|  |
| --- |
| FPT POLYTECHNIC |
| **DỰ ÁN 1** |
| NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM) |



|  |
| --- |
| HÀ NỘI 2018 |

|  |
| --- |
| GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN VĂN NAM |
| SINH VIÊN THỰC HIỆN |
| 1. Ngô Hữu Duần (PD02217), Trưởng nhóm 2. Đỗ Thành Thịnh (PD02189) 3. Trần Ngọc Hải (PD02302) |

Contents

[1 Phân tích 1](#_Toc529305033)

[1.1 Hiện trạng 1](#_Toc529305034)

[1.2 Yêu cầu hệ thống 1](#_Toc529305035)

[1.3 Use case 2](#_Toc529305036)

[2 Thiết kế 6](#_Toc529305037)

[2.1 Mô hình triển khai 6](#_Toc529305038)

[2.2 Thiết kế CSDL 6](#_Toc529305039)

[2.2.1 Sơ đồ quan hệ thực thể 6](#_Toc529305040)

[2.2.2 ERD Diagram level 1 7](#_Toc529305041)

[2.2.3 ERD Diagram level 2 8](#_Toc529305042)

[2.2.4 Thiết kế chi tiết thực thể 8](#_Toc529305043)

[2.3 Thiết kế giao diện 12](#_Toc529305044)

[2.3.1 Sơ đồ tổ chức giao diện 12](#_Toc529305045)

[2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ 13](#_Toc529305046)

[3 Thực hiện viết mã 23](#_Toc529305047)

[3.1 Viết mã tạo CSDL 23](#_Toc529305048)

[3.1.1 Tạo CSDL 23](#_Toc529305049)

[3.1.2 SQL truy vấn và thao tác 23](#_Toc529305050)

[3.1.3 Các thủ tục lưu tổng hợp thống kê 23](#_Toc529305051)

[3.2 Lập trình JDBC 23](#_Toc529305052)

[3.2.1 Lớp hỗ trợ 23](#_Toc529305053)

[3.2.2 Model class - Các lớp mô tả dữ liệu 23](#_Toc529305054)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 23](#_Toc529305055)

[3.3 Viết mã cho ứng dụng 23](#_Toc529305056)

[3.3.1 Xử lý Form X 23](#_Toc529305057)

[3.3.2 Xử lý Form Y 23](#_Toc529305058)

[4 Kiểm thử 23](#_Toc529305059)

[4.1 Kiểm thử form X 24](#_Toc529305060)

[4.2 Kiểm thử form Y 24](#_Toc529305061)

[5 Đóng gói và triển khai 24](#_Toc529305062)

[5.1 Hướng dẫn chuyển đổi jar thành exe 24](#_Toc529305063)

[5.2 Hướng dẫn cài đặt triển khai 24](#_Toc529305064)

[5.3 Hướng dẫn sử dụng phần mềm 24](#_Toc529305065)

# Phân tích

## Hiện trạng

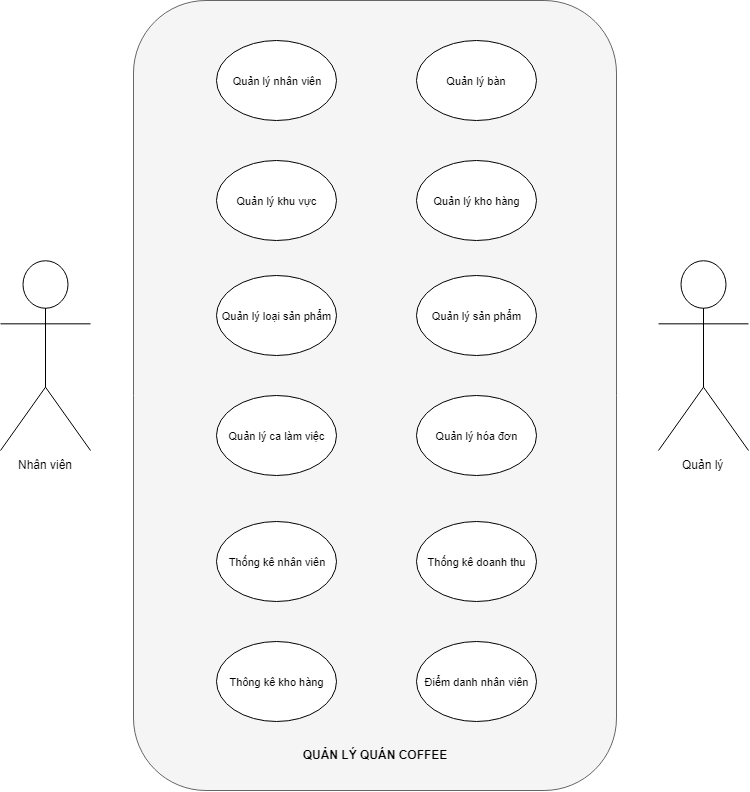
Sự bùng nổ và phát triển mạnh mẽ của công nghệ thông tin, muốn hiện đại hóa đất nước thì phải áp dụng CNTT vào tất cả các lĩnh vực. Các phần mềm hiện tại hiện đang hỗ trợ cho người dung thuận tiện sử dụng, thời gian xử lý nhanh chóng, và một số nghiệp vụ được tự động hóa cao, các phần mềm giúp tiết kiệm thời gian, công sức của con người, tăng độ chính xác và hiệu quả trong công việc.

Vì vậy, Việc quản lý quán coffee sẽ tiết kiệm được rất nhiều nhân viên, tiết kiệm được thời gian và tiền bạc. Phần mềm quản lý quán coffee sẽ có thể quản lý: nhân viên, hóa đơn, nhập hàng, bán hàng, đặc biệt là vấn đề thống kê sử dụng, phần mềm sẽ giúp thống kê nhanh chóng, tiện lợi và tránh sai sót.

## Yêu cầu hệ thống

* Yêu cầu chức năng nghiệp vụ:
  + Gọi món
  + Đổi món
  + Quản lý nhân viên
  + Quản lý bàn
  + Quản lý khu vực
  + Quản lý kho hàng
  + Quản lý các loại sản phẩm
  + Quản lý sản phẩm
  + Quản lý ca làm việc
  + Quản lý hóa đơn
  + Thống kê doanh thu
  + Thống kê kho hàng
  + Thống kê nhân viên
  + Phân quyền truy cập
  + Điểm danh nhân viên
  + Đăng nhập, đăng xuất
* Yêu cầu về bảo mật
  + Tất cả các thành viên phải đăng nhập mới sử dụng được phần mềm.
  + Người quản lý được phép dùng tất cả các chức năng.
  + Nhân viên không được truy cập vào thống kê doanh thu, quản lý nhân viên, quản lý kho hàng, quản lý lương.
* Yêu cầu về môi trường và công nghệ
  + Ứng dụng phải được thực với công nghệ Swing và JDBC chạy trên mọi hệ điều hành với môi trường JDK tối thiểu 1.8.
  + Hệ quản trị CSDL SQL Server 2008 trở lên.

## Use case



* Chi tiết các chức năng:

Mỗi chức năng quản lý bao gồm các chức năng con:

* + Xem: xem tất cả và chi tiết một mục.
  + Thêm: thêm mới vào cơ sở dữ liệu.
  + Xóa: xóa theo mã.
  + Sửa: cập nhật dữ liệu đang xem.
  + Tìm kiếm: tìm kiếm theo điều kiện.
  + Điều hướng: di chuyển đến dữ liệu của bản ghi chi tiết trước và sau.
  + Các thức năng thao tác dữ liệu (thêm, sửa, xóa) cần được kiểm lỗi hợp lý với dữ liệu.
* Chức năng tổng hợp thống kê số liệu:
  + Thống kê kho hàng theo loại sản phẩm:
    - Tên sản phẩm.
    - Số lượng tồn kho.
    - Số lượng đã bán.
  + Thống kê doanh thu theo năm, tháng:
    - Thời gian.
    - Doanh thu.
  + Thống kê nhân viên:
    - Mã nhân viên.
    - Số ngày làm việc.
    - Lương.
    - Hoa hồng.
* Chức năng đăng nhập và đăng xuất:
  + Đăng nhập: để sử dụng phần mềm thì bắt buộc người dùng phải đăng nhập.
  + Đăng xuất: đăng nhập lại với người khác hoặc dừng lại đi đâu đó.
* Chi tiết về yêu cầu bảo mật:
  + Cả nhân viên và quản lý đều phải đăng nhập mới sử dụng được các chức năng trong phần mềm.
  + Quản lý: sử dụng được tất cả các chức năng của phần mềm.
  + Nhân viên:
    - Không được phép sử dụng các chức năng xóa trong các chức năng quản lý.
    - Không được phép xem chức năng thống kê và quản lý nhân viên.
* Thông tin các thực thể:

|  |  |
| --- | --- |
| **Thực thể** | **Thuộc tính** |
| Nhân viên | * Mã nhân viên * Mã ca làm việc * Tên nhân viên * Ngày sinh * Số điện thoại * Giới tính * Email * Số ngày làm việc * Mật khẩu nhân viên * Vai trò * Ghi chú |
| Bàn | * Mã bàn * Mã khu vực bàn * Trạng thái |
| Khu vực | * Mã khu vực * Tên khu vực * Ghi chú * Vị trí |
| Kho hàng | * Mã kho hàng * Mã sản phẩm * Ngày nhập * Mã Nhân Viên * Số lượng * Ghi chú |
| Loại sản phẩm | * Mã loại sản phẩm * Tên loại sản phẩm |
| Ca làm việc | * Mã ca làm việc * Tên ca làm việc * Thời gian bắt đầu * Thời gian kêt thúc * Ghi chú |
| Sản phẩm | * Mã sản phẩm * Tên sản phẩm * Mã loại Sản Phẩm * Giá bán * Trạng thái * Ghi chú |
| Hóa đơn | * Mã hóa đơn * Mã sản phẩm * Số lượng mua * Mã bàn * Thanh tiền * Trạng thái * Ghi chú * Ngày thanh toán |

# Thiết kế

## Mô hình triển khai

Ứng dụng phần mềm được xây dựng để phục vụ cho nhiều người dùng nhưng cơ sở dữ liệu thì lưu trữ tập trung.

* Cần một máy cài SQL Server 2008+. Máy này cần hệ điều hành Window XP trở lên
* Các máy nhân viên cần JDK 1.8+ với hệ điều hành bất kỳ.

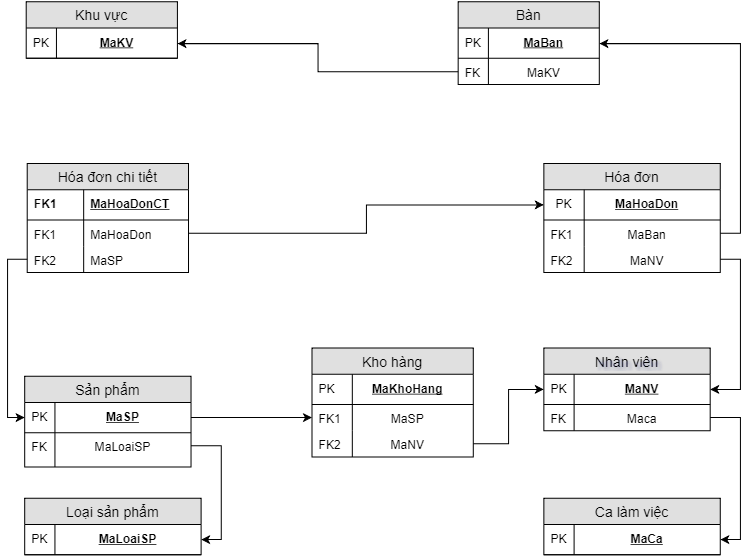
## Thiết kế CSDL

### Sơ đồ quan hệ thực thể

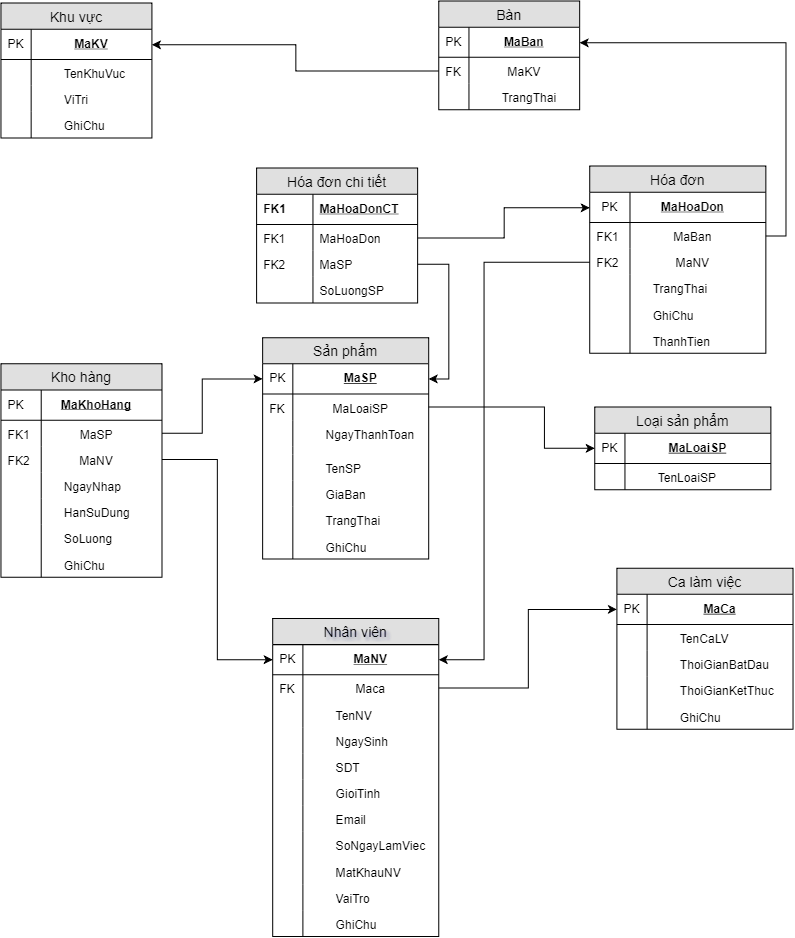
Trong hệ thống đã được mô tả trên chúng ta thấy có các thực thể sau:

* Nhân viên: quản lý thông tin, vai trò và số ngày làm việc của nhân viên.
* Bàn: quản lý bàn và trạng thái của bàn.
* Khu vực: quản lý các khu vực bàn.
* Kho hàng: quản lý số lượng hàng tồn kho, ngày nhập hàng, nhân viên nhập hàng.
* Loại sản phẩm: quản lý các loại sản phẩm.
* Ca làm việc: quản lý thời gian làm và kết thúc của ca.
* Sản phẩm: quản lý giá bán và trạng thái hàng.
* Hóa dơn: quản lý hóa đơn sau khi gọi món.

### ERD Diagram level 1



### ERD Diagram level 2



### Thiết kế chi tiết thực thể

#### Nhân viên

Bảng nhân viên lưu thông tin của nhân viên có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaNhanVien | INT | PK, Tự tăng | Mã nhân viên |
| HoTen | NVARCHAR(50) | NOT NULL | Họ và tên |
| MatKhau | NVARCHAR(50) | NOT NULL | Mật khẩu đăng nhập |
| VaiTro | BIT | DEFAULT 0 | Vai trò, 1 là quản lý |
| NgaySinh | DATE | NOT NULL | Ngày sinh |
| GioiTinh | BIT | DEFAULT 0 | Giới tính |
| Email | NVARCHAR(50) | NOT NULL, UNIQUE | Email |
| DienThoai | NVARCHAR(10) | NOT NULL, UNIQUE | Số điện thoại |
| GhiChu | NVARCHAR(255) | NULL | Ghi chú |
| MaCa | INT | FK, NOT NULL | Khóa ngoại liên kết với bảng CA LÀM VIỆC |
| SoNgayLV | INT | DEFAULT 0 | Số ngày làm việc của nhân viên, được đưa về 0 khi hết tháng. |

#### Bàn

Bảng bàn lưu thông tin của bàn có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaBan | INT | PK, NOT NULL | Mã bàn |
| MaKhuVuc | NCHAR(7) | FK, NOT NULL | Khóa ngoại liên kết đến bảng KHU VỰC |
| TrangThai | BIT | DEFAULT 0 | Trạng thái, 1 là có khách |

#### Khu vực

Bảng khu vực lưu thông tin của khu vực có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaKhuVuc | INT | PK, Tự tăng | Mã khu vực |
| TenKhuVuc | NVARCHAR(50) | NOT NULL, UNIQUE | Tên khu vực |
| ViTri | NVARCHAR(50) | NOT NULL | Vị trí khu vực |
| GhiChu | NVARCHAR(255) | NULL | Ghi chú |

#### Kho hàng

Bảng kho hàng lưu các thông tin của kho hàng đang tồn kho có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaKhoHang | INT | PK, Tự tăng | Mã kho hàng |
| MaSanPham | INT | FK, NOT NULL | Khóa ngoại liên kết với bảng SẢN PHẨM |
| NgayNhap | DATE | NOT NULL | Ngày nhập |
| MaNhanVien | INT | FK, NOT NULL | Khóa ngoại liên kết với bảng NHÂN VIÊN |
| SoLuong | INT | DEFAULT 0 | Số lượng tồn kho, >= 0 |
| GhiChu | NVARCHAR(255) | NULL | Ghi chú |
| HanSuDung | DATE | NULL | Hạn sử dụng của sản phẩm, 30 ngày kể từ ngày nhập |

#### Loại sản phẩm

Bảng loại sản phẩm lưu thông tin các loại sản phẩm có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaLoaiSP | INT | PK, Tự tăng | Mã loại sản phẩm |
| TenLoaiSP | NVARCHAR(50) | NOT NULL, UNIQUE | Tên loại sản phẩm |

#### Ca làm việc

Bảng ca làm việc lưu thông tin của các ca làm việc có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaCa | INT | PK, Tự tăng | Mã ca làm việc |
| BatDau | DATE | NOT NULL | Thời gian bắt đầu ca làm việc |
| KetThuc | DATE | NOT NULL | Thời gian kết thúc ca làm việc |
| GhiChu | NVARCHAR(255) | NOT NULL | Ghi chú |
| TenCaLV | NVARCHAR(50) | NOT NULL, UNIQUE | Tên ca làm việc |

#### Sản phẩm

Bảng sản phẩm để lưu các thông tin của sản phẩm bán ra có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaSanPham | INT | PK, Tự tăng | Mã sản phẩm |
| TenSanPham | NVARCHAR(50) | NOT NULL | Tên sản phẩm |
| MaLoaiSP | INT | FK, NOT NULL | Khóa ngoại liên kết đến bảng LOẠI SẢN PHẨM |
| GiaBan | FLOAT | NOT NULL | Đơn giá của sản phẩm, >= 0 |
| TrangThai | BIT | DEFAULT 0 | Trạng thái còn hàng hay hết hàng của sản phẩm, 1 là còn hàng. |
| GhiChu | NVARCHAR(255) | NOT NULL | Ghi chú |

#### Hóa đơn

Bảng hóa đơn lưu thông tin của các hóa đơn có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaHoaDon | INT | PK, Tự tăng | Mã hóa đơn |
| MaSanPham | INT | FK, NOT NULL | Khóa ngoại liên kết đến bảng SẢN PHẨM |
| MaBan | INT | FK, NOT NULL | Khóa ngoại liên kết đến bảng BÀN |
| SoLuongMua | INT | NOT NULL | Số lượng sản phẩm đã mua, >= 0 |
| TrangThai | BIT | DEFAULT 0 | Trạng thái thanh toán, 1 là đã thanh toán |
| NgayThanhToan | DATE | NOT NULL | Ngày thanh toán, để lưu lại ngày tháng để sử dụng vào bảng thống kê doanh thu theo tháng và theo năm |

#### Hóa đơn chi tiết

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| MaHoaDonCT | INT | PK, Tự tăng | Mã hóa đơn chi tiết |
| MaHoaDon | INT | FK, NOT NULL | Khóa ngoại liên kết với bảng HÓA ĐƠN |
| SoLuongSP | INT | NOT NULL | Số lượng sản phẩm |

