TRƯỜNG ĐẠI HỌC NAM CẦN THƠ

**TRUNG TÂM ĐÀO TẠO CHUẨN ĐẦU RA**

**VÀ PHÁT TRIỂN NGUỒN NHÂN LỰC**

****

**NGUYỄN ĐÌNH NGUYÊN**

**QUÁCH HOÀNG LUÂN**

ĐỀ TÀI

**PHẦN MỀM QUẢN LÝ COFFEE**

**BÁO CÁO KỸ NĂNG NGHỀ NGHIỆP**

**Cần Thơ - 2022**

TRƯỜNG ĐẠI HỌC NAM CẦN THƠ

**TRUNG TÂM ĐÀO TẠO CHUẨN ĐẦU RA**

**VÀ PHÁT TRIỂN NGUỒN NHÂN LỰC**

****

**NGUYỄN ĐÌNH NGUYÊN**

**180886**

**QUÁCH HOÀNG LUÂN**

**188525**

**KHÓA 06**

ĐỀ TÀI

**PHẦN MỀM QUẢN LÝ COFFEE**

**BÁO CÁO KỸ NĂNG NGHỀ NGHIỆP**

**Cần Thơ - 2022**

# MỤC LỤC

[MỤC LỤC 3](#_Toc107865625)

[DANH MỤC BẢNG 4](#_Toc107865626)

[PHẦN MỀM QUẢN LÝ COFFEE 2](#_Toc107865627)

[1. Mục đích của phần mềm 2](#_Toc107865628)

[2. Chức năng chính của phần mềm 2](#_Toc107865629)

[3. Thiết kế cơ sở dữ liệu 2](#_Toc107865630)

[3.1. Cơ sở dữ liệu 2](#_Toc107865631)

[3.2. Thiết kế các lớp 4](#_Toc107865632)

[4. Xây dựng các lớp trong chương trình 5](#_Toc107865633)

[4.1. Lớp tài khoản 5](#_Toc107865634)

[4.1.1 Controllers – Models 5](#_Toc107865635)

[4.1.2 Event 8](#_Toc107865636)

[4.2. Lớp bàn 9](#_Toc107865637)

[4.2.1 Controllers – Models 9](#_Toc107865638)

[4.2.2 Events 12](#_Toc107865639)

[4.3 Lớp loại thức uống 13](#_Toc107865640)

[4.3.1 Controllers – Models 13](#_Toc107865641)

[4.3.2 Events 16](#_Toc107865642)

[4.4 Thức uống 17](#_Toc107865643)

[4.4.1 Controllers – Models 17](#_Toc107865644)

[4.4.2 Events 20](#_Toc107865645)

[TÀI LIỆU THAM KHẢO 23](#_Toc107865646)

# DANH MỤC BẢNG

[Bảng 3. 1 Thực thể Tài Khoản 2](#_Toc107865924)

[Bảng 3. 2 Thực thể loại thức uống 3](#_Toc107865925)

[Bảng 3. 3 Thực thể Hoá Đơn 3](#_Toc107865926)

[Bảng 3. 4 Thực thể Chi Tiết Hoá Đơn 3](#_Toc107865927)

**DANH MỤC HÌNH**

[Hình 1 Mô hình thiết kế 4](#_Toc107865647)

# LỜI NÓI ĐẦU

Hiện nay, với sự phát triển ngày càng gia tăng không ngừng của xã hộ hiện đại, tin học đã trở thành một thứ không thể thiếu và nó đã phổ biến những bước không ngừng vượt bậc được thể qua nhiều lĩnh vực trong đời sống. Tin học đã dần trở thành nhu cầu cần thiết ở mỗi quốc gia trên thế giới chúng không thể thiếu trong cuộc sống của con người. Hầu như các quốc gia phát triển do tin học hoá đem lại đều có phần thiết yếu và chiếm giữ một vị trí quan trọng trong các lĩnh vực kinh tế, chính trị, văn hoá, giáo dục.

Trước đây, nếu muốn mua một đồ thức uống thì phải vào quán coffee hoặc các cửa tiệm và phải đi tìm kiếm như thế rất nhiều thời gian và tiềm rất lâu có khi đến nơi thì cửa tiệm lại đóng cửa. Và để đáp ứng nhu cầu đó của con người thì ngày nay việc kinh doanh quán coffee không còn là quá xa lạ với mọi người. Các quán cà phê mọc lên như một xu hướng, đủ thể loại, đủ hình thức thu hút khách hàng. Ngoài ra, thức uống cũng là một phần quan trọng dẫn đến lượng khách đến quán nhiều hay ít.

