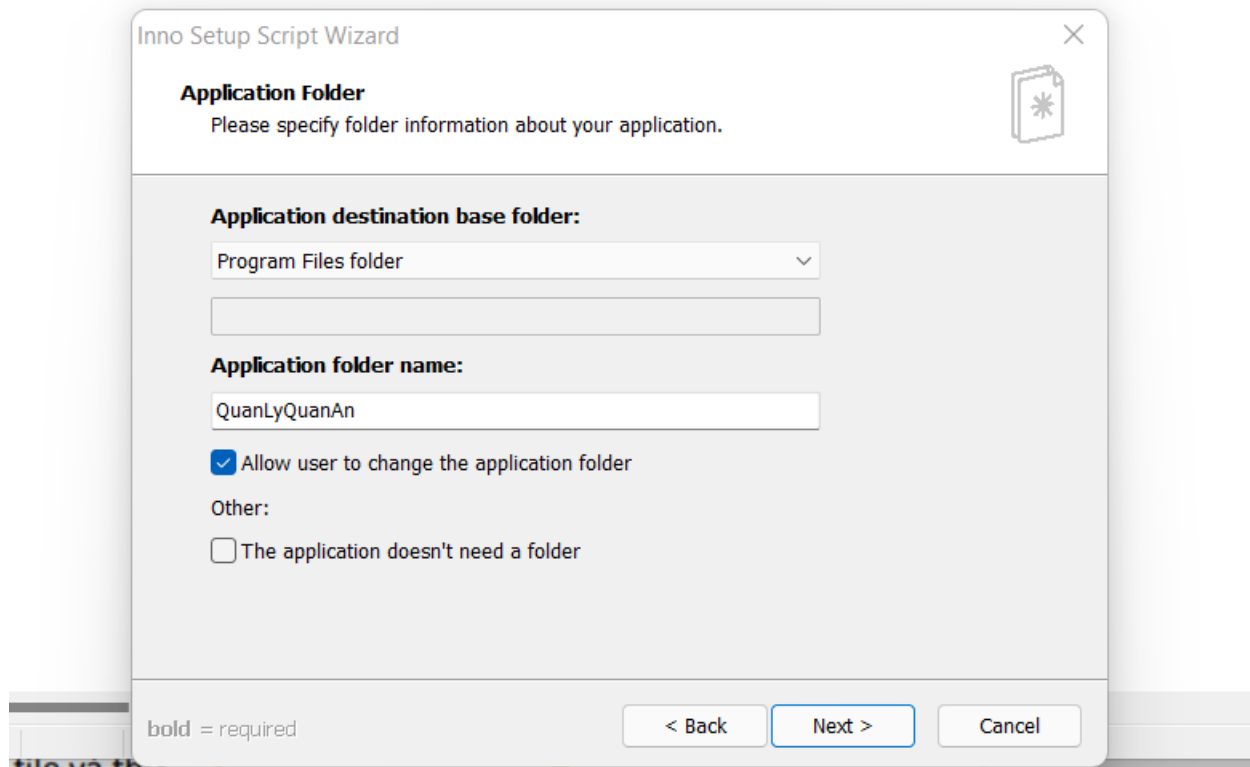




Name	Date modified	Type	Size
classes	12/5/2021 6:37 PM	File folder	
generated-sources	12/5/2021 6:36 PM	File folder	
libs	12/5/2021 6:48 PM	File folder	
maven-archiver	12/5/2021 6:37 PM	File folder	
maven-status	12/5/2021 6:36 PM	File folder	
test-classes	12/5/2021 6:42 PM	File folder	
PMQuanLyQuanAn-1.0-SNAPSHOT.jar	12/5/2021 6:37 PM	Executable Jar File	8,047 KB
QuanLyQuanAn.exe	12/5/2021 6:41 PM	Application	468 KB

File EXE có thể chạy được





Inno Setup Script Wizard

Application Shortcuts

Please specify which shortcuts should be created for your application.

☒ Create a shortcut to the main executable in the common Start Menu Programs folder

Application Start Menu folder name:

QuanLyQuanAn

☒ Allow user to change the Start Menu folder name

☐ Allow user to disable Start Menu folder creation

☐ Create an Internet shortcut in the Start Menu folder

☐ Create an Uninstall shortcut in the Start Menu folder

Other shortcuts to the main executable:

☒ Allow user to create a desktop shortcut

☒ Allow user to create a Quick Launch shortcut on older versions of Windows

bold = required

< Back Next > Cancel

Inno Setup Script Wizard

Application Documentation

Please specify which documentation files should be shown by Setup during installation.