## Thiết kế giao diện

### Sơ đồ tổ chức giao diện

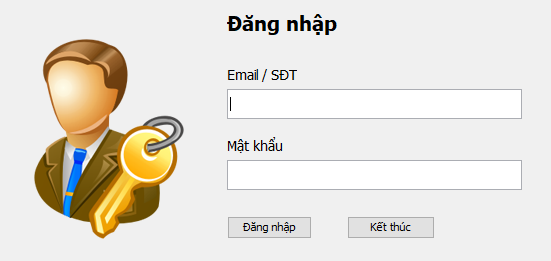


### Thiết kế giao diện cho các chức năng nghiệp vụ

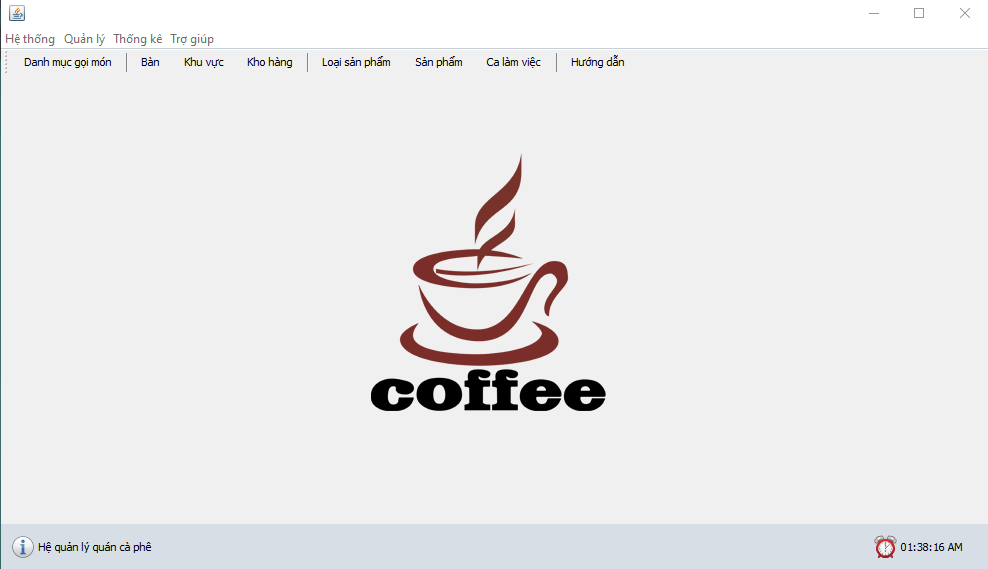
#### Màn hình chào

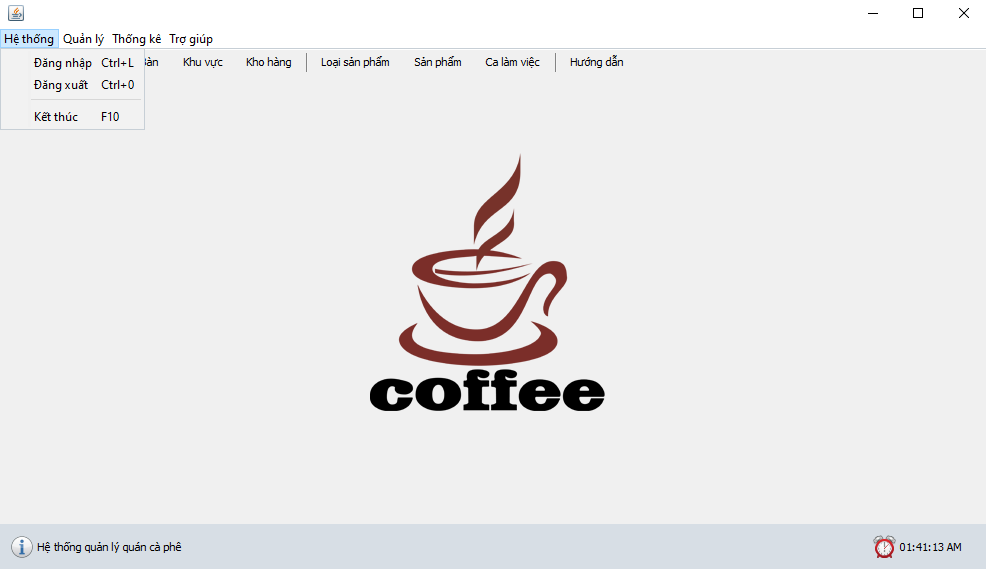


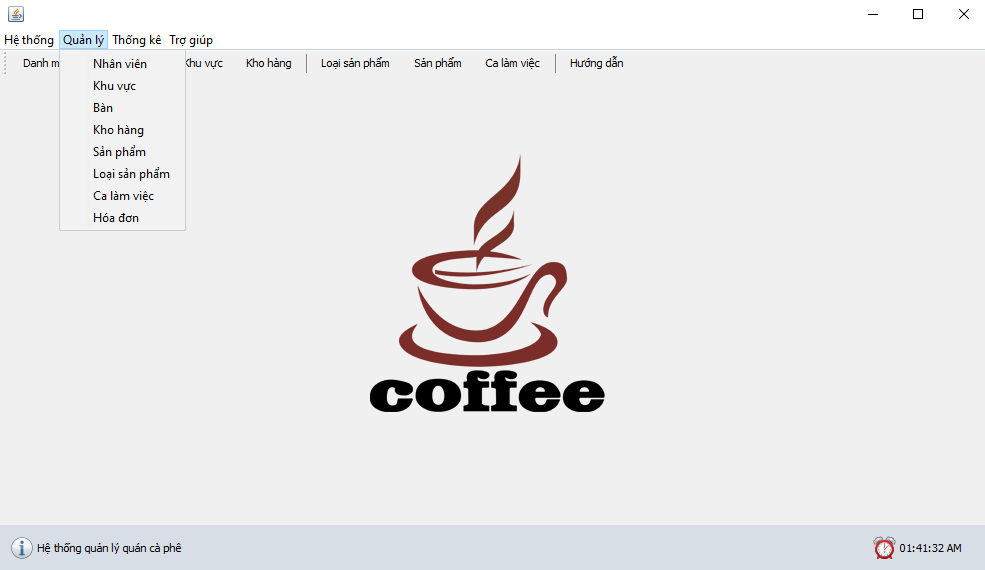
#### Đăng nhâp

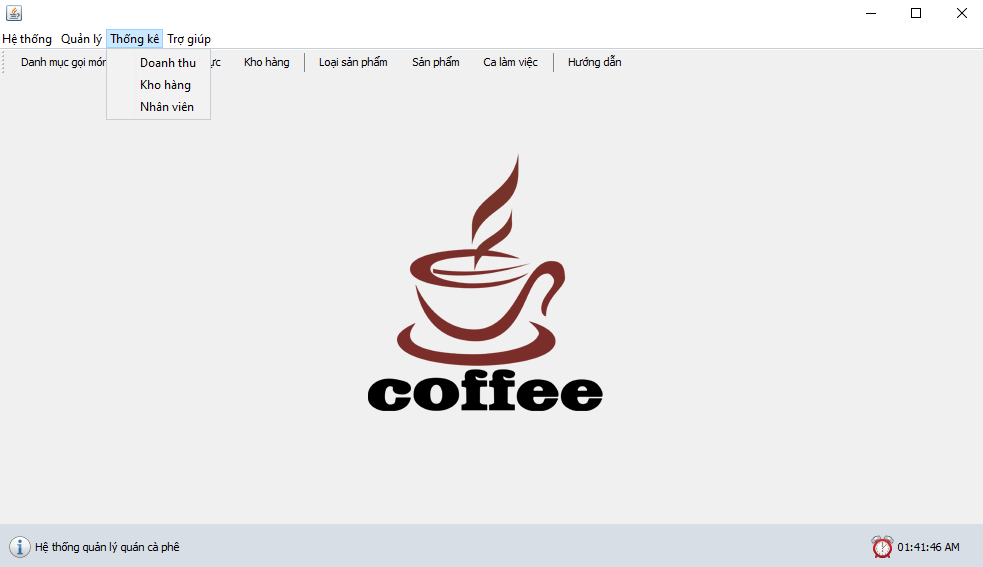


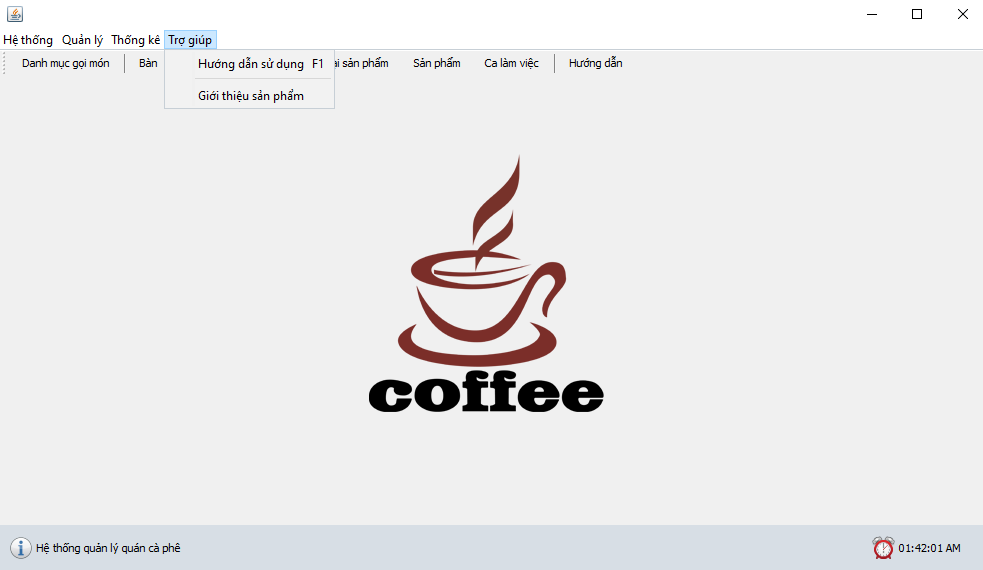
#### Giao diện chính

****

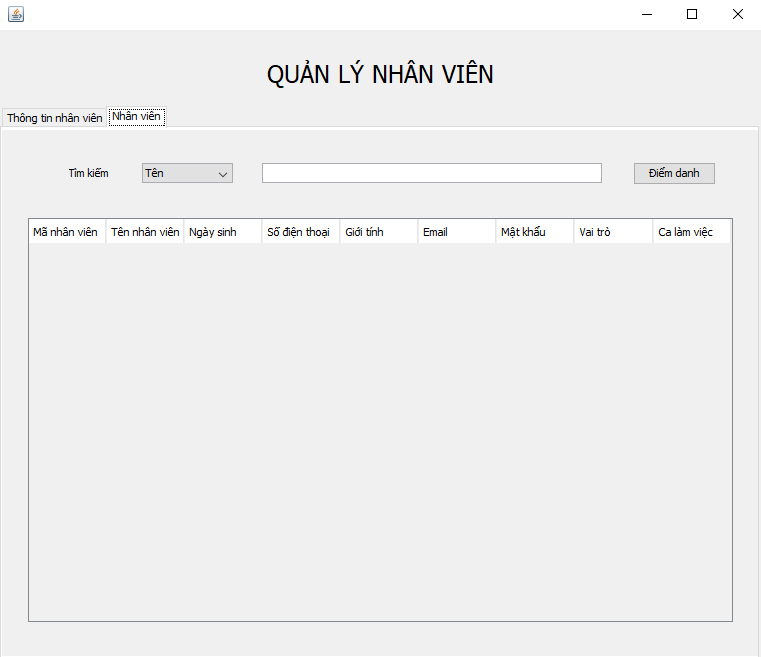
****

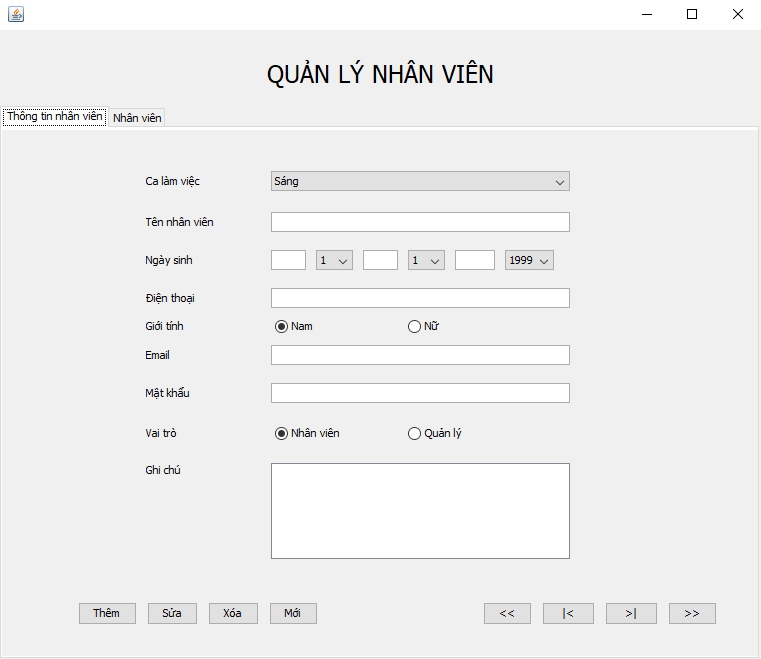
****

****

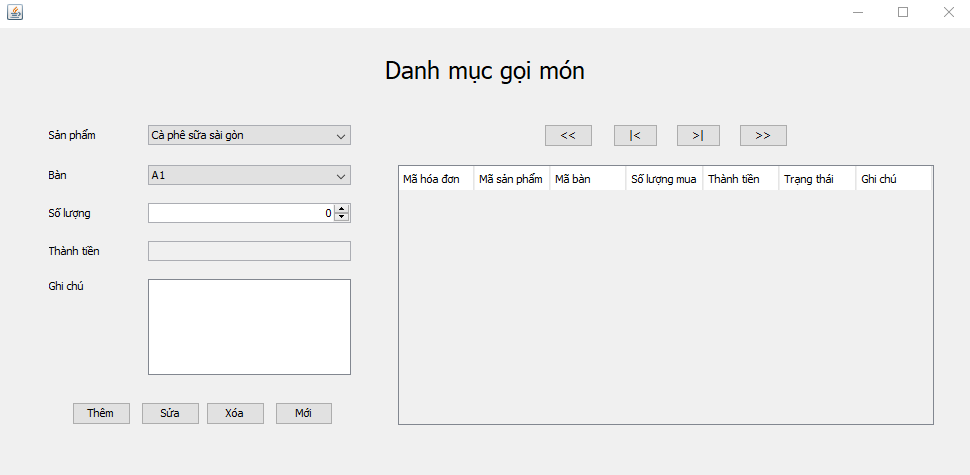
****

#### Quản lý nhân viên

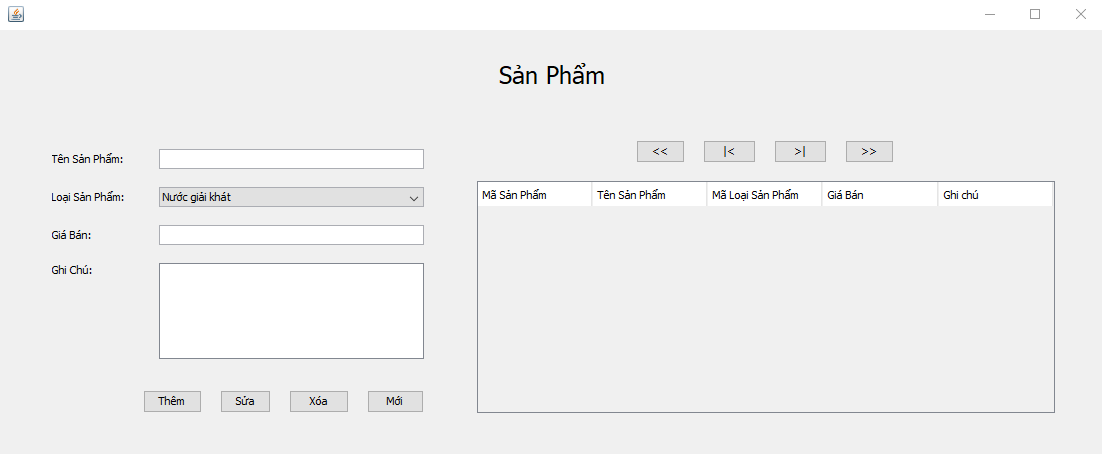




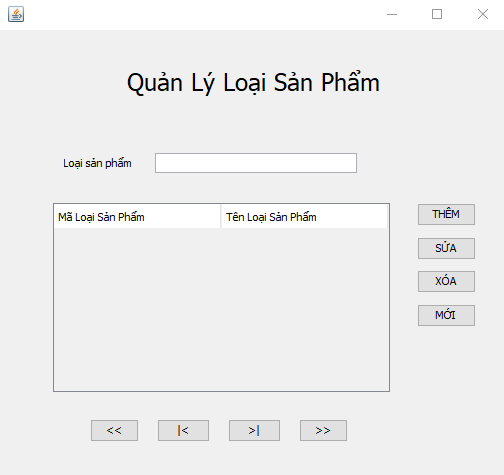
#### Quản lý gọi món



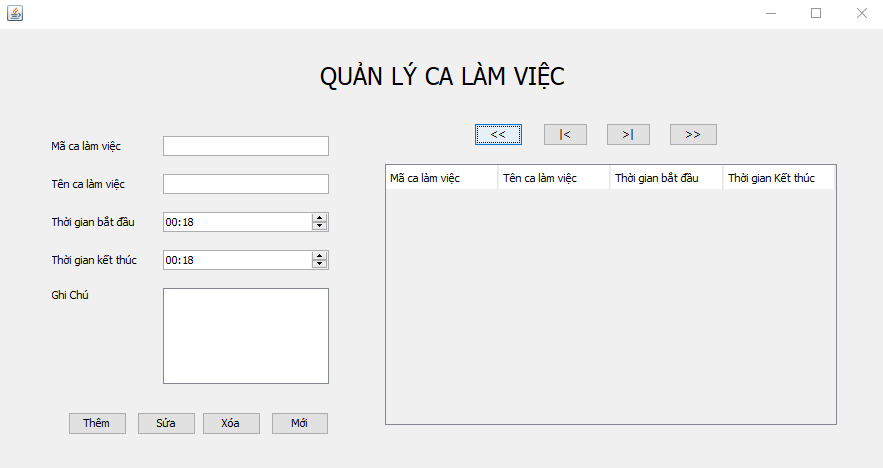
#### Quảm lý sản phẩm



#### Quản lý loại sản phẩm



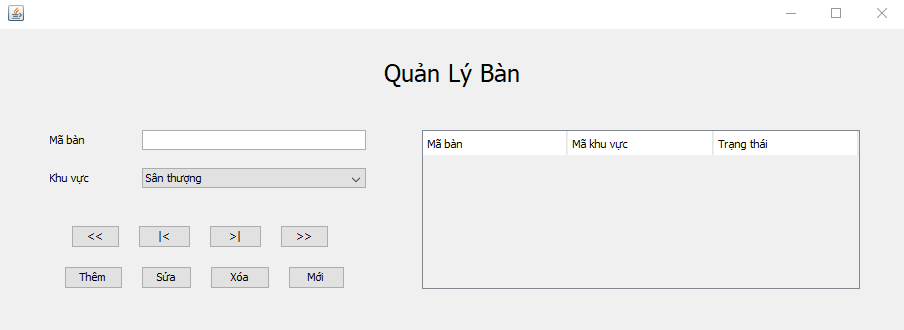
#### Quản lý ca làm việc



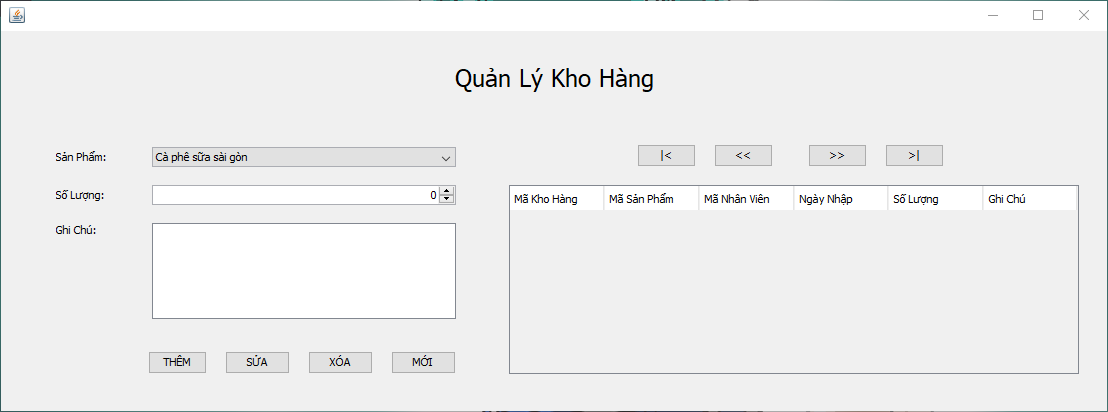
#### Quản lý khu vực



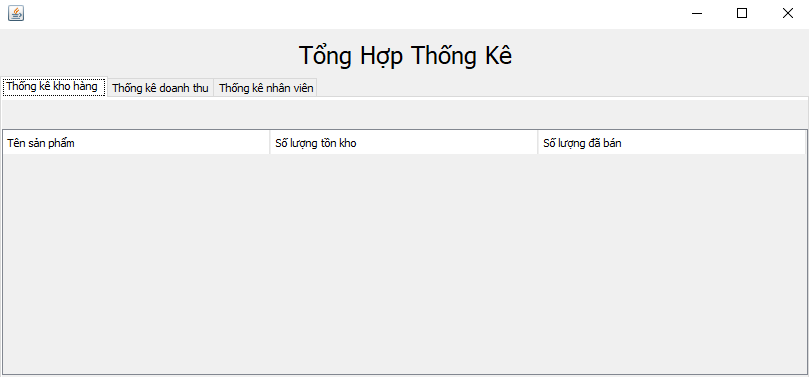
#### Quản lý bàn

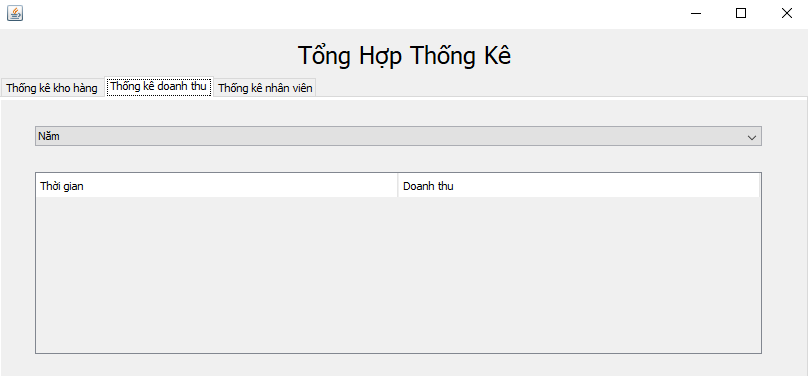


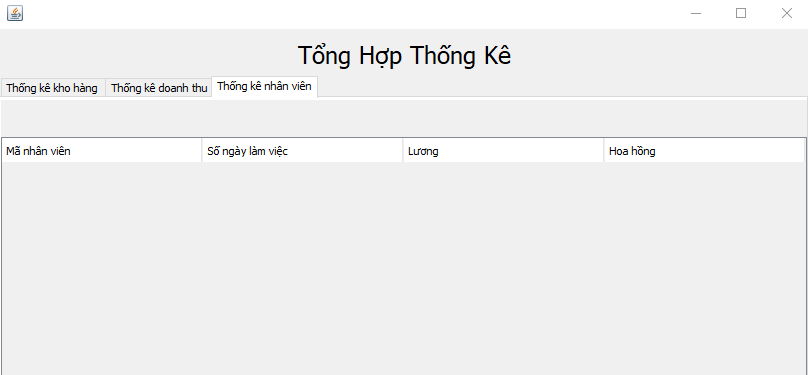
#### Quản lý kho hàng



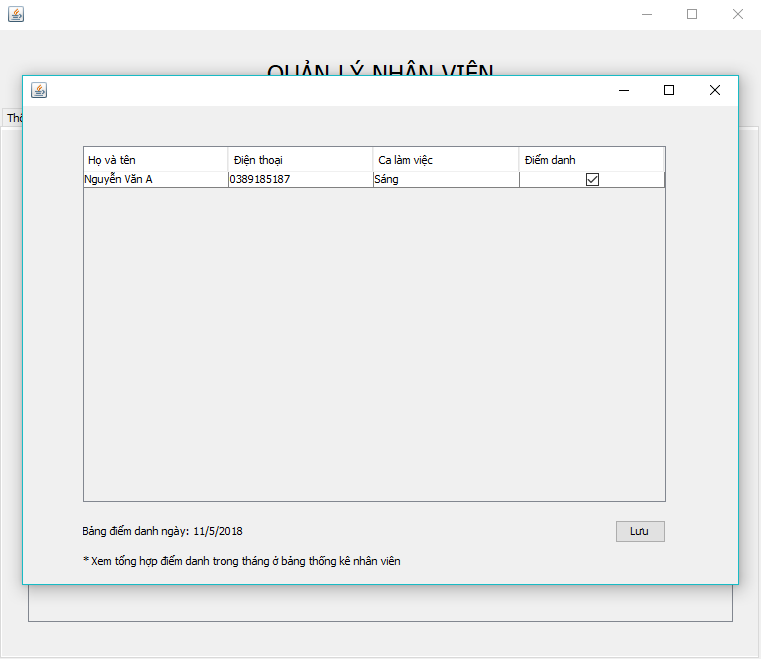
#### Tổng hợp thống kê



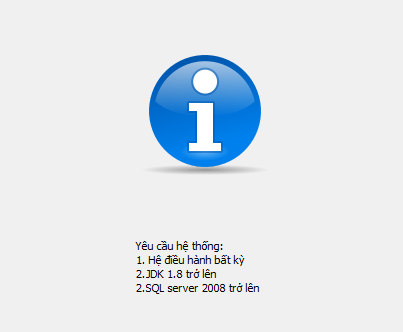




#### Điểm danh nhân viên



#### Thông báo



# Thực hiện viết mã

## Viết mã tạo CSDL

### Tạo CSDL

|  |
| --- |
| DROP DATABASE QlCoffee  GO  USE QlCoffee  GO  CREATE TABLE KhuVuc(  MaKhuVuc INT IDENTITY(1,1) PRIMARY KEY,  TenKhuVuc NVARCHAR(50) NOT NULL UNIQUE,  ViTri NVARCHAR(50) NOT NULL,  GhiChu NVARCHAR(255) NULL  )  GO  CREATE TABLE Ban(  MaBan INT PRIMARY KEY,  MaKhuVuc INT NOT NULL REFERENCES KhuVuc(MaKhuVuc),  TrangThai BIT DEFAULT 0  )  GO  CREATE TABLE LoaiSanPham(  MaLoaiSP INT IDENTITY(1,1) PRIMARY KEY,  TenLoaiSP NVARCHAR(50) NOT NULL UNIQUE  )  GO  CREATE TABLE SanPham(  MaSanPham INT IDENTITY(1,1) PRIMARY KEY,  TenSanPham NVARCHAR(50) NOT NULL,  MaLoaiSP INT NOT NULL REFERENCES LoaiSanPham(MaLoaiSP),  GiaBan FLOAT NOT NULL, CHECK(GiaBan>=0),  TrangThai BIT DEFAULT 0,  GhiChu NVARCHAR(255) NOT NULL  )  GO  CREATE TABLE CaLamViec(  MaCaLV INT IDENTITY(1,1) PRIMARY KEY,  BatDau NVARCHAR(50) NOT NULL,  KetThuc NVARCHAR(50) NOT NULL,  GhiChu NVARCHAR(255) NOT NULL,  TenCaLV NVARCHAR(50) NOT NULL UNIQUE  )  GO  CREATE TABLE NhanVien(  MaNhanVien INT IDENTITY(1,1) PRIMARY KEY,  HoTen NVARCHAR(50) NOT NULL,  MatKhau NVARCHAR(50) NOT NULL,  VaiTro BIT DEFAULT 0,  NgaySinh DATE NOT NULL,  GioiTinh BIT DEFAULT 0,  Email NVARCHAR(50) NOT NULL UNIQUE,  DienThoai NVARCHAR(50) NOT NULL UNIQUE,  GhiChu NVARCHAR(50) NULL,  MaCaLV INT REFERENCES CaLamViec(MaCaLV),  SoNgayLV INT DEFAULT 0  )  GO  CREATE TABLE HoaDon(  MaHoaDon INT IDENTITY(1,1) PRIMARY KEY,  MaNhanVien INT REFERENCES NhanVien(MaNhanVien),  MaBan INT REFERENCES Ban(MaBan),  GhiChu NVARCHAR(255) NOT NULL,  TrangThai BIT DEFAULT 0,  NgayThanhToan DATE  )  GO  CREATE TABLE KhoHang(  MaKhoHang INT IDENTITY(1,1) PRIMARY KEY,  MaSanPham INT REFERENCES SanPham(MaSanPham),  NgayNhap DATE NOT NULL,  MaNhanVien INT REFERENCES NhanVien(MaNhanVien),  SoLuong INT DEFAULT 0, CHECK(SoLuong>=0),  GhiChu NVARCHAR(255) NULL,  HanSuDung DATE,  )  GO  CREATE TABLE HoaDonChiTiet(  MaHoaDonCT INT IDENTITY(1,1) PRIMARY KEY,  MaHoaDon INT REFERENCES HoaDon(MaHoaDon),  SoLuongSP INT NOT NULL, CHECK(SoLuongSP>=0),  MaSanPham INT REFERENCES SanPham(MaSanPham)  ) |

### SQL truy vấn và thao tác

#### KhuVuc

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

#### Ban

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

#### LoaiSanPham

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

#### SanPham

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

#### CaLamViec

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

#### NhanVien

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

#### KhoHang

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

#### HoaDon

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

#### HoaDonChiTiet

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới |  |
| Cập nhật theo mã |  |
| Xóa theo mã |  |
| Truy vấn tất cả |  |
| Truy vấn theo mã |  |

### Các thủ tục lưu tổng hợp thống kê

#### Thống kê kho hàng

|  |
| --- |
| MaSanPham,  SUM(SoLuongSP)  FROM HoaDonChiTiet  GROUP BY MaSanPham  END  exec sp\_ThongKeKhoHang |

#### Thống kê doanh thu

|  |
| --- |
| CREATE PROC sp\_ThongKeDoanhThu  AS BEGIN  SELECT  YEAR(NgayThanhToan),  SUM(ThanhTien)  from HoaDon  Group by YEAR(NgayThanhToan)  END  exec sp\_ThongKeDoanhThu |

#### Thống kê doanh thu theo tháng

|  |
| --- |
| CREATE PROC sp\_ThongKeDoanhThuTheoThang  AS BEGIN  SELECT  NgayThanhToan,  SUM(ThanhTien)  from HoaDon  Group by NgayThanhToan  END  exec sp\_ThongKeDoanhThuTheoThang |

#### Thống kê nhân viên

|  |
| --- |
| CREATE PROC sp\_ThongKeNhanVien  AS BEGIN  SELECT  HoTen,  SoNgayLV,  VaiTro,  MaCaLV  FROM NhanVien  END |

## Lập trình JDBC

Trong dự án này chúng ta chọn mô hình lập trình với CSDL như hình sau:

DAO

DB

Jbdc

Models

UI

Trong đó:

* UI: các thành phần giao diện
* DB: cơ sở dữ liệu
* Jdbc: là lớp tiện ích cung cấp các hàm tiện ích làm việc với CSDL thông qua câu lệnh sql hoặc lời gọi thủ tục lưu
* Model: là các lớp mô tả dữ liệu theo cấu trúc các bảng trong CSDL
* DAO: là các lớp thao tác và truy vấn dữ liệu. Nó có nhiệm vụ chuyển đổi Model sang SQL và ngược lại.

Theo mô hình này thì lập trình giao diện làm việc với CSDL thông qua các lớp model. Model như là một gói dữ liệu được sử dụng để giao tiếp giữa giao diện và các thành phần DAO. DAO mới thực sự là lập trình CSDL nó làm nhiệm vụ nhận model từ giao diện, chuyển đổi sang SQL và sử dụng Jdbc để thực hiện các thao tác dữ liệu. Ngược lại thì truy vấn dữ liệu từ CSDL, chuyển đổi thành model để hiển thị lên các thành phần giao diện. Việc tách thành các phần riêng biết giúp quản lý dự án tốt hơn, dễ bảo trì nâng cấp hơn.

### Lớp hỗ trợ

Các lớp hỗ trợ chứa các phương thức tiện ích giúp chúng ta thực hiện các xử lý trong ứng dụng đơn giản hơn, ngắn gọn hơn, dễ hiểu hơn. Trong phần này chúng ta xây dựng 2 lớp là XDate và Jdbc như sau.

#### Lớp tiện ích XDate

Chứa các phương thức chuyển đổi thời gian và chuỗi giúp lập trình sau này khi xử lý thời gian dễ dàng hơn.

|  |
| --- |
| package helper;  import java.text.ParseException;  import java.text.SimpleDateFormat;  import java.util.Calendar;  import java.util.Date;  public class XDate {  static final SimpleDateFormat DATE\_FORMATER = new SimpleDateFormat("dd/MM/yyyy");  // lấy thời gian hiện tại  public static Date now() {  return new Date();  }  // convert String => Date  // date là String cần chuyển  // pattern là định dạng thời gian  public static Date toDate(String date, String... pattern) {  try {  if (pattern.length > 0) {  DATE\_FORMATER.applyPattern(pattern[0]);  }  if (date == null) {  return XDate.now();  }  return DATE\_FORMATER.parse(date);  } catch (ParseException e) {  throw new RuntimeException(e);  }  }  public static String toString(Date date, String... pattern) {  if (pattern.length > 0) {  DATE\_FORMATER.applyPattern(pattern[0]);  }  if (date == null) {  date = XDate.now();  }  return DATE\_FORMATER.format(date);  }  public static Date addDays(Date date, int days) {  Calendar cal = Calendar.getInstance();  cal.setTime(date);  cal.add(Calendar.DATE, days); //minus number would decrement the days  return cal.getTime();  }  } |