Lâu nay cà phê là thức uống dành cho người đi làm, dân công sở nhưng dạo gần đây thì cà phê được giới trẻ yêu thích không kém. Sự phát triển du nhập của cà phê đã hình thành một sở thích uống cà phê mới trong giới trẻ. Và để đáp ứng nhu cầu khi các bạn trẻ hoặc người đi làm muốn dùng đồ uống nhanh không cần phải chờ đợi thì nhóm tụi em đã cùng nhau thực hiện phần mềm này nhằm được tạo ra để hy vọng mang đến cho người dùng sự tiện lợi cũng như giúp quán có thể xem xét tổng quát tình hình kinh doanh quán của mình.

Phần mềm còn mang lại lợi ích kinh tế khi là giải pháp giúp việc quản lý trở nên đơn giản và thân thiện với người dùng và khách hàng.

**Nhóm sinh viên thực hiện**

Nguyễn Đình Nguyên

Quách Hoàng Luân

# PHẦN MỀM QUẢN LÝ COFFEE

## Mục đích của phần mềm

Nghiên cứu tổng quan về phần mềm quản lý coffee hổ trợ việc quản lý các nhiệm vụ như tính tiền, in hoá đơn, thống kê, quản lý nhân viên,…v.v. Phần mềm quản lý coffee giúp cho mọi người sử dụng tiết kiệm cho thời gian, nhanh chóng, thao tác đơn giản, dễ dàng trong việc sử dụng và quản lý.

Nghiên cứu các công cụ, nền tảng lập trình được sử dụng để xây dựng phần mềm: Eclipse, MySQL, Powdesigner, và ngôn ngữ java. Cài đặt triển khai phần mềm trên nền tảng App desktop và hoàn thiện các tính năng theo yêu cầu của đề tài đặt ra.

## Chức năng chính của phần mềm

* Quản lý nhân viên.
* Quản lý đồ uống.
* Quản lý các bàn gọi đồ uống.
* Quản lý các loại đồ uống.

## 3. Thiết kế cơ sở dữ liệu

### 3.1. Cơ sở dữ liệu

Bảng 3. 1 Thực thể Tài Khoản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Khoá | Giải thích |
| MaTK | int |  | Khoá chính | Mã tài khoản |
| Username | Varchar | 32 | Khoá chính | Tên tài khoản |
| Password | Varchar | 32 |  | Tên mật khẩu |
| HoTen | Text |  |  | Họ và tên |

Bảng 3. 2 Thực thể loại thức uống

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Khoá | Giải thích |
| MaLoai | int |  | Khoá chính | Mã loại |
| TenLoai | Text |  |  | Tên loại |

Bảng 3. 3 Thực thể Hoá Đơn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Khoá | Giải thích |
| MaHD | int |  | Khoá chính | Mã hoá đơn |
| MaBan | int |  | Khoá chính | Mã bàn |
| TongTien | int |  |  | Tổng tiền |
| GioDen | datetime |  |  | Giờ đến |

Bảng 3. 4 Thực thể Chi Tiết Hoá Đơn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Khoá | Giải thích |
| MaHD | int |  | Khóa Chính | Mã hoá đơn |
| MaTA | int |  | Khoá chính | Mã thức ăn |
| Gia | int |  |  | Giá |
| SoLuong | int |  |  | Số lượng |