License file:

Browse...

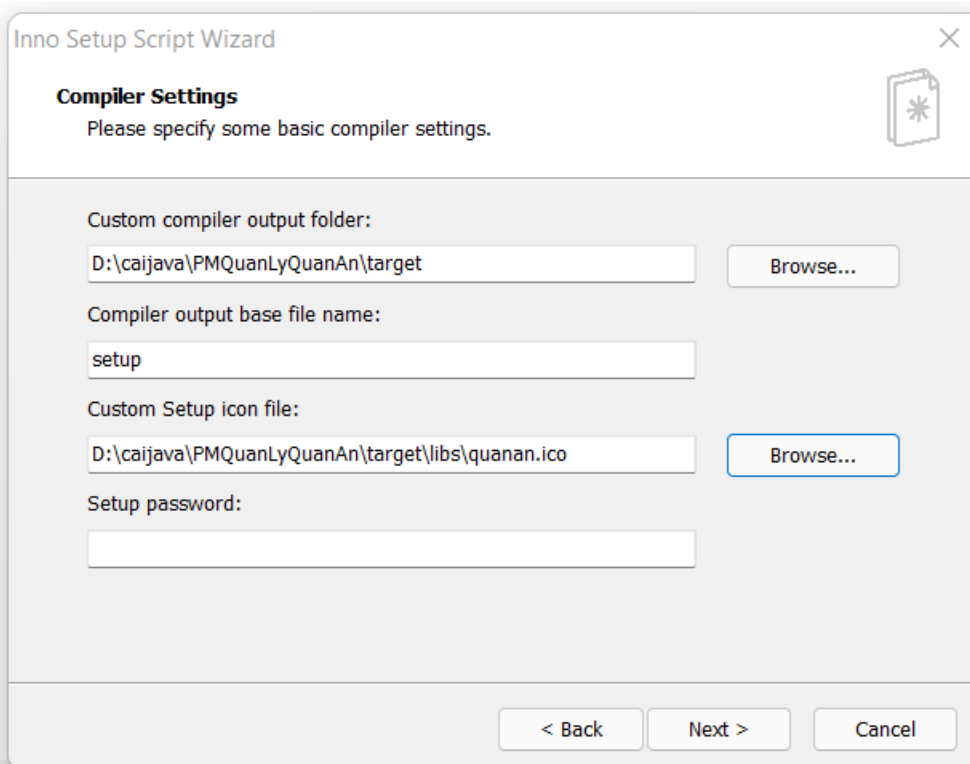
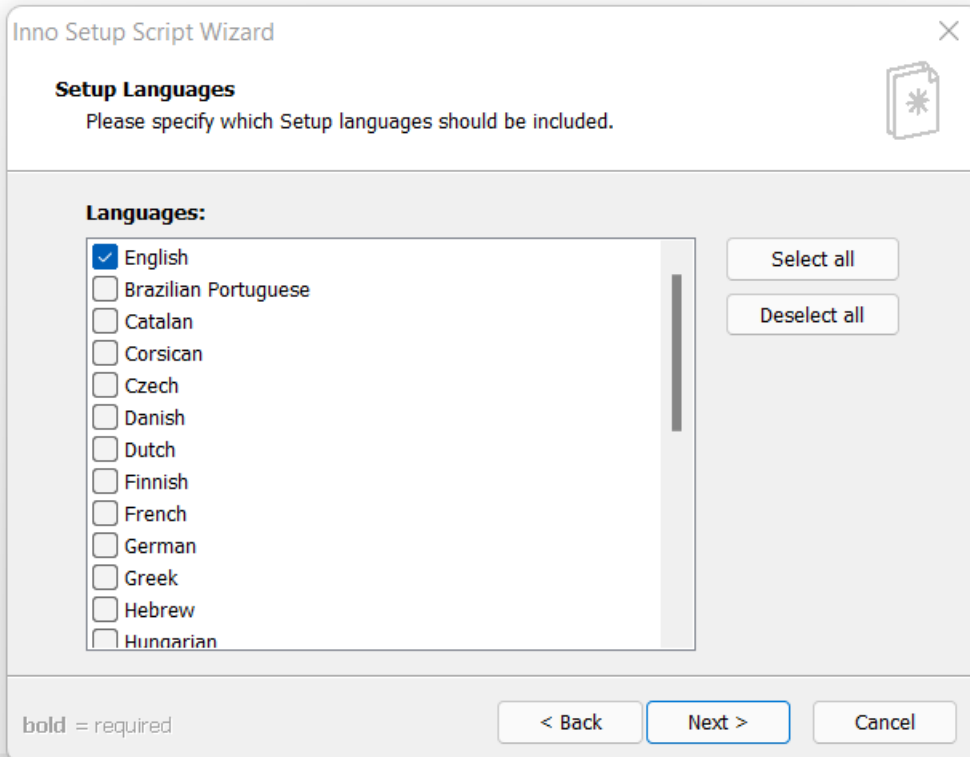
Information file shown before installation:

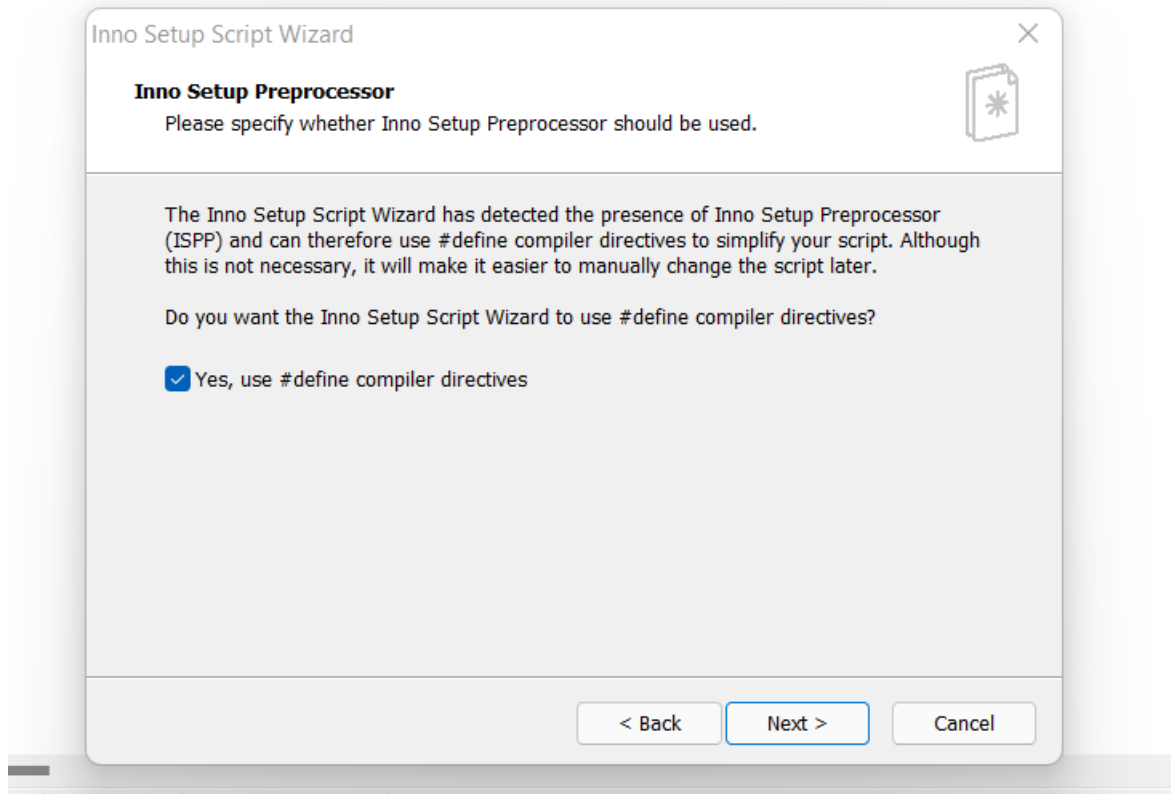
Browse...