#### Lớp tiện ích Jdbc

|  |
| --- |
| package helper;  import java.sql.Connection;  import java.sql.DriverManager;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  public class Jdbc {    private static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";  private static String dburl = "jdbc:sqlserver://localhost:1433;databaseName=QlCoffee";  private static String username = "sa";  private static String password = "123456";  // nạp driver  static {  try {  Class.forName(driver);  } catch (ClassNotFoundException e) {  throw new RuntimeException(e);  }  }  // xây dựng PreparedStatement  // sql là câ lệnh truy vấn sql nó có thể có hoặc không có tham số, nó có thể là 1 thủ tục lưu trữ  public static PreparedStatement preparedStatement(String sql, Object... args) throws SQLException {  Connection connection = DriverManager.getConnection(dburl, username, password);  PreparedStatement statment = null;  if (sql.trim().startsWith("{")) {  statment = connection.prepareCall(sql);  } else {  statment = connection.prepareStatement(sql);  }  for (int i = 0; i < args.length; i++) {  statment.setObject(i + 1, args[i]);  }  return statment;  }  // thực hiện các thao tác sql insert, update, delete hoặc thủ tục truy vấn dữ liệu  // sql là câu lệnh sql có thể chứa tham số. nó có thể là một lời gọi thủ tục lưu trữ  // args là danh sách các giá trị được cung cấp cho các tham số trong câu lệnh sql  public static void executeUpdate(String sql, Object... args) {  try {  PreparedStatement statement = preparedStatement(sql, args);  try {  statement.executeUpdate();  } finally {  statement.getConnection().close();  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  }  // các câu lệnh select hoặc thủ tục truy vấn dữ liệu  // sql là câu lệnh sql có thể chứa các tham số, nó có thể là một lời gọi thủ tục lưu trữ  // args là danh sách các giá trị được cung cấp cho các tham số trong câu lệnh sql  public static ResultSet executeQuery(String sql, Object... args) {  try {  PreparedStatement statement = preparedStatement(sql, args);  return statement.executeQuery();  } catch (SQLException e) {  throw new RuntimeException(e);  }  }  } |

### Model class - Các lớp mô tả dữ liệu

#### KhuVuc

|  |
| --- |
| package model;  public class KhuVuc {  private int maKhuVuc;  private String tenKhuVuc;  private String viTri;  private String ghiChu;  public KhuVuc() {  }  public KhuVuc(int maKhuVuc, String tenKhuVuc, String viTri, String ghichu) {  this.maKhuVuc = maKhuVuc;  this.tenKhuVuc = tenKhuVuc;  this.viTri = viTri;  this.ghiChu = ghichu;  }  public int getMaKhuVuc() {  return maKhuVuc;  }  public void setMaKhuVuc(int maKhuVuc) {  this.maKhuVuc = maKhuVuc;  }  public String getTenKhuVuc() {  return tenKhuVuc;  }  public void setTenKhuVuc(String tenKhuVuc) {  this.tenKhuVuc = tenKhuVuc;  }  public String getViTri() {  return viTri;  }  public void setViTri(String viTri) {  this.viTri = viTri;  }  public String getGhiChu() {  return ghiChu;  }  public void setGhiChu(String ghiChu) {  this.ghiChu = ghiChu;  }  } |

#### Ban

|  |
| --- |
| package model;  public class Ban {  private int maBan;  private int maKhuvuc;  private boolean trangThai = false;  public Ban() {  }  public Ban(int maBan, int maKhuvuc) {  this.maBan = maBan;  this.maKhuvuc = maKhuvuc;  }  public int getMaBan() {  return maBan;  }  public void setMaBan(int maBan) {  this.maBan = maBan;  }  public int getMaKhuvuc() {  return maKhuvuc;  }  public void setMaKhuvuc(int maKhuvuc) {  this.maKhuvuc = maKhuvuc;  }  public boolean isTrangThai() {  return trangThai;  }  public void setTrangThai(boolean trangThai) {  this.trangThai = trangThai;  }  } |

#### LoaiSanPham

|  |
| --- |
| package model;  public class LoaiSanPham {  private int maLoaiSP;  private String tenLoaiSP;  public LoaiSanPham() {  }  public LoaiSanPham(int maLoaiSP, String tenLoaiSP) {  this.maLoaiSP = maLoaiSP;  this.tenLoaiSP = tenLoaiSP;  }  public int getMaLoaiSP() {  return maLoaiSP;  }  public void setMaLoaiSP(int maLoaiSP) {  this.maLoaiSP = maLoaiSP;  }  public String getTenLoaiSP() {  return tenLoaiSP;  }  public void setTenLoaiSP(String tenLoaiSP) {  this.tenLoaiSP = tenLoaiSP;  }  } |

#### SanPham

|  |
| --- |
| package model;  public class SanPham {  private int maSanPham;  private String tenSanPham;  private int maLoaiSP;  private Float giaBan;  private Boolean trangThai;  private String ghiChu;  public SanPham() {  }  public SanPham(int maSanPham, String tenSanPham, int maLoaiSP, Float giaBan, Boolean trangThai, String ghiChu) {  this.maSanPham = maSanPham;  this.tenSanPham = tenSanPham;  this.maLoaiSP = maLoaiSP;  this.giaBan = giaBan;  this.trangThai = trangThai;  this.ghiChu = ghiChu;  }  public int getMaSanPham() {  return maSanPham;  }  public void setMaSanPham(int maSanPham) {  this.maSanPham = maSanPham;  }  public String getTenSanPham() {  return tenSanPham;  }  public void setTenSanPham(String tenSanPham) {  this.tenSanPham = tenSanPham;  }  public int getMaLoaiSP() {  return maLoaiSP;  }  public void setMaLoaiSP(int maLoaiSP) {  this.maLoaiSP = maLoaiSP;  }  public Float getGiaBan() {  return giaBan;  }  public void setGiaBan(Float giaBan) {  this.giaBan = giaBan;  }  public Boolean getTrangThai() {  return trangThai;  }  public void setTrangThai(Boolean trangThai) {  this.trangThai = trangThai;  }  public String getGhiChu() {  return ghiChu;  }  public void setGhiChu(String ghiChu) {  this.ghiChu = ghiChu;  }  } |

#### CaLamViec

|  |
| --- |
| package model;  public class CaLamViec {  private int maCaLamViec;  private String batDau;  private String ketThuc;  private String ghiChu;  private String tenCaLamViec;  public CaLamViec() {  }  public CaLamViec(int maCaLamViec, String batDau, String ketThuc, String ghiChu, String tenCaLamViec) {  this.maCaLamViec = maCaLamViec;  this.batDau = batDau;  this.ketThuc = ketThuc;  this.ghiChu = ghiChu;  this.tenCaLamViec = tenCaLamViec;  }  public int getMaCaLamViec() {  return maCaLamViec;  }  public void setMaCaLamViec(int maCaLamViec) {  this.maCaLamViec = maCaLamViec;  }  public String getBatDau() {  return batDau;  }  public void setBatDau(String batDau) {  this.batDau = batDau;  }  public String getKetThuc() {  return ketThuc;  }  public void setKetThuc(String ketThuc) {  this.ketThuc = ketThuc;  }  public String getGhiChu() {  return ghiChu;  }  public void setGhiChu(String ghiChu) {  this.ghiChu = ghiChu;  }  public String getTenCaLamViec() {  return tenCaLamViec;  }  public void setTenCaLamViec(String tenCaLamViec) {  this.tenCaLamViec = tenCaLamViec;  }  } |