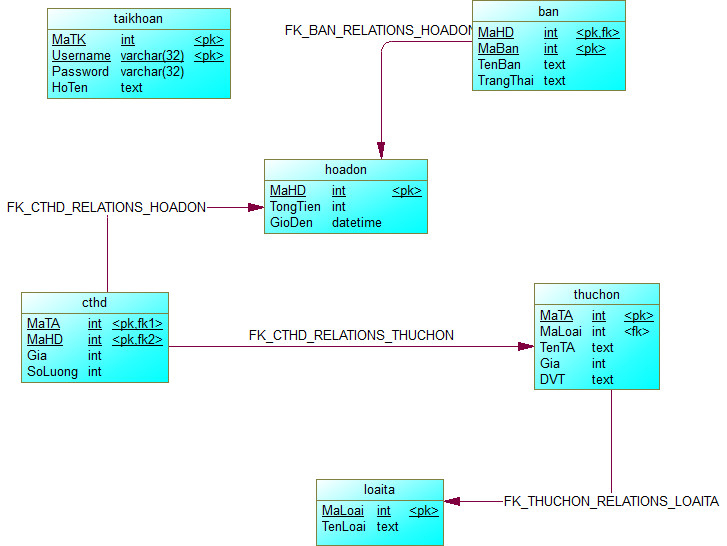
Bảng 3. 5 Thực thể Chi Tiếc Thực Đơn

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Khoá | Giải thích |
| MaTA | int |  | Khóa Chính | Mã thức ăn |
| TenTA | int |  |  | Tên thức ăn |
| Gia | int |  |  | Số lượng |
| DVT | Text |  |  | Đơn vị tính |
| MaLoai | int |  |  | Thông số |

*Bảng 3. 6 Thực thể Bàn*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Thuộc tính | Kiểu dữ liệu | Kích thước | Khoá | Giải thích |
| MaBan | int |  | Khóa Chính | Mã bàn |
| TenBan | int |  |  | Tên bàn |
| TrangThai | int |  |  | Trạng Thái |

### 3.2. Thiết kế các lớp



Hình 1 Mô hình thiết kế

## 4. Xây dựng các lớp trong chương trình

## 4.1. Lớp tài khoản

### 4.1.1 Controllers – Models

**public** **class** TaiKhoan {

// constructor

**private** **int** maTK;

**private** String userName, passWord, Hoten;

**private** **static** Connection *con* = MyConnect.*con*;

**public** TaiKhoan(**int** maTK, String userName, String passWord, String hoten) {

**super**();

**this**.maTK = maTK;

**this**.userName = userName;

**this**.passWord = passWord;

Hoten = hoten;

}

**public** TaiKhoan() {

**super**();

}

**public** **int** getMaTK() {

**return** maTK;

}

**public** **void** setMaTK(**int** maTK) {

**this**.maTK = maTK;

}

**public** String getUserName() {

**return** userName;

}

**public** **void** setUserName(String userName) {

**this**.userName = userName;

}

**public** String getPassWord() {

**return** passWord;

}

**public** **void** setPassWord(String passWord) {

**this**.passWord = passWord;

}

**public** String getHoten() {

**return** Hoten;

}

**public** **void** setHoten(String hoten) {

Hoten = hoten;

}

// models

**public** **void** Insert(TaiKhoan tk) {

**try** {

String sql = String.*format*("insert into taikhoan (username,password, hoten) values ('%s', '%s','%s')",

tk.getUserName(), tk.getPassWord(), tk.getHoten());

PreparedStatement s = *con*.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showInternalMessageDialog*(**null**, "Thêm thành công");

} **else** {

JOptionPane.*showInternalMessageDialog*(**null**, "Thêm thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** DoiMatKhau(TaiKhoan tk) {

**try** {

String sql = String.*format*("update taikhoan password='%s',hoten='%s' where matk=%d", tk.getPassWord(),

tk.getHoten(), tk.getMaTK());

PreparedStatement s = *con*.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showInternalMessageDialog*(**null**, "Đổi thông tin thành công");

} **else** {

JOptionPane.*showInternalMessageDialog*(**null**, "Đổi thông tin thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** Update(TaiKhoan tk) {

**try** {

String sql = String.*format*("update taikhoan set username ='%s',password='%s',hoten='%s' where matk=%d",

tk.getUserName(), tk.getPassWord(), tk.getHoten(), tk.getMaTK());

PreparedStatement s = *con*.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showInternalMessageDialog*(**null**, "Cập nhật tài khoản thành công");

} **else** {

JOptionPane.*showInternalMessageDialog*(**null**, "Cập nhật tài khoản thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** Delete(TaiKhoan tk) {

**try** {

String sql = String.*format*("delete from taikhoan where matk=%d", tk.getMaTK());

PreparedStatement s = *con*.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showInternalMessageDialog*(**null**, "Xoá tài khoản thành công");

} **else** {

JOptionPane.*showInternalMessageDialog*(**null**, "Xoá tài khoản thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **boolean** CheckLogin(TaiKhoan tk) {

**boolean** check = **false**;

String sql = String.*format*("select \* from taikhoan where username='%s' and password='%s'", tk.getUserName(),

tk.getPassWord());

**try** {

Statement st = *con*.createStatement();