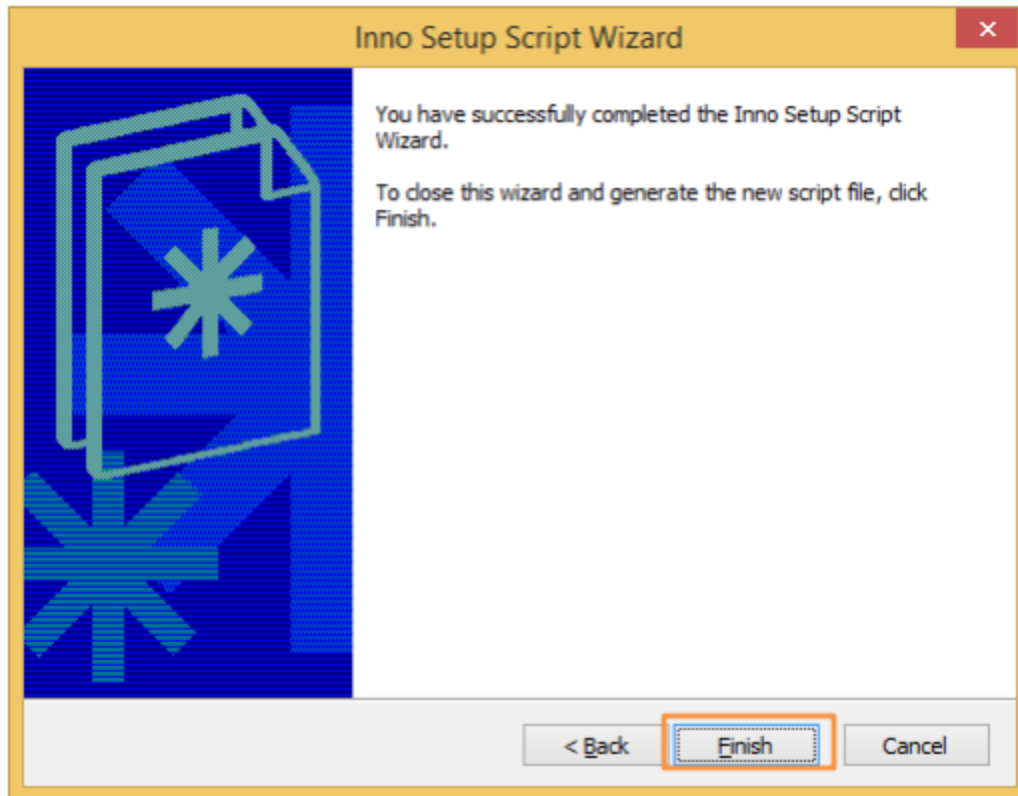
Information file shown after installation:

Browse...

< Back Next > Cancel







```

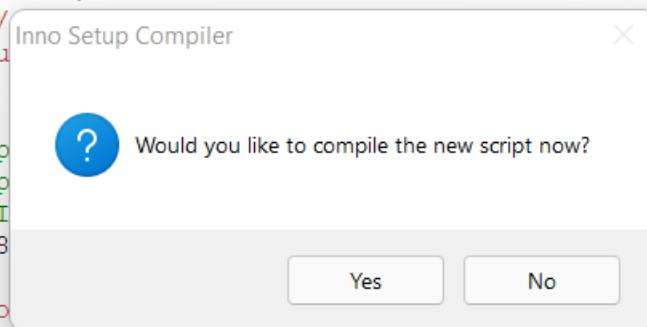
ne "QuanLyQuanAn"
rsion "1.3"
olisher "DTXC, Inc."
L "http://
eName "Qu