#### NhanVien

|  |
| --- |
| package model;  import java.util.Date;  public class NhanVien {  private int maNhanVien;  private String hoTen;  private String matKhau;  private boolean vaiTro = false;  private Date ngaySinh;  private boolean gioiTinh;  private String email;  private String dienThoai;  private String ghiChu;  private int maCaLamViec;  private int soNgayLamViec = 0;  private CaLamViec caLamViec;  public CaLamViec getCaLamViec() {  return caLamViec;  }  public void setCaLamViec(CaLamViec caLamViec) {  this.caLamViec = caLamViec;  }  public NhanVien(int maNhanVien, String hoTen, String matKhau, Date ngaySinh, boolean gioiTinh, String email, String dienThoai, String ghiChu, int maCaLamViec) {  this.maNhanVien = maNhanVien;  this.hoTen = hoTen;  this.matKhau = matKhau;  this.ngaySinh = ngaySinh;  this.gioiTinh = gioiTinh;  this.email = email;  this.dienThoai = dienThoai;  this.ghiChu = ghiChu;  this.maCaLamViec = maCaLamViec;  }  public NhanVien() {  }  public int getMaNhanVien() {  return maNhanVien;  }  public void setMaNhanVien(int maNhanVien) {  this.maNhanVien = maNhanVien;  }  public String getHoTen() {  return hoTen;  }  public void setHoTen(String hoTen) {  this.hoTen = hoTen;  }  public String getMatKhau() {  return matKhau;  }  public void setMatKhau(String matKhau) {  this.matKhau = matKhau;  }  public boolean isVaiTro() {  return vaiTro;  }  public void setVaiTro(boolean vaiTro) {  this.vaiTro = vaiTro;  }  public Date getNgaySinh() {  return ngaySinh;  }  public void setNgaySinh(Date ngaySinh) {  this.ngaySinh = ngaySinh;  }  public boolean isGioiTinh() {  return gioiTinh;  }  public void setGioiTinh(boolean gioiTinh) {  this.gioiTinh = gioiTinh;  }  public String getEmail() {  return email;  }  public void setEmail(String email) {  this.email = email;  }  public String getDienThoai() {  return dienThoai;  }  public void setDienThoai(String dienThoai) {  this.dienThoai = dienThoai;  }  public String getGhiChu() {  return ghiChu;  }  public void setGhiChu(String ghiChu) {  this.ghiChu = ghiChu;  }  public int getMaCaLamViec() {  return maCaLamViec;  }  public void setMaCaLamViec(int maCaLamViec) {  this.maCaLamViec = maCaLamViec;  }  public int getSoNgayLamViec() {  return soNgayLamViec;  }  public void setSoNgayLamViec(int soNgayLamViec) {  this.soNgayLamViec = soNgayLamViec;  }  } |

#### KhoHang

|  |
| --- |
| package model;  import helper.XDate;  import java.util.Date;  public class KhoHang {  private int maKhoHang;  private int maSanPham;  private Date ngayNhap;  private int maNhanVien;  private int soLuong;  private String ghiChu;  private Date hanSuDung = XDate.now();  public KhoHang() {  }  public KhoHang(int maKhoHang, int maSanPham, Date ngayNhap, int maNhanVien, int soLuong, String ghiChu) {  this.maKhoHang = maKhoHang;  this.maSanPham = maSanPham;  this.ngayNhap = ngayNhap;  this.maNhanVien = maNhanVien;  this.soLuong = soLuong;  this.ghiChu = ghiChu;  }  public int getMaKhoHang() {  return maKhoHang;  }  public void setMaKhoHang(int maKhoHang) {  this.maKhoHang = maKhoHang;  }  public int getMaSanPham() {  return maSanPham;  }  public void setMaSanPham(int maSanPham) {  this.maSanPham = maSanPham;  }  public Date getNgayNhap() {  return ngayNhap;  }  public void setNgayNhap(Date ngayNhap) {  this.ngayNhap = ngayNhap;  }  public int getMaNhanVien() {  return maNhanVien;  }  public void setMaNhanVien(int maNhanVien) {  this.maNhanVien = maNhanVien;  }  public int getSoLuong() {  return soLuong;  }  public void setSoLuong(int soLuong) {  this.soLuong = soLuong;  }  public String getGhiChu() {  return ghiChu;  }  public void setGhiChu(String ghiChu) {  this.ghiChu = ghiChu;  }  public Date getHanSuDung() {  return hanSuDung;  }  public void setHanSuDung(Date hanSuDung) {  this.hanSuDung = hanSuDung;  }  } |

#### HoaDon

|  |
| --- |
| package model;  import java.util.Date;  public class HoaDon {  private int maHoaDon;  private int maNhanVien;  private int maBan;  private String ghiChu;  private boolean trangThai = false;  private Date ngayThanhToan;  private SanPham sanPham;  private HoaDonChiTiet hoaDonChiTiet;  private float thanhTien;  public float getThanhTien() {  return thanhTien;  }  public void setThanhTien(float thanhTien) {  this.thanhTien = thanhTien;  }  public HoaDonChiTiet getHoaDonChiTiet() {  return hoaDonChiTiet;  }  public void setHoaDonChiTiet(HoaDonChiTiet hoaDonChiTiet) {  this.hoaDonChiTiet = hoaDonChiTiet;  }  public SanPham getSanPham() {  return sanPham;  }  public void setSanPham(SanPham sanPham) {  this.sanPham = sanPham;  }  public HoaDon() {  }  public HoaDon(int maHoaDon, int maNhanVien, int maBan, String ghiChu, Date ngayThanhToan) {  this.maHoaDon = maHoaDon;  this.maNhanVien = maNhanVien;  this.maBan = maBan;  this.ghiChu = ghiChu;  this.ngayThanhToan = ngayThanhToan;  }  public int getMaHoaDon() {  return maHoaDon;  }  public void setMaHoaDon(int maHoaDon) {  this.maHoaDon = maHoaDon;  }  public int getMaNhanVien() {  return maNhanVien;  }  public void setMaNhanVien(int maNhanVien) {  this.maNhanVien = maNhanVien;  }  public int getMaBan() {  return maBan;  }  public void setMaBan(int maBan) {  this.maBan = maBan;  }  public String getGhiChu() {  return ghiChu;  }  public void setGhiChu(String ghiChu) {  this.ghiChu = ghiChu;  }  public boolean isTrangThai() {  return trangThai;  }  public void setTrangThai(boolean trangThai) {  this.trangThai = trangThai;  }  public Date getNgayThanhToan() {  return ngayThanhToan;  }  public void setNgayThanhToan(Date ngayThanhToan) {  this.ngayThanhToan = ngayThanhToan;  }  } |

#### HoaDonChiTiet

|  |
| --- |
| package model;  public class HoaDonChiTiet {  private int maHoaDonCT;  private int maHoaDon;  private int soLuongSP;  private int maSanPham;  public HoaDonChiTiet() {  }  public HoaDonChiTiet(int maHoaDonCT, int maHoaDon, int soLuongSP, int maSanPham) {  this.maHoaDonCT = maHoaDonCT;  this.maHoaDon = maHoaDon;  this.soLuongSP = soLuongSP;  this.maSanPham = maSanPham;  }  public int getMaHoaDonCT() {  return maHoaDonCT;  }  public void setMaHoaDonCT(int maHoaDonCT) {  this.maHoaDonCT = maHoaDonCT;  }  public int getMaHoaDon() {  return maHoaDon;  }  public void setMaHoaDon(int maHoaDon) {  this.maHoaDon = maHoaDon;  }  public int getSoLuongSP() {  return soLuongSP;  }  public void setSoLuongSP(int soLuongSP) {  this.soLuongSP = soLuongSP;  }  public int getMaSanPham() {  return maSanPham;  }  public void setMaSanPham(int maSanPham) {  this.maSanPham = maSanPham;  }  } |

### DAO Class - Các lớp truy xuất dữ liệu

DAO (Data Access Object) là các lớp giúp giúp làm việc với các bảng trong CSDL. Mỗi lớp được định nghĩa một cách đầy đủ nhất về các thao tác và truy vấn dữ liệu sẽ được sử dụng trong ứng dụng. DAO sau đó sẽ được sử dụng trong các điều khiển sự kiện của giao diện để làm việc với CSDL một cách thuận tiện.

#### KhuVucDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import model.KhuVuc;  public class KhuVucDAO {  public void insert(KhuVuc model) {  String sql = "INSERT INTO KhuVuc(TenKhuVuc, ViTri, GhiChu) VALUES (?,?,?)";  Jdbc.executeUpdate(sql,  model.getTenKhuVuc(),  model.getViTri(),  model.getGhiChu());  }  public void update(KhuVuc model) {  String sql = "UPDATE KhuVuc SET TenKhuVuc=?, ViTri=?, GhiChu=? WHERE MaKhuVuc=?";  Jdbc.executeUpdate(sql,  model.getTenKhuVuc(),  model.getViTri(),  model.getGhiChu(),  model.getMaKhuVuc());  }  public void delete(Integer maKhuVuc) {  String sql = "DELETE FROM KhuVuc WHERE MaKhuVuc=?";  Jdbc.executeUpdate(sql, maKhuVuc);  }  private KhuVuc readFromResultSet(ResultSet rs) throws SQLException {  KhuVuc model = new KhuVuc();  model.setMaKhuVuc(rs.getInt(1));  model.setTenKhuVuc(rs.getString(2));  model.setViTri(rs.getString(3));  model.setGhiChu(rs.getString(4));  return model;  }  private List<KhuVuc> select(String sql, Object... args) {  List<KhuVuc> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, args);  while (rs.next()) {  KhuVuc model = readFromResultSet(rs);  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  }  public List<KhuVuc> select() {  String sql = "SELECT \* FROM KhuVuc";  return select(sql);  }  public KhuVuc findById(int maKhuVuc) {  String sql = "SELECT \* FROM KhuVuc WHERE MaKhuVuc=?";  List<KhuVuc> list = select(sql, maKhuVuc);  return list.size() > 0 ? list.get(0) : null;  }  public KhuVuc findByName(String tenBan) {  String sql = "SELECT \* FROM KhuVuc WHERE TenKhuVuc=?";  List<KhuVuc> list = select(sql, tenBan);  return list.size() > 0 ? list.get(0) : null;  }  } |

#### BanDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import model.Ban;  public class BanDAO {  public void insert(Ban model) {  String sql = "INSERT INTO Ban(MaBan,MaKhuVuc) VALUES(?,?)";  Jdbc.executeUpdate(sql,  model.getMaBan(),  model.getMaKhuvuc()  );  }  public void update(Ban model) {  String sql = "UPDATE Ban SET MaKhuVuc=?, TrangThai=? WHERE MaBan=?";  Jdbc.executeUpdate(sql,  model.getMaKhuvuc(),  model.isTrangThai(),  model.getMaBan()  );  }  public void datBan(int trangThai, int maBan) {  String sql = "UPDATE Ban SET TrangThai=? WHERE MaBan=?";  Jdbc.executeUpdate(sql, trangThai, maBan);  }  public void delete(Integer maBan) {  String sql = "DELETE FROM Ban WHERE MaBan=?";  Jdbc.executeUpdate(sql, maBan);  }  private Ban readFromResultSet(ResultSet rs) throws SQLException {  Ban model = new Ban();  model.setMaBan(rs.getInt(1));  model.setMaKhuvuc(rs.getInt(2));  model.setTrangThai(rs.getBoolean(3));  return model;  }  private List<Ban> select(String sql, Object... arg) {  List<Ban> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, arg);  while (rs.next()) {  Ban model = readFromResultSet(rs);  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  }  public List<Ban> select() {  String Sql = "SELECT \* FROM Ban";  return select(Sql);  }  public Ban findById(Integer maBan) {  String sql = "SELECT \* FROM Ban WHERE MaBan=?";  List<Ban> list = select(sql, maBan);  return list.size() > 0 ? list.get(0) : null;  }  public boolean check(String maBan) {  String sql = "SELECT \* FROM Ban WHERE MaBan LIKE ?";  List<Ban> list = select(sql, maBan);  return list.size() == 0;  }  } |

#### LoaiSanPhamDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import model.LoaiSanPham;  public class LoaiSanPhamDAO {  public void insert(LoaiSanPham model) {  String sql = "INSERT INTO LoaiSanPham (TenLoaiSp) VALUES(?)";  Jdbc.executeUpdate(sql,  model.getTenLoaiSP());  }  public void update(LoaiSanPham model) {  String sql = "UPDATE LoaiSanPham SET TenLoaiSP=? WHERE MaLoaiSP=?";  Jdbc.executeUpdate(sql,  model.getTenLoaiSP(),  model.getMaLoaiSP());  }  public void delete(int maLoaiSP) {  String sql = "DELETE FROM LoaiSanPham WHERE MaLoaiSP=?";  Jdbc.executeUpdate(sql,maLoaiSP);  }  private LoaiSanPham readFormResultSet(ResultSet rs) throws SQLException {  LoaiSanPham loaisanpham = new LoaiSanPham();  loaisanpham.setMaLoaiSP(rs.getInt(1));  loaisanpham.setTenLoaiSP(rs.getString(2));  return loaisanpham;  }  public List<LoaiSanPham> select(String sql, Object... args) {  List<LoaiSanPham> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, args);  while (rs.next()) {  LoaiSanPham loaiSanPham = readFormResultSet(rs);  list.add(loaiSanPham);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  }  public List<LoaiSanPham> select() {  String sql = "SELECT \* FROM LoaiSanPham";  return select(sql);  }  public LoaiSanPham findById(int maLoaiSP) {  String sql = "SELECT \* FROM LoaiSanPham WHERE MaLoaiSP=?";  List<LoaiSanPham> list = select(sql, maLoaiSP);  return list.size() > 0 ? list.get(0) : null;  }  public LoaiSanPham findByName(String tenLoaiSP) {  String sql = "SELECT \* FROM LoaiSanPham WHERE TenLoaiSP=?";  List<LoaiSanPham> list = select(sql, tenLoaiSP);  return list.size() > 0 ? list.get(0) : null;  }  } |

#### SanPhamDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import model.SanPham;  public class SanPhamDAO {  public void insert(SanPham model) {  String sql = "INSERT INTO SanPham (TenSanPham, MaLoaiSP, GiaBan, TrangThai, GhiChu) VALUES (?,?,?,?,?)";  Jdbc.executeUpdate(sql,  model.getTenSanPham(),  model.getMaLoaiSP(),  model.getGiaBan(),  model.getTrangThai(),  model.getGhiChu());  }  public void update(SanPham model) {  String sql = "UPDATE SanPham SET TenSanPham=?, MaLoaiSP=?, GiaBan=?, GhiChu=? WHERE MaSanPham=?";  Jdbc.executeUpdate(sql,  model.getTenSanPham(),  model.getMaLoaiSP(),  model.getGiaBan(),  model.getGhiChu(),  model.getMaSanPham());  }  public void delete(int maSanPham) {  String sql = "DELETE FROM SanPham WHERE MaSanPham=?";  Jdbc.executeUpdate(sql, maSanPham);  }  private SanPham readFormResultSet(ResultSet rs) throws SQLException {  SanPham sanpham = new SanPham();  sanpham.setMaSanPham(rs.getInt(1));  sanpham.setTenSanPham(rs.getString(2));  sanpham.setMaLoaiSP(rs.getInt(3));  sanpham.setGiaBan(rs.getFloat(4));  sanpham.setTrangThai(rs.getBoolean(5));  sanpham.setGhiChu(rs.getString(6));  return sanpham;  }  public List<SanPham> select(String sql, Object... args) {  List<SanPham> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, args);  while (rs.next()) {  SanPham sanpham = readFormResultSet(rs);  list.add(sanpham);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  }  public List<SanPham> select() {  String sql = "SELECT \* FROM SanPham";  return select(sql);  }  public SanPham findById(int maSanPham) {  String sql = "SELECT \* FROM SanPham WHERE MaSanPham=?";  List<SanPham> list = select(sql, maSanPham);  return list.size() > 0 ? list.get(0) : null;  }  public SanPham findByName(String sanpham) {  String sql = "SELECT \* FROM SanPham WHERE TenSanPham like ?";  List<SanPham> list = select(sql, sanpham);  return list.size() > 0 ? list.get(0) : null;  }  public List<SanPham> selectByKeyword(String keyword) {  String sql = "SELECT \* FROM SanPham WHERE TenSanPham LIKE ?";  return select(sql, "%" + keyword + "%");  }  } |

#### CaLamViecDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import model.CaLamViec;  public class CaLamViecDAO {  public void insert(CaLamViec model) {  String sql = "INSERT INTO CaLamViec (BatDau, KetThuc, GhiChu,TenCaLV) VALUES (?,?,?,?)";  Jdbc.executeUpdate(sql,  model.getBatDau(),  model.getKetThuc(),  model.getGhiChu(),  model.getTenCaLamViec());  }  public void update(CaLamViec model) {  String sql = "UPDATE CaLamViec SET BatDau=?, KetThuc=?, GhiChu=?, TenCaLV=? WHERE MaCaLV=?";  Jdbc.executeUpdate(sql,  model.getBatDau(),  model.getKetThuc(),  model.getGhiChu(),  model.getTenCaLamViec(),  model.getMaCaLamViec());  }  public void delete(Integer maCaLamViec) {  String sql = "DELETE FROM CaLamViec WHERE MaCaLV=?";  Jdbc.executeUpdate(sql, maCaLamViec);  }    private CaLamViec readFromResultSet(ResultSet rs) throws SQLException {  CaLamViec model = new CaLamViec();  model.setMaCaLamViec(rs.getInt(1));  model.setBatDau(rs.getString(2));  model.setKetThuc(rs.getString(3));  model.setGhiChu(rs.getString(4));  model.setTenCaLamViec(rs.getString(5));  return model;  }  private List<CaLamViec> select(String sql, Object... args) {  List<CaLamViec> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, args);  while (rs.next()) {  CaLamViec model = readFromResultSet(rs);  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  }  public List<CaLamViec> select() {  String sql = "SELECT \* FROM CaLamviec";  return select(sql);  }  public CaLamViec findById(int maCaLV) {  String sql = "SELECT \* FROM CaLamviec WHERE MaCaLV=?";  List<CaLamViec> list = select(sql, maCaLV);  return list.size() > 0 ? list.get(0) : null;  }  public CaLamViec findByName(String tenCaLV) {  String sql = "SELECT \* FROM CaLamviec WHERE TenCaLV like ?";  List<CaLamViec> list = select(sql, tenCaLV);  return list.size() > 0 ? list.get(0) : null;  }  } |

#### NhanVien

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import model.CaLamViec;  import model.NhanVien;  public class NhanVienDAO {  public void insert(NhanVien model) {  String sql = "INSERT INTO NhanVien(HoTen, MatKhau, VaiTro, NgaySinh, GioiTinh, Email, DienThoai, GhiChu, MaCaLV, SoNgayLV) VALUES (?,?,?,?,?,?,?,?,?,?)";  Jdbc.executeUpdate(sql,  model.getHoTen(),  model.getMatKhau(),  model.isVaiTro(),  model.getNgaySinh(),  model.isGioiTinh(),  model.getEmail(),  model.getDienThoai(),  model.getGhiChu(),  model.getMaCaLamViec(),  model.getSoNgayLamViec()  );  }  public void update(NhanVien model) {  String sql = "UPDATE NhanVien SET HoTen=?, MatKhau=?, VaiTro=?, NgaySinh=?, GioiTinh=?, Email=?, DienThoai=?, GhiChu=?, MaCaLV=?, SoNgayLV=? WHERE MaNhanVien=?";  Jdbc.executeUpdate(sql,  model.getHoTen(),  model.getMatKhau(),  model.isVaiTro(),  model.getNgaySinh(),  model.isGioiTinh(),  model.getEmail(),  model.getDienThoai(),  model.getGhiChu(),  model.getMaCaLamViec(),  model.getSoNgayLamViec(),  model.getMaNhanVien()  );  }  public void diemDanh(NhanVien model) {  String sql = "UPDATE NhanVien SET SoNgayLV = ? WHERE MaNhanVien = ?";  Jdbc.executeUpdate(sql,  model.getSoNgayLamViec(),  model.getMaNhanVien()  );  }  public void delete(int maNhanVien) {  String sql = "DELETE FROM NhanVien WHERE MaNhanVien=?";  Jdbc.executeUpdate(sql, maNhanVien);  }  private NhanVien readFormResultSet(ResultSet rs) throws SQLException {  NhanVien nhanVien = new NhanVien();  nhanVien.setMaNhanVien(rs.getInt(1));  nhanVien.setHoTen(rs.getString(2));  nhanVien.setMatKhau(rs.getString(3));  nhanVien.setVaiTro(rs.getBoolean(4));  nhanVien.setNgaySinh(rs.getDate(5));  nhanVien.setGioiTinh(rs.getBoolean(6));  nhanVien.setEmail(rs.getString(7));  nhanVien.setDienThoai(rs.getString(8));  nhanVien.setGhiChu(rs.getString(9));  nhanVien.setMaCaLamViec(rs.getInt(10));  nhanVien.setSoNgayLamViec(rs.getInt(11));  CaLamViec caLamViec = new CaLamViec();  caLamViec.setMaCaLamViec(rs.getInt(12));  caLamViec.setBatDau(rs.getString(13));  caLamViec.setKetThuc(rs.getString(14));  caLamViec.setGhiChu(rs.getString(15));  caLamViec.setTenCaLamViec(rs.getString(16));  nhanVien.setCaLamViec(caLamViec);  return nhanVien;  }  public List<NhanVien> select(String sql, Object... args) {  List<NhanVien> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, args);  while (rs.next()) {  NhanVien nhanVien = readFormResultSet(rs);  list.add(nhanVien);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  }  public List<NhanVien> select() {  String sql = "SELECT \* from NhanVien join CaLamViec on NhanVien.MaCaLV = CaLamViec.MaCaLV";  return select(sql);  }  public NhanVien findById(int maNhanVien) {  String sql = "SELECT \* from NhanVien join CaLamViec on NhanVien.MaCaLV = CaLamViec.MaCaLV WHERE NhanVien.MaNhanVien = ?";  List<NhanVien> list = select(sql, maNhanVien);  return list.size() > 0 ? list.get(0) : null;  }  public NhanVien findByName(String hoTen) {  String sql = "SELECT \* from NhanVien join CaLamViec on NhanVien.MaCaLV = CaLamViec.MaCaLV WHERE NhanVien.HoTen like ?";  List<NhanVien> list = select(sql, "%" + hoTen + "%");  return list.size() > 0 ? list.get(0) : null;  }  public NhanVien findByEmail(String email) {  String sql = "SELECT \* FROM NhanVien Where Email like ?";  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, email);  while (rs.next()) {  NhanVien nhanVien = new NhanVien();  nhanVien.setMaNhanVien(rs.getInt(1));  nhanVien.setHoTen(rs.getString(2));  nhanVien.setMatKhau(rs.getString(3));  nhanVien.setVaiTro(rs.getBoolean(4));  nhanVien.setNgaySinh(rs.getDate(5));  nhanVien.setGioiTinh(rs.getBoolean(6));  nhanVien.setEmail(rs.getString(7));  nhanVien.setDienThoai(rs.getString(8));  nhanVien.setGhiChu(rs.getString(9));  nhanVien.setMaCaLamViec(rs.getInt(10));  nhanVien.setSoNgayLamViec(rs.getInt(11));  return nhanVien;  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return null;  }  } |

#### KhoHangDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import model.KhoHang;  public class KhoHangDAO {  public void insert(KhoHang model) {  String sql = "INSERT INTO KhoHang( MaSanPham, NgayNhap, MaNhanVien, SoLuong, GhiChu, HanSuDung) VALUES (?,?,?,?,?,?)";  Jdbc.executeUpdate(sql,  model.getMaSanPham(),  model.getNgayNhap(),  model.getMaNhanVien(),  model.getSoLuong(),  model.getGhiChu(),  model.getHanSuDung());  }  public void update(KhoHang model) {  String sql = "UPDATE KhoHang SET MaSanPham=?, SoLuong=?, GhiChu=? WHERE MaKhoHang=?";  Jdbc.executeUpdate(sql,  model.getMaSanPham(),  model.getSoLuong(),  model.getGhiChu(),  model.getMaKhoHang());  }  public void delete(int MaKhoHang) {  String sql = "DELETE FROM KhoHang WHERE MaKhoHang=?";  Jdbc.executeUpdate(sql, MaKhoHang);  }  private KhoHang readFormResultSet(ResultSet rs) throws SQLException {  KhoHang khohang = new KhoHang();  khohang.setMaKhoHang(rs.getInt(1));  khohang.setMaSanPham(rs.getInt(2));  khohang.setNgayNhap(rs.getDate(3));  khohang.setMaNhanVien(rs.getInt(4));  khohang.setSoLuong(rs.getInt(5));  khohang.setGhiChu(rs.getString(6));  khohang.setHanSuDung(rs.getDate(7));  return khohang;  }  public List<KhoHang> select(String sql, Object... args) {  List<KhoHang> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, args);  while (rs.next()) {  KhoHang khohang = readFormResultSet(rs);  list.add(khohang);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  throw new RuntimeException(e);  }  return list;  }  public List<KhoHang> select() {  String sql = "SELECT \* FROM KhoHang ";  return select(sql);  }  public KhoHang findById(int maKhoHang) {  String sql = "SELECT \* FROM KhoHang WHERE MaKhoHang=?";  List<KhoHang> list = select(sql, maKhoHang);  return list.size() > 0 ? list.get(0) : null;  }  public KhoHang findByName(String khohang) {  String sql = "SELECT \* FROM KhoHang WHERE MaKhoHang like ?";  List<KhoHang> list = select(sql, khohang);  return list.size() > 0 ? list.get(0) : null;  }  public int getSLByMaSP(int maSP) {  try {  ResultSet rs = null;  String sql = "SELECT SUM(SoLuong) FROM KhoHang WHERE MaSanPham = ? group by MaSanPham";  try {  rs = Jdbc.executeQuery(sql, maSP);  while (rs.next()) {  return rs.getInt(1);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return 0;  }  public int getMaKhoHangByMaSP(int maSanPham) {  try {  ResultSet rs = null;  String sql = "SELECT MaKhoHang FROM KhoHang WHERE MaSanPham = ? and SoLuong > 0 ORDER BY NgayNhap ASC";  try {  rs = Jdbc.executeQuery(sql, maSanPham);  while (rs.next()) {  return rs.getInt(1);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return 0;  }    public int getMaSLHang(int maKhoHang) {  try {  ResultSet rs = null;  String sql = "SELECT SoLuong FROM KhoHang WHERE MaKhoHang = ?";  try {  rs = Jdbc.executeQuery(sql, maKhoHang);  while (rs.next()) {  return rs.getInt(1);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return 0;  }    public void updateSLByMaSP(int soLuong, int maSP, int maKhoHang) {  String sql = "update KhoHang set SoLuong = SoLuong - ? where MaSanPham = ? and MaKhoHang = ?";  Jdbc.executeUpdate(sql, soLuong, maSP, maKhoHang);  }  public void themSLByMaSP(int soLuong, int maSP, int maKhoHang) {  String sql = "update KhoHang set SoLuong = SoLuong + ? where MaSanPham = ? and MaKhoHang = ?";  Jdbc.executeUpdate(sql, soLuong, maSP, maKhoHang);  }  } |

#### HoaDonDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.Date;  import java.util.List;  import model.HoaDon;  import model.HoaDonChiTiet;  import model.SanPham;  public class HoaDonDAO {  public List<HoaDon> getAll() {  List<HoaDon> list = new ArrayList<>();  try {  ResultSet rs = null;  String sql = "SELECT \* from HoaDon JOIN HoaDonChiTiet "  + "ON HoaDon.MaHoaDon = HoaDonChiTiet.MaHoaDon "  + "JOIN SanPham on HoaDonChiTiet.MaSanPham = SanPham.MaSanPham";  try {  rs = Jdbc.executeQuery(sql);  while (rs.next()) {  HoaDon hoaDon = new HoaDon();  hoaDon.setMaHoaDon(rs.getInt(1));  hoaDon.setMaNhanVien(rs.getInt(2));  hoaDon.setMaBan(rs.getInt(3));  hoaDon.setGhiChu(rs.getString(4));  hoaDon.setTrangThai(rs.getBoolean(5));  hoaDon.setNgayThanhToan(rs.getDate(6));  hoaDon.setThanhTien(rs.getFloat(7));  HoaDonChiTiet hdct = new HoaDonChiTiet(rs.getInt(8), rs.getInt(9), rs.getInt(10), rs.getInt(11));  hoaDon.setHoaDonChiTiet(hdct);  SanPham sanPham = new SanPham(rs.getInt(12), rs.getString(13), rs.getInt(14), rs.getFloat(15), rs.getBoolean(16), rs.getString(17));  hoaDon.setSanPham(sanPham);  list.add(hoaDon);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return list;  }  public HoaDon getAllByMaHD(int maHD) {  List<HoaDon> list = new ArrayList<>();  try {  ResultSet rs = null;  String sql = "SELECT \* from HoaDon JOIN HoaDonChiTiet "  + "ON HoaDon.MaHoaDon = HoaDonChiTiet.MaHoaDon "  + "JOIN SanPham on HoaDonChiTiet.MaSanPham = SanPham.MaSanPham where HoaDon.MaHoaDon = ?";  try {  rs = Jdbc.executeQuery(sql, maHD);  while (rs.next()) {  HoaDon hoaDon = new HoaDon();  hoaDon.setMaHoaDon(rs.getInt(1));  hoaDon.setMaNhanVien(rs.getInt(2));  hoaDon.setMaBan(rs.getInt(3));  hoaDon.setGhiChu(rs.getString(4));  hoaDon.setTrangThai(rs.getBoolean(5));  hoaDon.setNgayThanhToan(rs.getDate(6));  hoaDon.setThanhTien(rs.getFloat(7));  HoaDonChiTiet hdct = new HoaDonChiTiet(rs.getInt(8), rs.getInt(9), rs.getInt(10), rs.getInt(11));  hoaDon.setHoaDonChiTiet(hdct);  SanPham sanPham = new SanPham(rs.getInt(12), rs.getString(13), rs.getInt(14), rs.getFloat(15), rs.getBoolean(16), rs.getString(17));  hoaDon.setSanPham(sanPham);  list.add(hoaDon);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return list.get(0);  }  public int getIDIdentity() {  try {  ResultSet rs = null;  String sql = "SELECT IDENT\_CURRENT('HoaDon')";  try {  rs = Jdbc.executeQuery(sql);  while (rs.next()) {  return rs.getInt(1);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return 0;  }  public static List<Object[]> getHoaDonTheoBan() {  List<Object[]> list = new ArrayList<>();  try {  ResultSet rs = null;  String sql = "SELECT ban.MaBan, KhuVuc.TenKhuVuc from HoaDon JOIN Ban ON HoaDon.MaBan = Ban.MaBan JOIN KhuVuc ON Ban.MaKhuVuc = KhuVuc.MaKhuVuc where ban.TrangThai = 1 and HoaDon.TrangThai = 0";  try {  rs = Jdbc.executeQuery(sql);  while (rs.next()) {  Object[] model = {  rs.getInt(1),  rs.getString(2),};  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return list;  }  public boolean insert(HoaDon model) {  try {  String sql = "INSERT INTO HoaDon (MaNhanVien, MaBan, GhiChu, TrangThai, NgayThanhToan, ThanhTien) VALUES(?,?,?,?,?,?)";  Jdbc.executeUpdate(sql,  model.getMaNhanVien(),  model.getMaBan(),  model.getGhiChu(),  model.isTrangThai(),  model.getNgayThanhToan(),  model.getThanhTien()  );  return true;  } catch (Exception e) {  System.out.println(e.toString());  }  return false;  }  public boolean update(HoaDon model) {  try {  String sql = "UPDATE HoaDon SET MaNhanVien= ?, MaBan= ?, GhiChu= ?, TrangThai= ?, NgayThanhToan= ?, ThanhTien = ? WHERE MaHoaDon= ?";  Jdbc.executeUpdate(sql,  model.getMaNhanVien(),  model.getMaBan(),  model.getGhiChu(),  model.isTrangThai(),  model.getNgayThanhToan(),  model.getThanhTien(),  model.getMaHoaDon()  );  System.out.println(model.getMaHoaDon());  return true;  } catch (Exception e) {  System.out.println(e.toString());  }  return false;  }  public void updateTrangThaiHD(int trangThai, Date ngayThanhToan, int maHD, float ThanhTien) {  String sql = "UPDATE HoaDon SET TrangThai= ?, NgayThanhToan= ?, ThanhTien = ? WHERE MaHoaDon= ?";  Jdbc.executeUpdate(sql, trangThai, ngayThanhToan, ThanhTien, maHD);  }  public void delect(int maHoaDon) {  String sql = "DELETE FROM HoaDon WHERE MaHoaDon= ?";  Jdbc.executeUpdate(sql, maHoaDon);  }  private HoaDon readFormResultSet(ResultSet rs) throws SQLException {  HoaDon hoaDon = new HoaDon();  hoaDon.setMaHoaDon(rs.getInt(1));  hoaDon.setMaNhanVien(rs.getInt(2));  hoaDon.setMaBan(rs.getInt(3));  hoaDon.setGhiChu(rs.getString(4));  hoaDon.setTrangThai(rs.getBoolean(5));  hoaDon.setNgayThanhToan(rs.getDate(6));  return hoaDon;  }  private List<HoaDon> select(String sql, Object... args) {  List<HoaDon> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, args);  while (rs.next()) {  HoaDon hoaDon = readFormResultSet(rs);  list.add(hoaDon);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return list;  }  public List<HoaDon> select() {  String sql = "SELECT \* FROM HoaDon";  return select(sql);  }  public HoaDon findById(int maHoaDon) {  String sql = "SELECT \* FROM HoaDon WHERE MaHoaDon= ?";  List<HoaDon> list = select(sql, maHoaDon);  return list.size() > 0 ? list.get(0) : null;  }  } |

#### HoaDonChiTietDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import model.HoaDonChiTiet;  public class HoaDonChiTietDAO {  public int getMaHDByMaHDCT(int id) {  try {  ResultSet rs = null;  String sql = "select MaHoaDon from HoaDonChiTiet where MaHoaDonCT = ?";  try {  rs = Jdbc.executeQuery(sql, id);  while (rs.next()) {  return rs.getInt(1);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  System.out.println(e.toString());  }  return -1;  }  public void insert(HoaDonChiTiet model, int maHoaDon) {  String sql = "INSERT INTO HoaDonChiTiet (MaHoaDon, SoLuongSP, MaSanPham) VALUES (?,?,?)";  Jdbc.executeUpdate(sql,  maHoaDon,  model.getSoLuongSP(),  model.getMaSanPham()  );  }  public void update(HoaDonChiTiet model, int id) {  String sql = "UPDATE HoaDonChiTiet SET SoLuongSP=?, MaSanPham= ? WHERE MaHoaDonCT=?";  Jdbc.executeUpdate(sql,  model.getSoLuongSP(),  model.getMaSanPham(),  id  );  }  public void delete(int maHoaDonCT) {  String sql = "DELETE FROM HoaDonChiTiet WHERE MaHoaDonCT=?";  Jdbc.executeUpdate(sql, maHoaDonCT);  }  private HoaDonChiTiet readFormResultSet(ResultSet rs) throws SQLException {  HoaDonChiTiet hoaDonChiTiet = new HoaDonChiTiet();  hoaDonChiTiet.setMaHoaDonCT(rs.getInt(1));  hoaDonChiTiet.setMaHoaDon(rs.getInt(2));  hoaDonChiTiet.setSoLuongSP(rs.getInt(3));  hoaDonChiTiet.setMaSanPham(rs.getInt(4));  return hoaDonChiTiet;  }  private List<HoaDonChiTiet> select(String sql, Object... args) {  List<HoaDonChiTiet> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  rs = Jdbc.executeQuery(sql, args);  while (rs.next()) {  HoaDonChiTiet hoaDonChiTiet = readFormResultSet(rs);  list.add(hoaDonChiTiet);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return list;  }  public List<HoaDonChiTiet> select() {  String sql = "SELECT \* FROM HoaDonChiTiet";  return select(sql);  }  public HoaDonChiTiet findById(int id) {  String sql = "SELECT \* FROM HoaDonChiTiet WHERE MaHoaDonCT=?";  List<HoaDonChiTiet> list = select(sql, id);  return list.size() > 0 ? list.get(0) : null;  }  public List<HoaDonChiTiet> selectByKeyword(int id) {  String sql = "SELECT \* FROM HoaDonChiTiet WHERE MaHoaDonCT=?";  List<HoaDonChiTiet> list = select(sql, "%" + id + "%");  return list;  }  } |

#### DanhMucDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  public class DanhMucDAO {  public int getMaHDCTByBanAndHD(int maBan, int maHD) {  try {  ResultSet rs = null;  String sql = "select MaHoaDonCT from HoaDon join HoaDonChiTiet on "  + "HoaDon.MaHoaDon = HoaDonChiTiet.MaHoaDon where "  + "HoaDon.MaBan = ? and HoaDon.MaHoaDon = ?";  try {  rs = Jdbc.executeQuery(sql, maBan, maHD);  while (rs.next()) {  return rs.getInt(1);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  System.out.println(e.toString());  }  return 0;  }  public List<Object[]> getDanhSachSPTheoBan(int maBan) {  List<Object[]> list = new ArrayList<>();  try {  ResultSet rs = null;  String sql = "select HoaDonChiTiet.MaHoaDonCT, SanPham.TenSanPham, LoaiSanPham.TenLoaiSP, SanPham.GiaBan, HoaDonChiTiet.SoLuongSP, \n"  + "(SanPham.GiaBan\*HoaDonChiTiet.SoLuongSP) as 'ThanhTien' from Ban join HoaDon on Ban.MaBan = \n"  + "HoaDon.MaBan join HoaDonChiTiet on HoaDon.MaHoaDon = HoaDonChiTiet.MaHoaDon join SanPham on \n"  + "HoaDonChiTiet.MaSanPham = SanPham.MaSanPham join LoaiSanPham on SanPham.MaLoaiSP = \n"  + "LoaiSanPham.MaLoaiSP where Ban.MaBan = ? and Ban.TrangThai = 1 and HoaDon.TrangThai = 0";  try {  rs = Jdbc.executeQuery(sql, maBan);  while (rs.next()) {  Object[] model = {  rs.getInt(1),  rs.getString(2),  rs.getString(3),  rs.getFloat(4),  rs.getInt(5),  rs.getFloat(6),};  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  }  return list;  }  public float getThanhToanTheoBan(int maBan) {  try {  ResultSet rs = null;  String sql = "SELECT Sum(HoaDon.ThanhTien) from HoaDon JOIN Ban ON "  + "HoaDon.MaBan = Ban.MaBan where ban.TrangThai = 1 and "  + "HoaDon.TrangThai = 0 and Ban.maBan = ?";  try {  rs = Jdbc.executeQuery(sql, maBan);  while (rs.next()) {  return rs.getFloat(1);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (Exception e) {  System.out.println(e.toString());  }  return 0;  }  } |

#### ThongKeDAO

|  |
| --- |
| package DAO;  import helper.Jdbc;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  /\*\*  \*  \* @author duann  \*/  public class ThongKeDAO {      public List<Object[]> getKhoHang() {  List<Object[]> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  String sql = "{call sp\_ThongKeKhoHang}";  rs = Jdbc.executeQuery(sql);  while (rs.next()) {  Object[] model = {  rs.getInt(1),  rs.getInt(2),  };  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  return list;  }      public List<Object[]> getNhanVien() {  List<Object[]> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  String sql = "{call sp\_ThongKeNhanVien}";  rs = Jdbc.executeQuery(sql);  while (rs.next()) {  Object[] model = {  rs.getString(1),  rs.getInt(2),  rs.getBoolean(3),  rs.getInt(4),  };  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  return list;  }    public List<Object[]> getThongKeNhanVien() {  List<Object[]> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  String sql = "{call sp\_ThongKeNhanVien}";  rs = Jdbc.executeQuery(sql);  while (rs.next()) {  Object[] model = {  rs.getString(1),  rs.getInt(2),  rs.getDate(3),  rs.getDate(4)  };  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  return list;  }  public List<Object[]> getDoanhThuNam() {  List<Object[]> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  String sql = "{call sp\_ThongKeDoanhThu}";  rs = Jdbc.executeQuery(sql);  while (rs.next()) {  Object[] model = {  rs.getInt(1),  rs.getInt(2)  };  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  return list;  }  public List<Object[]> getDoanhThuThang() {  List<Object[]> list = new ArrayList<>();  try {  ResultSet rs = null;  try {  String sql = "{call sp\_ThongKeDoanhThuTheoThang}";  rs = Jdbc.executeQuery(sql);  while (rs.next()) {  Object[] model = {  rs.getDate(1),  rs.getInt(2)  };  list.add(model);  }  } finally {  rs.getStatement().getConnection().close();  }  } catch (SQLException e) {  throw new RuntimeException(e);  }  return list;  }    } |

## Viết mã cho ứng dụng

### Các lớp tiện ích

Lập trình điều khiển hành vi tương tác của người dùng lên các giao diện của ứng dụng. Trước khi lập trình, chúng ta cần thực hiện 2 điều sau đây

#### ShareHelper

Lớp ShareHelper cung cấp phương thức tiện ích quản lý người đăng nhập và làm việc với hình ảnh chia sẽ trong tất cả các form của ứng dụng

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package helper;  import java.awt.Color;  import java.awt.Font;  import java.awt.Image;  import javax.swing.ImageIcon;  import javax.swing.JScrollPane;  import javax.swing.JTable;  import model.NhanVien;  /\*\*  \*  \* @author duann  \*/  public class ShareHelper {    // ảnh biểu tượng ứng dụng xuất hiện trên mọi cửa sổ  public static final Image APP\_ICON;  static {  // tải biểu tượng ứng dụng  String file = "/images/icons8-cafe-80.png";  APP\_ICON = new ImageIcon(ShareHelper.class.getResource(file)).getImage();  }  // đối tượng chứa thông tin của người sử dụng sau khi đăng nhập  public static NhanVien USER = null;  // Xóa thông tin của người sử dụng khi có yêu cầu đăng suất  public static void logOff() {  ShareHelper.USER = null;  }  // Kiểm tra xem đăng nhập hay chưa  public static boolean authenticated() {  return ShareHelper.USER != null;  }  // Set Table  public static void DrawTable(JTable tblGridView) {  tblGridView.getTableHeader().setFont(new Font("SansSerif", Font.BOLD, 12));  tblGridView.getTableHeader().setForeground(Color.blue);  // Column Width  tblGridView.getColumnModel().getColumn(0).setPreferredWidth(30);  tblGridView.setForeground(Color.black);  // Row Height  tblGridView.setRowHeight(20);  }      public static void setBoderForTable(JScrollPane scp) {  scp.setViewportBorder(null);  scp.setBorder(null);  }    } |

#### DialogHelper

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package helper;  import java.awt.Color;  import java.awt.Component;  import java.awt.Font;  import java.awt.event.ActionEvent;  import java.awt.event.ActionListener;  import javax.swing.JLabel;  import javax.swing.JOptionPane;  import javax.swing.Timer;  /\*\*  \*  \* @author duann  \*/  public class DialogHelper {  // hiển thị thông báo cho người dùng  public static void alert(Component parent, String messager) {  JOptionPane.showMessageDialog(parent, messager, "Hệ thống quản lý quán coffee", JOptionPane.INFORMATION\_MESSAGE);  }  // Hiển thị thông báo và yêu cầu người dùng xác nhận  public static boolean confirm(Component parent, String messager) {  int result = JOptionPane.showConfirmDialog(parent, messager, "Hệ thống quản lý quán coffee", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);  return result == JOptionPane.YES\_OPTION;  }  // hiển thị thông báo yêu cầu nhập dữ liệu;  public static String prompt(Component parent, String messager) {  return JOptionPane.showInputDialog(parent, messager, "Hệ thống quản lý quán coffee", JOptionPane.INFORMATION\_MESSAGE);  }  public static void setInfinity(JLabel label, String text) {  label.setVisible(true);  label.setForeground(new Color(86, 188, 27));  label.setFont(new Font("SansSerif", Font.BOLD, 13));  label.setText(text);  new Timer(3000, new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  label.setText("");  label.setVisible(false);  }  }).start();  }  } |

### Viết mã cho các chức năng

Trong phần này chúng ta sẽ tiến hành viết mã cho các giao diện đã thiết kế trước đây. Cụ thể chúng ta sẽ lần lược viết mã thực hiện các chức năng cho các giao diện sau đây.