ResultSet rs = st.executeQuery(sql);

**while** (rs.next()) {

check = **true**;

}

} **catch** (SQLException ex) {

JOptionPane.*showMessageDialog*(**null**, "Đã xảy ra lỗi khi đăng nhập !");

}

**return** check;

}

**public** TaiKhoan GetTaiKhoan(String name, String pass) {

TaiKhoan tk = **null**;

String sql = String.*format*("select \* from taikhoan where username='%s' and password='%s'", name, pass);

**try** {

Statement st = *con*.createStatement();

ResultSet rs = st.executeQuery(sql);

**while** (rs.next()) {

tk = **new** TaiKhoan(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4));

}

} **catch** (SQLException ex) {

JOptionPane.*showMessageDialog*(**null**, "lỗi !");

}

**return** tk;

}

**public** ArrayList<TaiKhoan> ShowTable(**int** matk) {

ArrayList<TaiKhoan> arrTK = **null**;

String sql;

**if** (matk == 0)

sql = "Select \* From taikhoan";

**else**

sql = "Select \* From taikhoan Where matk = " + matk;

**try** {

Statement st = *con*.createStatement();

ResultSet rs = st.executeQuery(sql);

arrTK = **new** ArrayList<TaiKhoan>();

**while** (rs.next()) {

TaiKhoan tk = **new** TaiKhoan(rs.getInt(1), rs.getString(2), rs.getString(3), rs.getString(4));

arrTK.add(tk);

}

} **catch** (SQLException ex) {

JOptionPane.*showMessageDialog*(**null**, "Đã xảy ra lỗi !");

}

**return** arrTK;

}

}

### 4.1.2 Event

**private** **void** addevents() {

btnDangNhap.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

String user = txtUser.getText();

String pass = txtPass.getText();

tk.setUserName(user);

tk.setPassWord(pass);

**boolean** check = tk.CheckLogin(tk);

**if** (check) {

Run.*tk* = tk.GetTaiKhoan(user, pass);

*name* = Run.*tk*.getHoten();

*pasS* = Run.*tk*.getPassWord();

*matk*= Run.*tk*.getMaTK();

Run.*runMain*();

setVisible(**false**);

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Tài khoản hoặc mật khẩu không đúng", "Login Error", JOptionPane.***ERROR\_MESSAGE***);

}

}

});

btnThoat.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

System.*exit*(0);

}

});

}

## 4.2. Lớp bàn

### 4.2.1 Controllers – Models

**public** **class** Ban {

**private** **int** maBan;

**private** String tenBan, trangThai;

**public** Ban(**int** maBan, String tenBan, String trangThai) {

**super**();

**this**.maBan = maBan;

**this**.tenBan = tenBan;

**this**.trangThai = trangThai;

}

**public** Ban() {

**super**();

}

**public** **int** getMaBan() {

**return** maBan;

}

**public** **void** setMaBan(**int** maBan) {

**this**.maBan = maBan;

}

**public** String getTenBan() {

**return** tenBan;

}

**public** **void** setTenBan(String tenBan) {

**this**.tenBan = tenBan;

}

**public** String getTrangThai() {

**return** trangThai;

}

**public** **void** setTrangThai(String trangThai) {

**this**.trangThai = trangThai;

}

@Override

**public** String toString() {

**return** **this**.tenBan;

}

// xu ly

**private** Connection con = MyConnect.*con*;

**public** **void** Insert(Ban b) {

**try** {

String sql = String.*format*("insert into ban (tenban,trangthai) values ('%s', '%s')", b.getTenBan(),

b.getTrangThai());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showInternalMessageDialog*(**null**, "Thêm thành công");

} **else** {

JOptionPane.*showInternalMessageDialog*(**null**, "Thêm thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** UpdateBan(Ban b) {

**try** {

String sql = String.*format*("update ban set tenban='%s' where maban=%d", b.getTenBan(), b.getMaBan());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showInternalMessageDialog*(**null**, "Cập nhật bàn thành công");

} **else** {

JOptionPane.*showInternalMessageDialog*(**null**, "Cập nhật bàn thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** UpdateTrangThaiBan(Ban b) {

**try** {

String sql = String.*format*("update ban set trangthai='%s' where maban=%d", b.getTrangThai(), b.getMaBan());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showInternalMessageDialog*(**null**, "Cập nhật trạng thái bàn thành công");