```

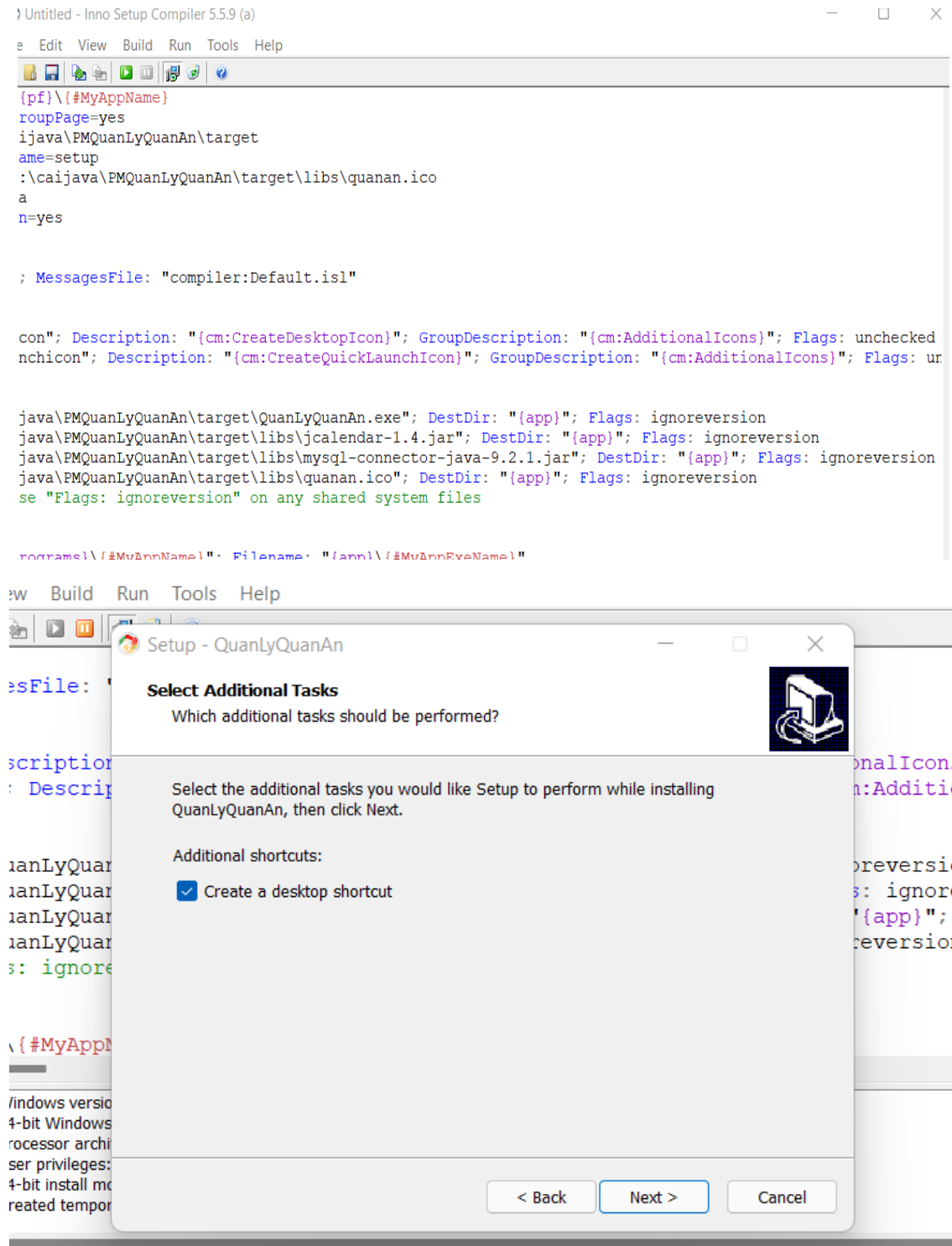
```

le of App
e same Ap
a new GUI
3-206F-48
Name}
AppVersio
yAppName} {#MyAppVersion}
MyAppPublisher}
={#MyAppURL}

```



tations.
e IDE.)



2. Hướng dẫn cài đặt triển khai

1. Chạy QuanLyQuanAn1.sql để tạo CSDL QuanLyQuanAn

2. Đặt mật khẩu sa của SQL Server là 123456
3. Chạy file setup và cài đặt (chú ý môi trường JDK1.8+)
4. Chạy ứng dụng: Run as Administrator

Các bạn lưu ý không để setup.exe trong thư mục có dấu tiếng việt (chứa trực tiếp hoặc gián tiếp) nhé, để tránh xảy ra lỗi trong quá trình cài đặt.

3. Hướng dẫn sử dụng phần mềm

- ✓ Đăng nhập với tài khoản username: admin và password:admin
- ✓ Cấp các tài khoản có vai trò là admin và cập nhật tài khoản cũ sang vai trò nhân viên
- ✓ Đăng nhập tài khoản mới và xóa tài khoản admin
- ✓ Xem hướng dẫn sử dụng trong menu trợ giúp của phần mềm

Hướng dẫn bằng video

<https://www.youtube.com/watch?v=FB3jKNJuLNQ>

4. Kết Luận Và Hướng Phát Triển

1. Những việc đã làm được

- ✓ Chức năng của Admin:
 - Quản lý Món Ăn
 - Quản lý Nhân Viên
 - Quản lý Bán Hàng
 - Thống Kê
- ✓ Chức năng của Nhân viên:
 - Quản lý Nhân Viên
 - Quản lý Món Ăn
 - Quản lý Bán Hàng
- ✓ Những việc chưa làm được:
 - Đặt Bàn
 - Quản lý đặt bàn

2. Hướng phát triển

- ✓ Phát triển quản lý chuỗi Quán Ăn, Admin có thể xem xét các hoạt động chung của các quán ăn.
- ✓ Cải Tiến Giao Diện Và Bảo Mật Cao Hơn...
- ✓ Nâng cấp lên thêm chức năng:
 - Phân biệt rõ ràng chức năng của Bếp và Phục vụ
 - Quản lý khách hàng thân thiết

- Thêm phần giảm giá cho khách hàng thân thiết