#### Màn hình chào (ChaoJDialog)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

Bổ sung vào cuối cửa sổ hàm init() sau đây.

|  |
| --- |
| void init() {  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  setLocationRelativeTo(null);  new Timer(5000, new ActionListener() {  @Override  public void actionPerformed(ActionEvent e) {  ChaoJDialog.this.dispose();  }  }).start();  } |

#### Form đăng nhập (DangNhapJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| btnDangNhap | login(); |
| btnKetThuc | exit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ

Bổ sung mã sau vào cuối lớp DangNhapJFrame.

|  |
| --- |
| void init() {  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  }  boolean checkLogin() {  boolean check = true;  if (txtTenDangNhap.getText().isEmpty()) {  DialogHelper.alert(this, "Bạn phải nhập email để đăng nhập.");  check = false;  }  if (txtMatKhau.getText().isEmpty()) {  DialogHelper.alert(this, "Bạn phải nhập mật khẩu để đăng nhập.");  check = false;  }  return check;  }  void login() {  if (checkLogin()) {  String email = txtTenDangNhap.getText();  String matKhau = new String(txtMatKhau.getPassword());  try {  NhanVien model = dao.findByEmail(email);  if (model != null) {  String matKhau2 = model.getMatKhau();  if (matKhau.equals(matKhau2)) {  ShareHelper.USER = model;  DialogHelper.alert(this, "Đăng nhập thành công!");  this.dispose();  } else {  DialogHelper.alert(this, "Sai mật khẩu!");  }  } else {  DialogHelper.alert(this, "Sai email đăng nhập!");  }  } catch (Exception e) {  DialogHelper.alert(this, "Đăng nhập thất bại, vui lòng kiểm tra lại kết nối.");  }  }  }  void exit() {  if (DialogHelper.confirm(this, "Bạn có muốn thoát khỏi ứng dụng không!")) {  System.exit(0);  }  } |

#### Giao diện chính (DanhMucJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblHoaDonMouseClicked.mouseClicked | if (evt.getClickCount() == 2) {  this.index = tblHoaDon.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  thongTinTheoBan((int) tblHoaDon.getValueAt(index, 0));  }  } |
| mniBan | new BanJFrame().setVisible(true); |
| mniKhuVuc | new KhuVucJFrame().setVisible(true); |
| mniSanPham | new SanPhamJFrame().setVisible(true); |
| mniLoaiSanPham | new LoaiSanPhamJFrame().setVisible(true); |
| mniNhanVien | new NhanVienJFrame().setVisible(true); |
| mniCaLamViec | new CaLamViecJFrame().setVisible(true); |
| mniKhoHang | new KhoHangJFrame().setVisible(true); |
| mniThongKe | new ThongKeJFrame().setVisible(true); |
| mniChiTietMouseClicked | this.xemThongTinChiTiet(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.\*;  import helper.\*;  import java.awt.event.\*;  import java.util.List;  import javax.swing.\*;  import javax.swing.table.DefaultTableModel;  import model.Ban;  /\*\*  \*  \* @author duann  \*/  public class DanhMucJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form DanhMucJFrame  \*/  public static BanDAO banDAO = new BanDAO();  public static KhoHangDAO khDAO = new KhoHangDAO();  public static SanPhamDAO spDAO = new SanPhamDAO();  public static LoaiSanPhamDAO lspDAO = new LoaiSanPhamDAO();  public static HoaDonDAO hdDAO = new HoaDonDAO();  public static KhuVucDAO kvDAO = new KhuVucDAO();  DanhMucDAO dmdao = new DanhMucDAO();  public static JLabel ban;  public static int index = 0;  public static ImageIcon CafeGreen;  public static ImageIcon CafeBlack;  static {  CafeGreen = new ImageIcon(DanhMucJFrame.class.getResource("/images/icons8-cafe-50.png"));  CafeBlack = new ImageIcon(DanhMucJFrame.class.getResource("/images/icons8-cafe-filled-50.png"));  }  public DanhMucJFrame() {  initComponents();  init();  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  this.setLocationRelativeTo(null);  this.setTitle("Hệ thống quản lý quán coffee");  this.openWelcome();  this.openLogin();  loadTabs();  loadDonHangTheoBan();  setBoderForTable(jScrollPane1);  }  void openWelcome() {  new ChaoJDialog(this, true).setVisible(true);  }  void openLogin() {  new DangNhapJDialog(this, true).setVisible(true);  }  public static void loadTabs() {  loadBanChung();  }  void setBoderForTable(JScrollPane scp) {  scp.setViewportBorder(null);  scp.setBorder(null);  }  public static void loadBanChung() {  List<Ban> list = banDAO.select();  pnlBan.removeAll();  for (int i = 0; i < list.size(); i++) {  if (list.get(i).isTrangThai()) {  int id = i;  ban = new JLabel(CafeGreen);  if (list.get(i).getMaBan() < 10) {  ban.setText("Bàn 0" + list.get(i).getMaBan());  } else {  ban.setText("Bàn " + list.get(i).getMaBan());  }  ban.addMouseListener(new MouseAdapter() {  @Override  public void mouseReleased(MouseEvent e) {  if (e.getClickCount() == 2) {  thongTinTheoBan(list.get(id).getMaBan());  }  }  });  pnlBan.add(ban);  pnlBan.revalidate();  pnlBan.setVisible(true);  } else {  ban = new JLabel(CafeBlack);  if (list.get(i).getMaBan() < 10) {  ban.setText("Bàn 0" + list.get(i).getMaBan());  } else {  ban.setText("Bàn " + list.get(i).getMaBan());  }  int id = i;  ban.addMouseListener(new MouseAdapter() {  @Override  public void mouseReleased(MouseEvent e) {  if (e.getClickCount() == 2) {  thongTinTheoBan(list.get(id).getMaBan());  }  }  });  pnlBan.add(ban);  pnlBan.revalidate();  pnlBan.setVisible(true);  }  }  }  public static void thongTinTheoBan(int id) {  new ThongTinDonHangJFrame(id).setVisible(true);  }  void xemThongTinBan() {  DialogHelper.alert(this, "Xem thông tin bàn");  }  void xemThongTinChiTiet() {  new HoaDonChiTietJFrame().setVisible(true);  }  public static void loadDonHangTheoBan() {  DefaultTableModel model = (DefaultTableModel) tblHoaDon.getModel();  model.setRowCount(0);  try {  List<Object[]> list = hdDAO.getHoaDonTheoBan();  for (Object[] objects : list) {  model.addRow(objects);  }  } catch (Exception e) {  System.out.println(e.toString());  }  } |

#### Quản lý khu vực(KhuVucJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblKhuVuc.mouseClicked | if (evt.getClickCount() == 1) {  this.index=tblKhuVuc.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  tabs.setSelectedIndex(0);  this.fillToForm();  }  } |
| btnThem | insert(); |
| btnSua | update(); |
| btnXao | delete(Integer.parseInt(txtTenKhuVuc.getToolTipText())); |
| btnMoi | clear(); |
| btnFirst | this.index = 0;  this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++;  this.edit(); |
| btnLast | this.index = tblKhuVuc.getRowCount() - 1; this.edit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.KhuVucDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import java.util.List;  import javax.swing.table.DefaultTableModel;  import model.KhuVuc;  /\*\*  \*  \* @author NgọcHải  \*/  public class KhuVucJFrame extends javax.swing.JFrame {  int index = 0;  KhuVucDAO dao = new KhuVucDAO();  /\*\*  \* Creates new form KhuVucJFrame  \*/  public KhuVucJFrame() {  initComponents();  init();  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  this.load();  this.setStatus(true);  this.setTitle("Hệ thống quản lý quán coffee");  ShareHelper.setBoderForTable(jScrollPane2);  }  void load() {  DefaultTableModel model = (DefaultTableModel) tblKhuVuc.getModel();  model.setRowCount(0);  try {  List<KhuVuc> list = dao.select();  for (KhuVuc kv : list) {  Object[] row = {  kv.getMaKhuVuc(),  kv.getTenKhuVuc(),  kv.getViTri(),  kv.getGhiChu()  };  model.addRow(row);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  }  }  // set du lieu nhap vao  void setModel(KhuVuc model) {  txtTenKhuVuc.setToolTipText(String.valueOf(model.getMaKhuVuc()));  txtTenKhuVuc.setText(model.getTenKhuVuc());  txtViTri.setText(model.getViTri());  txtGhiChu.setText(model.getGhiChu());  }  KhuVuc getModel() {  KhuVuc model = new KhuVuc();  model.setTenKhuVuc(txtTenKhuVuc.getText());  model.setViTri(txtViTri.getText());  model.setGhiChu(txtGhiChu.getText());  if (txtTenKhuVuc.getToolTipText() != null) {  model.setMaKhuVuc(Integer.parseInt(txtTenKhuVuc.getToolTipText()));  }  return model;  }  void setStatus(boolean insertable) {  btnThem.setEnabled(insertable);  btnSua.setEnabled(!insertable);  btnXoa.setEnabled(!insertable);  boolean first = this.index > 0;  boolean last = this.index < tblKhuVuc.getRowCount() - 1;  btnFirst.setEnabled(!insertable && first);  btnPrev.setEnabled(!insertable && first);  btnNext.setEnabled(!insertable && last);  btnLast.setEnabled(!insertable && last);  }  void clear() {  this.setModel(new KhuVuc());  this.setStatus(true);  }  void insert() {  KhuVuc model = getModel();  try {  dao.insert(model);  this.load();  this.clear();  DialogHelper.setInfinity(lblThongBao, "Thêm mới thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Thêm mới thất bại!");  }  }  void update() {  KhuVuc model = getModel();  try {  dao.update(model);  this.load();  DialogHelper.setInfinity(lblThongBao, "Cập nhật thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Cập nhật thất bại!");  }  }  void delete(Integer maKhuVuc) {  if (DialogHelper.confirm(this, "Bạn thực sự muốn xóa khu vực này?")) {  try {  dao.delete(maKhuVuc);  this.load();  this.clear();  DialogHelper.setInfinity(lblThongBao, "Xóa thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Không thể xóa được vì dữ liệu đã được liên kết");  }  }  }  void edit() {  try {  int maKhuVuc = (int) tblKhuVuc.getValueAt(this.index, 0);  KhuVuc model = dao.findById(maKhuVuc);  if (model != null) {  this.setModel(model);  this.setStatus(false);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  }  } |

#### Quản lý bàn(BanJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblBan.mouseClicked | if (evt.getClickCount() == 1) {  this.index = tblBan.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  tabs.setSelectedIndex(0);  this.fillToForm();  }  } |
| btnThem | this.insert();  DanhMucJFrame.loadTabs();  DanhMucJFrame.loadDonHangTheoBan(); |
| btnSua | update();  DanhMucJFrame.loadTabs();  DanhMucJFrame.loadDonHangTheoBan(); |
| btnXao | delete(Integer.parseInt(txtMaBan.getText()));  this.load();  DanhMucJFrame.loadTabs();  DanhMucJFrame.loadDonHangTheoBan(); |
| btnMoi | clear(); |
| btnFirst | this.index = 0;  this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++;  this.edit(); |
| btnLast | this.index = tblBan.getRowCount() - 1; this.edit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.BanDAO;  import DAO.KhuVucDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import java.util.List;  import javax.swing.DefaultComboBoxModel;  import javax.swing.table.DefaultTableModel;  import model.Ban;  import model.KhuVuc;  /\*\*  \*  \* @author NgọcHải  \*/  public class BanJFrame extends javax.swing.JFrame {  int index = 0;  BanDAO dao = new BanDAO();  KhuVucDAO kvdao = new KhuVucDAO();  /\*\*  \* Creates new form BanJFrame  \*/  public BanJFrame() {  initComponents();  init();  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  load();  this.fillComboBox();  this.setStatus(true);  this.setTitle("Hệ thống quản lý quán coffee");  ShareHelper.setBoderForTable(jScrollPane1);  }  void load() {  DefaultTableModel model = (DefaultTableModel) tblBan.getModel();  model.setRowCount(0);  try {  List<Ban> list = dao.select();  for (Ban ban : list) {  Object[] row = {  ban.getMaBan(),  ban.getMaKhuvuc(),};  model.addRow(row);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  }  }  void setModel(Ban model) {  try {  txtMaBan.setText(String.valueOf(model.getMaBan()));  cboKhuVuc.setSelectedItem(String.valueOf(model.getMaKhuvuc()));  KhuVuc cd = kvdao.findById(model.getMaKhuvuc());  cboKhuVuc.setSelectedItem(cd.getTenKhuVuc());  } catch (Exception e) {  }  }  Ban getModel() {  Ban model = new Ban();  KhuVuc khuVuc = new KhuVuc();  model.setMaBan(Integer.valueOf(txtMaBan.getText()));  String tenKhuVuc = (String) cboKhuVuc.getSelectedItem();  KhuVuc kv = kvdao.findByName(tenKhuVuc);  model.setMaKhuvuc(kv.getMaKhuVuc());  return model;  }  void setStatus(boolean insertable) {  btnThem.setEnabled(insertable);  btnSua.setEnabled(!insertable);  btnXoa.setEnabled(!insertable);  boolean first = this.index > 0;  boolean last = this.index < tblBan.getRowCount() - 1;  btnFirst.setEnabled(!insertable && first);  btnPrev.setEnabled(!insertable && first);  btnNext.setEnabled(!insertable && last);  btnLast.setEnabled(!insertable && last);  }  void clear() {  Ban model = new Ban();  // String khuVuc = (String) cboKhuVuc.getSelectedItem();  // KhuVuc a = kvdao.findByName(khuVuc);  // model.setMaKhuvuc(a.getMaKhuVuc());  this.setModel(model);  this.setStatus(true);  this.cboKhuVuc.setSelectedIndex(0);  txtMaBan.setText("");  }  void edit() {  try {  Integer maBan = (Integer) tblBan.getValueAt(this.index, 0);  Ban model = dao.findById(maBan);  if (model != null) {  this.setModel(model);  this.setStatus(false);  }  } catch (Exception e) {  }  }  void insert() {  Ban model = getModel();  try {  dao.insert(model);  this.load();  this.clear();  DialogHelper.setInfinity(lblThongBao, "Thêm mới thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Thêm mới thất bại!");  System.out.println(e.toString());  }  }  void update() {  Ban model = getModel();  try {  dao.update(model);  this.load();  DialogHelper.setInfinity(lblThongBao, "Thêm mới thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Thêm mới thất bại!");  }  }  void delete(Integer maBan) {  if (DialogHelper.confirm(this, "Bạn thực sự muốn xóa bàn này?")) {  try {  dao.delete(maBan);  this.load();  this.clear();  DialogHelper.setInfinity(lblThongBao, "Xóa thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Không thể xóa được vì dữ liệu đã được liên kết!");  }  }  }  void fillComboBox() {  DefaultComboBoxModel model = (DefaultComboBoxModel) cboKhuVuc.getModel();  model.removeAllElements();  try {  List<KhuVuc> list = kvdao.select();  for (KhuVuc kv : list) {  model.addElement(kv.getTenKhuVuc());  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  }  } |

#### Quản lý loại sản phẩm(LoaiSanPhamJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblLoaiSanPham.mouseClicked | if (evt.getClickCount() == 1) {  this.index= tblLoaiSanPham.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  tabs.setSelectedIndex(0);  this.fillToForm();  }  } |
| btnThem | insert(); |
| btnSua | update(); |
| btnXao | delete(Integer.parseInt(txtLoaiSanPham.getToolTipText())); |
| btnMoi | clear(); |
| btnFirst | this.index = 0;  this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++;  this.edit(); |
| btnLast | this.index = tblLoaiSanPham.getRowCount() - 1; this.edit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.LoaiSanPhamDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import java.util.List;  import javax.swing.table.DefaultTableModel;  import model.LoaiSanPham;  /\*\*  \*  \* @author Admin  \*/  public class LoaiSanPhamJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form LoaiSanPhamJFrame  \*/  public LoaiSanPhamJFrame() {  initComponents();  init();  }  int index = 0;  LoaiSanPhamDAO dao = new LoaiSanPhamDAO();    void init(){  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  this.load();  this.setStatus(true);  this.setTitle("Hệ thống quản lý quán coffee");  ShareHelper.setBoderForTable(jScrollPane1);  }    void load() {  DefaultTableModel model = (DefaultTableModel) tblLoaiSanPham.getModel();  model.setRowCount(0);  try {  List<LoaiSanPham> list = dao.select();  for (LoaiSanPham lsp : list) {  Object[] row = {  lsp.getMaLoaiSP(),  lsp.getTenLoaiSP()  };  model.addRow(row);    }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy ván dữ liệu");  }  }    void insert() {  LoaiSanPham model = getModel();  try {  dao.insert(model);  this.load();  this.clear();  DialogHelper.alert(this, "Thêm mới thành công");  } catch (Exception e) {  DialogHelper.alert(this, "Thêm mới thất bại");  }  }    void update() {  LoaiSanPham model = getModel();  try {  dao.update(model);  this.load();  this.clear();  DialogHelper.alert(this, "Cập nhâp thành công");  } catch (Exception e) {  DialogHelper.alert(this, "Cập nhập thất bại");  }  }    void delete(int maLoaiSP) {  if (DialogHelper.confirm(this, "Bạn muốn xóa hàng này")) {  try {  dao.delete(maLoaiSP);  this.load();  this.clear();  DialogHelper.alert(this, "Xoá thành công");  } catch (Exception e) {  DialogHelper.alert(this, "Xóa thất bại");  }  }  }    void clear() {  LoaiSanPham model = new LoaiSanPham();  model.setMaLoaiSP(model.getMaLoaiSP());  model.setTenLoaiSP(model.getTenLoaiSP());  this.setStatus(true);  txtLoaiSanPham.setText("");  }    void edit() {  try {  int malsp = (int) tblLoaiSanPham.getValueAt(this.index, 0);  LoaiSanPham model = dao.findById(malsp);    if (model != null) {  this.setModel(model);  this.setStatus(false);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");  }  }    void setModel(LoaiSanPham model) {  txtLoaiSanPham.setText(model.getTenLoaiSP());  txtLoaiSanPham.setToolTipText(String.valueOf(model.getMaLoaiSP()));  }    LoaiSanPham getModel() {  LoaiSanPham model = new LoaiSanPham();  model.setTenLoaiSP(txtLoaiSanPham.getText());  if (txtLoaiSanPham.getToolTipText() != null) {  model.setMaLoaiSP(Integer.parseInt(txtLoaiSanPham.getToolTipText()));  }  return model;  }    void setStatus(boolean insertable) {  btnThem.setEnabled(insertable);  btnSua.setEnabled(!insertable);  btnXoa.setEnabled(!insertable);    boolean first = this.index > 0;  boolean last = this.index < tblLoaiSanPham.getRowCount() - 1;    btnNext.setEnabled(!insertable && last);  btnLast.setEnabled(!insertable && last);  btnFirst.setEnabled(!insertable && first);  btnPrev.setEnabled(!insertable && first);    } |

#### Quản lý sản phẩm(SanPhamJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblSanPham.mouseClicked | if (evt.getClickCount() == 1) {  this.index = tblSanPham.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  tabs.setSelectedIndex(0);  this.fillToForm();  }  } |
| btnThem | insert(); |
| btnSua | update(); |
| btnXao | delete(); |
| btnMoi | clear(); |
| btnFirst | this.index = 0;  this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++;  this.edit(); |
| btnLast | this.index = tblSanPham.getRowCount() - 1; this.edit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.LoaiSanPhamDAO;  import DAO.SanPhamDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import java.util.List;  import javax.swing.DefaultComboBoxModel;  import javax.swing.table.DefaultTableModel;  import model.LoaiSanPham;  import model.SanPham;  /\*\*  \*  \* @author Admin  \*/  public class SanPhamJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form SanPhamJFrame  \*/  int index = 0;  SanPhamDAO dao = new SanPhamDAO();  LoaiSanPhamDAO lspdao = new LoaiSanPhamDAO();  public SanPhamJFrame() {  initComponents();  init();  }  void init() {  fillCombobox();  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  this.load();  this.setStatus(true);  this.setTitle("Hệ thống quản lý quán coffee");  ShareHelper.setBoderForTable(jScrollPane2);  }  void fillCombobox() {  DefaultComboBoxModel model = (DefaultComboBoxModel) cboLoaiSanPham.getModel();  model.removeAllElements();  try {  List<LoaiSanPham> list = lspdao.select();  for (LoaiSanPham loaisanpham : list) {  model.addElement(loaisanpham.getTenLoaiSP());  }  } catch (Exception e) {  }  }  void load() {  DefaultTableModel model = (DefaultTableModel) tblSanPham.getModel();  model.setRowCount(0);  try {  List<SanPham> list = dao.select();  for (SanPham sp : list) {  Object[] row = {  sp.getMaSanPham(),  sp.getTenSanPham(),  lspdao.findById(sp.getMaLoaiSP()).getTenLoaiSP(),  sp.getGiaBan(),  sp.getGhiChu()  };  model.addRow(row);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  }  }  void setModel(SanPham model) {  txtTenSanPham.setToolTipText(String.valueOf(model.getMaSanPham()));  txtTenSanPham.setText(model.getTenSanPham());  cboLoaiSanPham.setToolTipText(String.valueOf(model.getMaSanPham()));  LoaiSanPham lsp = lspdao.findById(model.getMaLoaiSP());  cboLoaiSanPham.setSelectedItem(lsp.getTenLoaiSP());  txtGiaBan.setText(String.valueOf(model.getGiaBan()));  txtGhiChu.setText(model.getGhiChu());  }  SanPham getModel() {  SanPham sanpham = new SanPham();  String tenLoaiSanPham = (String) cboLoaiSanPham.getSelectedItem();  LoaiSanPham lsp = lspdao.findByName(tenLoaiSanPham);  sanpham.setMaLoaiSP(lsp.getMaLoaiSP());  sanpham.setTenSanPham(txtTenSanPham.getText());  sanpham.setGiaBan(Float.valueOf(txtGiaBan.getText()));  sanpham.setGhiChu(txtGhiChu.getText());  if (txtTenSanPham.getToolTipText() != null) {  sanpham.setMaSanPham(Integer.valueOf(txtTenSanPham.getToolTipText()));  }  return sanpham;  }  void setStatus(boolean insertTable) {  try {  btnThem.setEnabled(insertTable);  btnSua.setEnabled(!insertTable);  btnXoa.setEnabled(!insertTable);  boolean first = this.index > 0;  boolean last = this.index < tblSanPham.getRowCount() - 1;  btnFirst.setEnabled(!insertTable && first);  btnPrev.setEnabled(!insertTable && first);  btnNext.setEnabled(!insertTable && last);  btnLast.setEnabled(!insertTable && last);  } catch (Exception e) {  }  }  void clear() {  SanPham model = new SanPham();  String loaisp = (String) cboLoaiSanPham.getSelectedItem();  LoaiSanPham a = lspdao.findByName(loaisp);  model.setMaLoaiSP(a.getMaLoaiSP());  this.cboLoaiSanPham.setSelectedItem(0);  this.setModel(model);  txtGiaBan.setText("");  }  void insert() {  SanPham model = getModel();  try {  dao.insert(model);  this.load();  this.clear();  DialogHelper.alert(this, "Thêm mới thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Thêm mới thất bại!");  }  }  void update() {  SanPham model = getModel();  try {  dao.update(model);  this.load();  this.clear();  DialogHelper.alert(this, "Cập nhập thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Cập nhập thất bại");  }  }  void delete() {  if (DialogHelper.confirm(this, "Bạn thực sự muốn xóa sản phẩm này")) {  Integer masp = Integer.valueOf(txtTenSanPham.getToolTipText());  try {  dao.delete(masp);  this.load();  this.clear();  DialogHelper.alert(this, "Xóa thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Xóa thất bại");  }  }  }  void edit() {  try {  int masp = (int) tblSanPham.getValueAt(this.index, 0);  SanPham model = dao.findById(masp);  if (model != null) {  this.setModel(model);  this.setStatus(false);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");  }  } |

#### Quản lý ca làm việc(CaLamViecJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblCaLamViec.mouseClicked | if(evt.getClickCount()==1){ this.index=tblCaLamViec.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  tabs.setSelectedIndex(0);  this.fillToForm();  }  } |
| btnThem | insert(); |
| btnSua | update(); |
| btnXao | delete(Integer.parseInt(txtTenCaLV.getToolTipText())); |
| btnMoi | clear(); |
| btnFirst | this.index = 0;  this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++;  this.edit(); |
| btnLast | this.index = tblCaLamViec.getRowCount() - 1; this.edit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.CaLamViecDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import java.text.ParseException;  import java.util.List;  import java.util.logging.Level;  import java.util.logging.Logger;  import javax.swing.table.DefaultTableModel;  import model.CaLamViec;  /\*\*  \*  \* @author duann  \*/  public class CaLamViecJFrame extends javax.swing.JFrame {  int index = 0;  CaLamViecDAO dao = new CaLamViecDAO();  /\*\*  \* Creates new form CaLamViecJFrame  \*/  public CaLamViecJFrame() {  initComponents();  init();  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  this.setLocationRelativeTo(null);  this.load();  this.setStatus(true);  this.setTitle("Hệ thống quản lý quán coffee");  ShareHelper.setBoderForTable(jScrollPane2);  }  void load() {  DefaultTableModel model = (DefaultTableModel) tblCaLamViec.getModel();  model.setRowCount(0);  try {  List<CaLamViec> list = dao.select();  for (CaLamViec clv : list) {  Object[] row = {  clv.getMaCaLamViec(),  clv.getTenCaLamViec(),  clv.getBatDau(),  clv.getKetThuc(),  clv.getGhiChu()};  model.addRow(row);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  }  }  void setModel(CaLamViec model) throws ParseException {  txtTenCaLV.setToolTipText(String.valueOf(model.getMaCaLamViec()));  txtTenCaLV.setText(model.getTenCaLamViec());  txtBatDau.setText(model.getBatDau());  txtKetThuc.setText(model.getKetThuc());  txtGhiChu.setText(model.getGhiChu());  }  CaLamViec getModel() {  CaLamViec model = new CaLamViec();  model.setTenCaLamViec(txtTenCaLV.getText());  model.setBatDau((String) txtBatDau.getText());  model.setKetThuc((String) txtKetThuc.getText());  model.setGhiChu((String) txtGhiChu.getText());  if (txtTenCaLV.getToolTipText() != null) {  model.setMaCaLamViec(Integer.parseInt(txtTenCaLV.getToolTipText()));  }  return model;  }  void setStatus(boolean insertable) {  btnThem.setEnabled(insertable);  btnSua.setEnabled(!insertable);  btnXoa.setEnabled(!insertable);  boolean first = this.index > 0;  boolean last = this.index < tblCaLamViec.getRowCount() - 1;  btnFirst.setEnabled(!insertable && first);  btnPrev.setEnabled(!insertable && first);  btnNext.setEnabled(!insertable && last);  btnLast.setEnabled(!insertable && last);  }  void clear() throws ParseException {  CaLamViec model = new CaLamViec();  this.setModel(model);  this.setStatus(true);  }  void insert() {  CaLamViec model = getModel();  try {  dao.insert(model);  this.load();  DialogHelper.setInfinity(lblThongBao, "Thêm mới thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Thêm mới thất bại!");  }  }  void update() {  CaLamViec model = getModel();  try {  dao.update(model);  this.load();  DialogHelper.setInfinity(lblThongBao, "Cập nhật thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Cập nhật thất bại!");  }  }  void delete(Integer maCaLamViec) {  if (DialogHelper.confirm(this, "Bạn thực sự muốn xóa khu vực này?")) {  try {  dao.delete(maCaLamViec);  this.load();  this.clear();  DialogHelper.setInfinity(lblThongBao, "Xóa thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Không thể xóa được vì dữ liệu đã được liên kết!");  }  }  }  void edit() {  try {  int maCa = (int) tblCaLamViec.getValueAt(this.index, 0);  CaLamViec model = dao.findById(maCa);  if (model != null) {  this.setModel(model);  this.setStatus(false);  }  } catch (Exception e) {  System.out.println(e.toString());  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu!");  }  } |

#### Quản lý nhân viên(NhanVienJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblNhanVien.mouseClicked | if (evt.getClickCount() == 1) {  this.index= tblNhanVien.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  tabs.setSelectedIndex(0);  this.fillToForm();  }  } |
| btnThem | insert(); |
| btnSua | update(); |
| btnXao | delete(); |
| btnMoi | clear(); |
| btnFirst | this.index = 0;  this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++;  this.edit(); |
| btnLast | this.index = tblNhanVien.getRowCount() - 1; this.edit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.CaLamViecDAO;  import DAO.NhanVienDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import helper.XDate;  import java.util.Date;  import java.util.List;  import javax.swing.ButtonGroup;  import javax.swing.DefaultComboBoxModel;  import javax.swing.table.DefaultTableModel;  import model.CaLamViec;  import model.NhanVien;  /\*\*  \*  \* @author duann  \*/  public class NhanVienJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form NhanVienJFrame  \*/  NhanVienDAO dao = new NhanVienDAO();  ButtonGroup grGioiTinh;  ButtonGroup grVaiTro;  CaLamViecDAO clvdao = new CaLamViecDAO();  int index = 0;  public NhanVienJFrame() {  initComponents();  init();  }  void init() {  this.setLocationRelativeTo(null);  setIconImage(ShareHelper.APP\_ICON);  grGioiTinh = new ButtonGroup();  grGioiTinh.add(rdoNam);  grGioiTinh.add(rdoNu);  grVaiTro = new ButtonGroup();  grVaiTro.add(rdoQuanLy);  grVaiTro.add(rdoNhanVien);  tabs.setSelectedIndex(1);  fillComboboxCaLV();  this.load();  this.clear();  }  void load() {  DefaultTableModel model = (DefaultTableModel) tblNhanVien.getModel();  model.setRowCount(0);  try {  List<NhanVien> list = dao.select();  for (NhanVien nhanVien : list) {  Object[] row = {  nhanVien.getMaNhanVien(),  nhanVien.getHoTen(),  XDate.toString(nhanVien.getNgaySinh()),  nhanVien.getDienThoai(),  nhanVien.isGioiTinh() ? "Nữ" : "Nam",  nhanVien.getEmail(),  nhanVien.getMatKhau().length() > 0 ? "\*\*\*\*\*" : "\*\*\*\*\*",  nhanVien.isVaiTro() ? "Nhân viên" : "Quản lý",  nhanVien.getCaLamViec().getTenCaLamViec()  };  model.addRow(row);  }  } catch (Exception e) {  }  }  void fillComboboxCaLV() {  DefaultComboBoxModel model = (DefaultComboBoxModel) cboCaLamViec.getModel();  model.removeAllElements();  try {  List<CaLamViec> list = clvdao.select();  for (CaLamViec caLamViec : list) {  model.addElement(caLamViec.getTenCaLamViec());  }  } catch (Exception e) {  }  }  NhanVien getModel() {  NhanVien nhanVien = new NhanVien();  try {  nhanVien.setMaNhanVien(Integer.parseInt(txtHoTen.getToolTipText()));  } catch (Exception e) {  }  nhanVien.setHoTen(txtHoTen.getText());  nhanVien.setMatKhau(txtMatKhau.getText());  nhanVien.setGioiTinh(rdoNu.isSelected());  nhanVien.setVaiTro(rdoNhanVien.isSelected());  nhanVien.setNgaySinh(XDate.toDate(txtNgaySinh.getText()));  nhanVien.setEmail(txtEmail.getText());  nhanVien.setDienThoai(txtDienThoai.getText());  nhanVien.setGhiChu(txtGhiChu.getText());  nhanVien.setMaCaLamViec(clvdao.findByName((String) cboCaLamViec.getSelectedItem()).getMaCaLamViec());  nhanVien.setSoNgayLamViec(0);  return nhanVien;  }  void setModel(NhanVien model) {  try {  cboCaLamViec.setSelectedItem(model.getCaLamViec().getTenCaLamViec());  } catch (Exception e) {  cboCaLamViec.setSelectedIndex(0);  }  txtHoTen.setToolTipText(String.valueOf(model.getMaNhanVien()));  txtHoTen.setText(model.getHoTen());  txtNgaySinh.setText(XDate.toString(model.getNgaySinh()));  txtDienThoai.setText(model.getDienThoai());  rdoNam.setSelected(!model.isGioiTinh());  rdoNu.setSelected(model.isGioiTinh());  txtEmail.setText(model.getEmail());  txtMatKhau.setText(model.getMatKhau());  txtMatKhauXacNhan.setText(model.getMatKhau());  rdoNhanVien.setSelected(model.isVaiTro());  rdoQuanLy.setSelected(!model.isVaiTro());  txtGhiChu.setText(model.getGhiChu());  }  void setStatus(boolean insertTable) {  try {  btnThem.setEnabled(insertTable);  btnSua.setEnabled(!insertTable);  btnXoa.setEnabled(!insertTable);  boolean first = this.index > 0;  boolean last = this.index < tblNhanVien.getRowCount() - 1;  btnFirst.setEnabled(!insertTable && first);  btnPrev.setEnabled(!insertTable && first);  btnNext.setEnabled(!insertTable && last);  btnLast.setEnabled(!insertTable && last);  } catch (Exception e) {  }  }  void fillToForm() {  try {  Integer maNhanVien = (Integer) tblNhanVien.getValueAt(this.index, 0);  NhanVien model = dao.findById(maNhanVien);  if (model != null) {  this.setModel(model);  this.setStatus(false);  }  } catch (Exception e) {  System.out.println(e.toString());  }  }  void clear() {  NhanVien model = new NhanVien();  this.setModel(model);  setStatus(true);  rdoNam.setSelected(true);  rdoNhanVien.setSelected(true);  cboCaLamViec.setSelectedIndex(0);  txtNgaySinh.setText("");  }  private void insert() {  NhanVien model = getModel();  try {  dao.insert(model);  this.load();  this.clear();  DialogHelper.setInfinity(lblMsg, "Thêm mới thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Thêm mới thất bại!");  }  }  void update() {  NhanVien model = getModel();  try {  dao.update(model);  this.load();  DialogHelper.setInfinity(lblMsg, "Cập nhật thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Cập nhật thất bại!");  System.out.println("error: " + e.toString());  }  }  void delete() {  if (DialogHelper.confirm(this, "Bạn có thực sự muốn xóa nhân viên này?")) {  Integer maNhanVien = Integer.valueOf(txtHoTen.getToolTipText());  try {  dao.delete(maNhanVien);  this.load();  this.clear();  DialogHelper.setInfinity(lblMsg, "Xóa thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Xóa thất bại!");  System.out.println(e.toString());  }  }  } |

#### Quản lý thông tin đơn hàng(ThongTinDonHangJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblThongTin.mouseClicked | this.index = tblThongTin.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  setModel(daoCT.findById((int) tblThongTin.getValueAt(index, 0)));  mahdct = (int) tblThongTin.getValueAt(index, 0);  soLuong = Integer.parseInt(txtSoLuong.getText());  maSanPham = spDAO.findByName((String) cboSanPham.getSelectedItem()).getMaSanPham();  }  this.tinhTien(); |
| tblSanPham.mouseClicked | this.index = tblSanPham.rowAtPoint(evt.getPoint());  if (tblSanPham.getValueAt(index, 0).toString().equals(cboSanPham.getSelectedItem())) {  dem += 1;  txtSoLuong.setText(String.valueOf(dem));  } else {  dem = 1;  txtSoLuong.setText(String.valueOf(dem));  }  if (this.index >= 0) {  cboSanPham.setSelectedItem(tblSanPham.getValueAt(index, 0));  } |
| btnGoiMon | this.insert();  this.setStatus();  this.clear();  this.tongTien(); |
| btnSuaMon | this.update();  this.tongTien(); |
| btnXaoMon | this.delete();  this.setStatus();  this.tongTien(); |
| btnClear | this.clear();  this.tongTien(); |
| txtSoLuongCaretUpdate | this.tinhTien(); |
| cboSanPham | this.tinhTien(); |
| btnThanhToan | try {  if (this.index >= 0) {  float tong = 0;  for (int i = 0; i < tblThongTin.getRowCount(); i++) {  tong += Float.parseFloat(tblThongTin.getValueAt(i, 5).toString());  }  if (DialogHelper.confirm(this, "Tổng cộng tiền cho bàn [ " + maBan + " ] là : " + tong + " vnđ")) {  banDAO.datBan(0, maBan);  hdDAO.updateTrangThaiHD(1, XDate.now(), daoCT.getMaHDByMaHDCT((int) tblThongTin.getValueAt(0, 0)), tong);  DanhMucJFrame.loadTabs();  DanhMucJFrame.loadDonHangTheoBan();  loadDonHangByBan(maBan);  }  }  } catch (Exception e) {  }  this.tongTien();  this.setStatus(); |
| txtFind | DefaultTableModel model = (DefaultTableModel) tblSanPham.getModel();  model.setRowCount(0);  try {  String keyWord = txtFind.getText();  List<SanPham> list = spDAO.selectByKeyword(keyWord);  for (SanPham sp : list) {  Object[] row = {  sp.getTenSanPham(),  lspDAO.findById(sp.getMaLoaiSP()).getTenLoaiSP(),  sp.getGiaBan(),  khDAO.getSLByMaSP(sp.getMaSanPham())  };  model.addRow(row);  }  } catch (Exception e) {  } |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.BanDAO;  import DAO.DanhMucDAO;  import DAO.HoaDonChiTietDAO;  import DAO.HoaDonDAO;  import DAO.KhoHangDAO;  import DAO.SanPhamDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import helper.XDate;  import java.util.List;  import javax.swing.DefaultComboBoxModel;  import javax.swing.JScrollPane;  import javax.swing.table.DefaultTableModel;  import model.HoaDon;  import model.HoaDonChiTiet;  import model.SanPham;  import static view.DanhMucJFrame.hdDAO;  import static view.DanhMucJFrame.loadDonHangTheoBan;  import static view.DanhMucJFrame.loadTabs;  import static view.DanhMucJFrame.lspDAO;  /\*\*  \*  \* @author duann  \*/  public class ThongTinDonHangJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form ThongTinDonHangJFrame  \*/  DanhMucDAO dmdao = new DanhMucDAO();  HoaDonDAO dao = new HoaDonDAO();  HoaDonChiTietDAO daoCT = new HoaDonChiTietDAO();  SanPhamDAO spDAO = new SanPhamDAO();  BanDAO banDAO = new BanDAO();  KhoHangDAO khDAO = new KhoHangDAO();  int index = 0;  int soLuong = 0;  int maSanPham;  int maBan;  int dem = 0;  int mahdct;  int maHD;  public ThongTinDonHangJFrame() {  initComponents();  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  this.setLocationRelativeTo(null);  }  public ThongTinDonHangJFrame(int id) {  initComponents();  loadDonHangByBan(id);  loadSanPham();  this.maBan = id;  lblThongTinCuaBan.setText("Thông tin của bàn: " + id);  tongTien();  fillComboBoxSanPham();  txtSoLuong.setText("0");  setStatus();  setBoderForTable(jScrollPane1);  setBoderForTable(jScrollPane2);  init();  this.setTitle("Hệ thống quản lý quán coffee");  }  void setBoderForTable(JScrollPane scp) {  scp.setViewportBorder(null);  scp.setBorder(null);  }  public void loadDonHangByBan(int id) {  DefaultTableModel model = (DefaultTableModel) tblThongTin.getModel();  model.setRowCount(0);  try {  List<Object[]> list = dmdao.getDanhSachSPTheoBan(id);  for (Object[] objects : list) {  model.addRow(objects);  }  } catch (Exception e) {  System.out.println(e.toString());  }  }  void setStatus() {  boolean insertable;  if (tblThongTin.getRowCount() > 0) {  insertable = false;  } else {  insertable = true;  }  btnSuaMon.setEnabled(!insertable);  btnThanhToan.setEnabled(!insertable);  btnXoaMon.setEnabled(!insertable);  }  void clear() {  cboSanPham.setSelectedIndex(0);  txtSoLuong.setText("0");  }  public void loadSanPham() {  DefaultTableModel model1 = (DefaultTableModel) tblSanPham.getModel();  model1.setRowCount(0);  try {  List<SanPham> list = spDAO.select();  for (SanPham sp : list) {  Object[] row = {  sp.getTenSanPham(),  lspDAO.findById(sp.getMaLoaiSP()).getTenLoaiSP(),  sp.getGiaBan(),  khDAO.getSLByMaSP(sp.getMaSanPham())  };  model1.addRow(row);  }  } catch (Exception e) {  }  }  void setModel(HoaDonChiTiet model) {  cboSanPham.setSelectedItem(spDAO.findById(model.getMaSanPham()).getTenSanPham());  txtSoLuong.setText(String.valueOf(model.getSoLuongSP()));  this.tinhTien();  }  void fillComboBoxSanPham() {  DefaultComboBoxModel model = (DefaultComboBoxModel) cboSanPham.getModel();  model.removeAllElements();  try {  List<SanPham> list = spDAO.select();  for (SanPham sp : list) {  model.addElement(sp.getTenSanPham());  }  } catch (Exception e) {  }  }  void tinhTien() {  try {  if (!txtSoLuong.getText().isEmpty()) {  String tenSP = (String) cboSanPham.getSelectedItem();  txtThanhTien.setText(String.valueOf(spDAO.findByName(tenSP).  getGiaBan() \* Integer.parseInt(txtSoLuong.getText())));  } else {  txtThanhTien.setText("");  }  } catch (Exception e) {  }  }  HoaDon getModelHD() {  HoaDon hd = new HoaDon();  hd.setMaBan(maBan);  hd.setMaNhanVien(1);  hd.setGhiChu("");  hd.setTrangThai(false); // false = 0  hd.setNgayThanhToan(XDate.now());  hd.setThanhTien(0);  return hd;  }  HoaDonChiTiet getModelHDCT() {  HoaDonChiTiet hdct = new HoaDonChiTiet();  hdct.setSoLuongSP(Integer.parseInt(txtSoLuong.getText()));  hdct.setMaSanPham(spDAO.findByName((String) cboSanPham.getSelectedItem()).getMaSanPham());  return hdct;  }  int insertHDWhenCallMon() {  HoaDon hd = getModelHD();  hdDAO.insert(hd);  return hdDAO.getIDIdentity();  }  boolean checkHang(int maKHmoi) {  int slspkho = khDAO.getSLByMaSP(spDAO.findByName((String) cboSanPham.getSelectedItem()).getMaSanPham());  if (slspkho > 0) {  if (slspkho >= Integer.parseInt(txtSoLuong.getText())) {  if (khDAO.getMaSLHang(maKHmoi) >= Integer.parseInt(txtSoLuong.getText())) {  return true;  } else {  DialogHelper.alert(this, "Số lượng sản phẩm trong kho hàng cũ chỉ còn : " + khDAO.getMaSLHang(maKHmoi) + ".");  return false;  }  } else {  DialogHelper.alert(this, "Số lượng trong kho chỉ còn : " + slspkho + ".");  return false;  }  } else {  DialogHelper.alert(this, "Sản phẩm tạm hết hàng!");  return false;  }  }  void insert() {  try {  // mã kho hàng lấy từ ngày thêm hàng sớm nhất  int maKhoHangMoi = khDAO.getMaKhoHangByMaSP(spDAO.findByName((String) cboSanPham.getSelectedItem()).getMaSanPham());  if (Integer.parseInt(txtSoLuong.getText()) > 0) {  if (checkHang(maKhoHangMoi)) {  if (tblThongTin.getRowCount() == 0) {  // nếu bảng rỗng thì chưa có hóa đơn => tạo mới hóa đơn rồi get mã hóa đơn vừa thêm gán vào =))  maHD = insertHDWhenCallMon();  } else {  // Tất cả những mã hóa đơn chi tiết đều map đến 1 hóa đơn  maHD = daoCT.getMaHDByMaHDCT((int) tblThongTin.getValueAt(0, 0));  }  HoaDonChiTiet model = getModelHDCT();  daoCT.insert(model, maHD);  // thêm thành công thì số lượng sản phẩm trong kho bị giảm  khDAO.updateSLByMaSP(Integer.parseInt(txtSoLuong.getText()), spDAO.findByName((String) cboSanPham.getSelectedItem()).getMaSanPham(), maKhoHangMoi);  // sửa lại trạng thái bàn sau khi gọi món  // 1 : trạng thái bàn đã được đặt chỗ.  banDAO.datBan(1, maBan);  lblTongTien.setText("Tổng thanh toán: " + dmdao.getThanhToanTheoBan(maBan) + " vnđ");  loadDonHangByBan(maBan);  loadSanPham();  loadTabs();  loadDonHangTheoBan();  DialogHelper.setInfinity(lblMSG, "Thêm mới thành công!");  }  } else {  DialogHelper.alert(this, "Vui lòng nhập số lượng.");  }  } catch (Exception e) {  }  }  public void update() {  try {  int maKhoHangMoi = khDAO.getMaKhoHangByMaSP(spDAO.findByName((String) cboSanPham.getSelectedItem()).getMaSanPham());  if (Integer.parseInt(txtSoLuong.getText()) > 0) {  if (checkHang(maKhoHangMoi)) {  HoaDonChiTiet hdct = getModelHDCT();  // mã hóa đơn chi tiết lấy được khi click vào bảng  daoCT.update(hdct, mahdct);  // mã sản phẩm được lấy khi click vào bảng hóa đơn ở trên  int maKhoHang = khDAO.getMaKhoHangByMaSP(maSanPham);  // trước khi sửa thì phải cộng lại số lượng đã mua rồi mới thêm hoặc bớt sau  khDAO.themSLByMaSP(soLuong, maSanPham, maKhoHang);  // sau khi cập nhật thì số lượng trong kho sẽ giảm đi txtSoluong lần  khDAO.updateSLByMaSP(Integer.parseInt(txtSoLuong.getText()), spDAO.findByName((String) cboSanPham.getSelectedItem()).getMaSanPham(), maKhoHangMoi);  lblTongTien.setText("Tổng thanh toán: " + dmdao.getThanhToanTheoBan(maBan) + " vnđ");  loadDonHangByBan(maBan);  loadSanPham();  loadTabs();  loadDonHangTheoBan();  this.clear();  DialogHelper.setInfinity(lblMSG, "Sửa thành công!");  }  } else {  DialogHelper.alert(this, "Vui lòng nhập số lượng.");  }  } catch (Exception e) {  }  }  void tongTien() {  float tong = 0;  for (int i = 0; i < tblThongTin.getRowCount(); i++) {  tong += Float.parseFloat(tblThongTin.getValueAt(i, 5).toString());  }  lblTongTien.setText("Tổng thanh toán: " + tong + " vnđ");  }  void delete() {  try {  // Mã hóa đơn lấy trước khi bị xóa để đảm bảo đủ dữ liệu  int mahd = daoCT.getMaHDByMaHDCT((int) tblThongTin.getValueAt(0, 0));  for (int i = 0; i < tblThongTin.getRowCount(); i++) {  Integer id = (Integer) tblThongTin.getValueAt(i, 0);  Integer masp = spDAO.findByName(tblThongTin.getValueAt(i, 1).toString()).getMaSanPham();  Integer sl = (Integer) tblThongTin.getValueAt(i, 4);  Boolean isDelete = (Boolean) tblThongTin.getValueAt(i, 6);  try {  if (isDelete) {  // Mã kho hàng lấy theo mã sản phẩm sắp xếp theo thời gian được thêm sản phẩm sớm nhất => để lấy hết sản phẩm cũ  int maKhoHang = khDAO.getMaKhoHangByMaSP(masp);  // Cộng lại số lượng hàng đã gọi trước khi xóa  khDAO.themSLByMaSP(sl, masp, maKhoHang);  daoCT.delete(id);  }  } catch (Exception e) {  }  }  loadDonHangByBan(maBan);  // Nếu xóa hết sản phẩm ở bảng thì bàn đó sẽ ở trạng thái chưa gọi hóa đơn và hóa đơn sẽ bị xóa  if (tblThongTin.getRowCount() == 0) {  hdDAO.delect(mahd);  banDAO.datBan(0, maBan);  }  loadDonHangTheoBan();  loadSanPham();  loadTabs();  tongTien();  clear();  } catch (Exception e) {  }  } |

#### Điểm danh nhân viên(DiemDanhNVJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| btnDiemDanh | DialogHelper.alert(this, this.diemDanh()); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.NhanVienDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import helper.XDate;  import java.util.List;  import javax.swing.table.DefaultTableModel;  import model.NhanVien;  /\*\*  \*  \* @author duann  \*/  public class DiemDanhNVJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form DiemDanhNVJFrame  \*/  NhanVienDAO nvdao = new NhanVienDAO();  public DiemDanhNVJFrame() {  initComponents();  init();  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  ShareHelper.setBoderForTable(jScrollPane1);  this.setTitle("Thông tin điểm danh ngày : " + XDate.toString(XDate.now()));  load();  checkDiemDanh();  }  void load() {  DefaultTableModel model = (DefaultTableModel) tblNhanVien.getModel();  model.setRowCount(0);  try {  boolean diemDanh = true;  List<NhanVien> list = nvdao.select();  for (NhanVien nhanVien : list) {  Object[] row = {  nhanVien.getMaNhanVien(),  nhanVien.getHoTen(),  nhanVien.getDienThoai(),  nhanVien.getCaLamViec().getTenCaLamViec(),  diemDanh  };  model.addRow(row);  }  } catch (Exception e) {  }  }  boolean checkDiemDanh() {  for (int j = 0; j < tblNhanVien.getRowCount(); j++) {  Integer manv = (Integer) tblNhanVien.getValueAt(j, 0);  NhanVien n = nvdao.findById(manv);  if (n.getSoNgayLamViec() >= Integer.parseInt(XDate.toString(XDate.now()).substring(0, 2))) {  btnDiemDanh.setVisible(false);  return true;  }  }  return false;  }  String diemDanh() {  if (checkDiemDanh()) {  return "Bạn đã điểm danh rồi!";  } else {  try {  for (int i = 0; i < tblNhanVien.getRowCount(); i++) {  Integer maNhanVien = (Integer) tblNhanVien.getValueAt(i, 0);  Boolean isDiemDanh = (Boolean) tblNhanVien.getValueAt(i, 4);  NhanVien nv = nvdao.findById(maNhanVien);  NhanVien model = new NhanVien();  model.setMaNhanVien(maNhanVien);  if (isDiemDanh == true) {  model.setSoNgayLamViec(nv.getSoNgayLamViec() + 1);  } else {  model.setSoNgayLamViec(nv.getSoNgayLamViec());  }  nvdao.diemDanh(model);  }  } catch (Exception e) {  System.out.println(e.toString());  }  }  return "Điểm danh thành công!";  } |

#### Quản kho hàng(KhoHangJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| tblQuanLyKhoHang.mouseClicked | if (evt.getClickCount() == 1) {  this.index = tblQuanLyKhoHang.rowAtPoint(evt.getPoint());  if (this.index >= 0) {  tabs.setSelectedIndex(0);  this.fillToForm();  }  } |
| btnThem | insert(); |
| btnSua | update(); |
| btnXao | delete(); |
| btnMoi | clear(); |
| btnFirst | this.index = 0;  this.edit(); |
| btnPrev | this.index--;  this.edit(); |
| btnNext | this.index++;  this.edit(); |
| btnLast | this.index = tblQuanLyKhoHang.getRowCount() - 1; this.edit(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.KhoHangDAO;  import DAO.NhanVienDAO;  import DAO.SanPhamDAO;  import helper.DialogHelper;  import helper.ShareHelper;  import helper.XDate;  import java.util.List;  import javax.swing.DefaultComboBoxModel;  import javax.swing.table.DefaultTableModel;  import model.KhoHang;  import model.SanPham;  /\*\*  \*  \* @author Admin  \*/  public class KhoHangJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form quanlykhohang  \*/  int index = 0;  KhoHangDAO dao = new KhoHangDAO();  SanPhamDAO spdao = new SanPhamDAO();  NhanVienDAO nvdao = new NhanVienDAO();  public KhoHangJFrame() {  initComponents();  init();  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  ShareHelper.setBoderForTable(jScrollPane2);  this.setTitle("Hệ thống quản lý quán coffee");  load();  fillCombobox();  load();  setLocationRelativeTo(null);  setIconImage(ShareHelper.APP\_ICON);  }  void fillCombobox() {  DefaultComboBoxModel model = (DefaultComboBoxModel) cboSanPham.getModel();  model.removeAllElements();  try {  List<SanPham> list = spdao.select();  for (SanPham sanPham : list) {  model.addElement(sanPham.getTenSanPham());  }  } catch (Exception e) {  }  }  void load() {  DefaultTableModel model = (DefaultTableModel) tblQuanLyKhoHang.getModel();  model.setRowCount(0);  try {  List<KhoHang> list = dao.select();  for (KhoHang kh : list) {  Object[] row = {  kh.getMaKhoHang(),  spdao.findById(kh.getMaSanPham()).getTenSanPham(),  nvdao.findById(kh.getMaNhanVien()).getHoTen(),  kh.getNgayNhap(),  kh.getHanSuDung(),  kh.getSoLuong()  };  model.addRow(row);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");  }  }  void insert() {  KhoHang model = getModel();  try {  dao.insert(model);  this.load();  this.clear();  DialogHelper.alert(this, "Thêm mới thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Thêm mới thất bại!");  System.out.println(e.toString());  }  }  void update() {  KhoHang model = getModel();  try {  dao.update(model);  this.load();  this.clear();  DialogHelper.alert(this, "Cập nhập thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Cập nhập thất bại!");  }  }  void delete() {  if (DialogHelper.confirm(this, "Bạn thực sự muôn xóa Kho Hàng này!")) {  Integer makh = Integer.valueOf(cboSanPham.getToolTipText());  try {  dao.delete(makh);  this.load();  this.clear();  DialogHelper.alert(this, "Xóa thành công!");  } catch (Exception e) {  DialogHelper.alert(this, "Xóa thất bại!");  }  }  }  void clear() {  KhoHang model = new KhoHang();  String sanpham = (String) cboSanPham.getSelectedItem();  SanPham sp = spdao.findByName(sanpham);  model.setSoLuong(model.getSoLuong());  model.setGhiChu(model.getGhiChu());  this.cboSanPham.setSelectedItem(0);  this.setModel(model);  this.setStatus(true);  }  void edit() {  try {  int makh = (int) tblQuanLyKhoHang.getValueAt(this.index, 0);  KhoHang model = dao.findById(makh);  if (model != null) {  this.setModel(model);  this.setStatus(false);  }  } catch (Exception e) {  DialogHelper.alert(this, "Lỗi truy vấn dữ liệu");  }  }  void setModel(KhoHang model) {  cboSanPham.setToolTipText(String.valueOf(model.getMaKhoHang()));  SanPham sp = spdao.findById(model.getMaSanPham());  try {  cboSanPham.setSelectedItem(sp.getTenSanPham());  } catch (Exception e) {  }  txtSoLuong.setText(String.valueOf(model.getSoLuong()));  txtGhiChu.setText(model.getGhiChu());  }  KhoHang getModel() {  KhoHang khoHang = new KhoHang();  String tenSanPham = (String) cboSanPham.getSelectedItem();  SanPham sp = spdao.findByName(tenSanPham);  if (txtSoLuong.getToolTipText() != null) {  khoHang.setMaKhoHang(Integer.valueOf(cboSanPham.getToolTipText()));  }  khoHang.setMaSanPham(sp.getMaSanPham());  khoHang.setMaNhanVien(ShareHelper.USER.getMaNhanVien());  khoHang.setNgayNhap(XDate.now());  khoHang.setSoLuong(Integer.valueOf(txtSoLuong.getText()));  khoHang.setGhiChu(txtGhiChu.getText());  khoHang.setHanSuDung(XDate.addDays(XDate.now(), 30)); // hạn sử dụng 1 tháng kể từ ngày thêm  return khoHang;  }  void setStatus(boolean insertable) {  btnthem.setEnabled(insertable);  btnsua.setEnabled(!insertable);  btnxoa.setEnabled(!insertable);  boolean first = this.index > 0;  boolean last = this.index < tblQuanLyKhoHang.getRowCount() - 1;  btnFirst.setEnabled(!insertable && first);  btnPrev.setEnabled(!insertable && first);  btnLast.setEnabled(!insertable && last);  btnNext.setEnabled(!insertable && last);  } |

#### Hoa đơn chiết tiết(HoaDonChiTietJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.HoaDonChiTietDAO;  import DAO.HoaDonDAO;  import DAO.NhanVienDAO;  import helper.ShareHelper;  import helper.XDate;  import java.util.List;  import javax.swing.table.DefaultTableModel;  import model.HoaDon;  /\*\*  \*  \* @author duann  \*/  public class HoaDonChiTietJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form HoaDonChiTietJFrame  \*/  static HoaDonDAO dao = new HoaDonDAO();  static HoaDonChiTietDAO daoCT = new HoaDonChiTietDAO();  static NhanVienDAO nvdao = new NhanVienDAO();  public HoaDonChiTietJFrame() {  initComponents();  init();  }  void init() {  this.setLocationRelativeTo(null);  load();  setIconImage(ShareHelper.APP\_ICON);  this.setTitle("Hệ thống quản lý quán coffee");  ShareHelper.setBoderForTable(jScrollPane2);  }  public static void load() {  DefaultTableModel model = (DefaultTableModel) tblHoaDon.getModel();  model.setRowCount(0);  try {  List<HoaDon> list = dao.getAll();  for (HoaDon hoaDon : list) {  Object[] row = {  hoaDon.getMaBan(),  hoaDon.getMaHoaDon(),  hoaDon.getSanPham().getTenSanPham(),  hoaDon.getHoaDonChiTiet().getSoLuongSP(),  hoaDon.getSanPham().getGiaBan(),  hoaDon.isTrangThai() ? "Đã thanh toán" : "Chưa thanh toán",  nvdao.findById(hoaDon.getMaNhanVien()).getHoTen(),  hoaDon.isTrangThai() ? XDate.toString(hoaDon.getNgayThanhToan()) : "",};  model.addRow(row);  }  } catch (Exception e) {  }  } |

#### Thống kê dữ liệu(ThongKeJFrame)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |
| cboThoiGian | fillTableDoanhThu(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import DAO.SanPhamDAO;  import DAO.ThongKeDAO;  import helper.ShareHelper;  import java.util.ArrayList;  import java.util.List;  import javax.swing.DefaultComboBoxModel;  import javax.swing.table.DefaultTableModel;  /\*\*  \*  \* @author NgọcHải  \*/  public class ThongKeJFrame extends javax.swing.JFrame {  /\*\*  \* Creates new form ThongKeJFrame  \*/  ThongKeDAO dao = new ThongKeDAO();  SanPhamDAO spDAO = new SanPhamDAO();  public ThongKeJFrame() {  initComponents();  init();  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  fillComboBoxThoiGian();  fillTableDoanhThu();  fillTableKhoHang();  }  void fillTableKhoHang() {  DefaultTableModel model = (DefaultTableModel) tblThongKeKhoHang.getModel();  model.setRowCount(0);  try {  List<Object[]> list = dao.getKhoHang();  for (Object[] objects : list) {  objects[0] = spDAO.findById((int) objects[0]).getTenSanPham();  model.addRow(objects);  }  } catch (Exception e) {  }  }  void fillComboBoxThoiGian() {  DefaultComboBoxModel model = (DefaultComboBoxModel) cboThoiGian.getModel();  model.removeAllElements();  try {  List<String> list = new ArrayList<>();  list.add("Năm");  list.add("Ngày");  for (String thoiGian : list) {  model.addElement(thoiGian);  }  } catch (Exception e) {  }  }  void fillTableDoanhThu() {  DefaultTableModel model = (DefaultTableModel) tblThongKeTheoThang.getModel();  model.setRowCount(0);  try {  if (cboThoiGian.getSelectedItem().equals("Năm")) {  List<Object[]> list = dao.getDoanhThuNam();  for (Object[] objects : list) {  model.addRow(objects);  }  } else {  List<Object[]> list = dao.getDoanhThuThang();  for (Object[] objects : list) {  model.addRow(objects);  }  }  } catch (Exception e) {  }  } |

#### Giới thiệu(GioiThieuJDialog)

Mô tả hoạt động.

|  |  |
| --- | --- |
| Sự kiện | Công việc |
| Constructor | Init(); |

HƯỚNG DẪN CÀI ĐẶT MÃ NGUỒN

|  |
| --- |
| package view;  import helper.ShareHelper;  public class GioiThieuJDialog extends javax.swing.JDialog {  public GioiThieuJDialog(java.awt.Frame parent, boolean modal) {  super(parent, modal);  initComponents();  init();  sp.setViewportBorder(null);  sp.setBorder(null);  }  void init() {  setIconImage(ShareHelper.APP\_ICON);  setLocationRelativeTo(null);  } |

# Kiểm thử

## Kiểm thử form X

## Kiểm thử form Y

# Đóng gói và triển khai

## Hướng dẫn chuyển đổi jar thành exe

## Hướng dẫn cài đặt triển khai

## Hướng dẫn sử dụng phần mềm