} **else** {

JOptionPane.*showInternalMessageDialog*(**null**, "Cập nhật trạng thái bàn thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** Delete(Ban b) {

**try** {

String sql = String.*format*("delete form ban where maban= %d)", b.getMaBan());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showConfirmDialog*(**null**, "Xoá bàn thành công");

} **else** {

JOptionPane.*showConfirmDialog*(**null**, "Xoá bàn thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** ArrayList<Ban> ShowBan(**int** maban) {

ArrayList<Ban> arrBan = **null**;

String sql;

**if** (maban == 0)

sql = "Select \* From ban";

**else**

sql = "Select \* From ban Where MaBan = " + maban;

**try** {

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(sql);

arrBan = **new** ArrayList<Ban>();

**while** (rs.next()) {

Ban b = **new** Ban(rs.getInt(1), rs.getString(2), rs.getString(3));

arrBan.add(b);

}

} **catch** (SQLException ex) {

JOptionPane.*showMessageDialog*(**null**, "Đã xảy ra lỗi !");

}

**return** arrBan;

}

}

### 4.2.2 Events

**private** **void** addevents() {

*btnThem*.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

*In\_UP* = *btnThem*.getText().toUpperCase();

JD\_QL\_Ban\_Edit ban = **new** JD\_QL\_Ban\_Edit(Run.*main*, **true**, "Thêm bàn mới");

ban.setVisible(**true**);

FillTable();

}

});

*btnSua*.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**if** (*maban* > 0) {

*In\_UP* = *btnSua*.getText().toUpperCase();

JD\_QL\_Ban\_Edit ban = **new** JD\_QL\_Ban\_Edit(Run.*main*, **true**, "Sửa bàn - Bàn ");

ban.setVisible(**true**);

FillTable();

} **else** {

JOptionPane.*showInternalMessageDialog*(**null**, "Bạn chưa chọn!");

}

}

});

btnXoa.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

b.setMaBan(*maban*);

b.Delete(b);

FillTable();

}

});

tabBan.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

ListSelectionModel model = tabBan.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

@Override

**public** **void** valueChanged(ListSelectionEvent e) {

**if** (e.getValueIsAdjusting()) {

**return**;

}

ListSelectionModel lsm = (ListSelectionModel) e.getSource();

**if** (lsm.isSelectionEmpty()) {

} **else** {

**int** selectedRow = lsm.getMinSelectionIndex();

*maban* = Integer.*parseInt*(tabBan.getModel().getValueAt(selectedRow, 0).toString());

}

}

});

}

**public** **void** FillTable() {

ArrayList<Ban> arrTable = b.ShowBan(0);

DefaultTableModel tbmodel = **new** DefaultTableModel();

tbmodel.addColumn("Mã Bàn");

tbmodel.addColumn("Tên bàn");

tbmodel.addColumn("Trạng thái");

**if** (arrTable != **null**) {

**int** soban = 0;

**for** (Ban b : arrTable) {

soban++;

tbmodel.addRow(**new** Object[] { b.getMaBan(), b.getTenBan(), b.getTrangThai() });

}

lblTotal.setText(String.*valueOf*(soban) + " bàn");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Không có bàn nào");

}

tabBan.setModel(tbmodel);

**for** (**int** i = 0; i < tabBan.getColumnCount(); i++) {

Class<?> col = tabBan.getColumnClass(i);

tabBan.setDefaultEditor(col, **null**);

}

}

## 4.3 Lớp loại thức uống

### 4.3.1 Controllers – Models

**public** **class** LoaiThucAn {

// tạo constructor

**private** **int** maLoai;

**private** String tenLoai;

**public** LoaiThucAn(**int** maLoai, String tenLoai) {

**super**();

**this**.maLoai = maLoai;

**this**.tenLoai = tenLoai;

}

**public** LoaiThucAn() {

**super**();

}

**public** **int** getMaLoai() {

**return** maLoai;

}

**public** **void** setMaLoai(**int** maLoai) {

**this**.maLoai = maLoai;

}

**public** String getTenLoai() {

**return** tenLoai;

}

**public** **void** setTenLoai(String tenLoai) {

**this**.tenLoai = tenLoai;

}

@Override

**public** String toString() {

**return** **this**.tenLoai;

}

//xu ly dư liệu

**private** Connection con = MyConnect.*con*;

**public** **void** Insert(LoaiThucAn l) {

**try** {

String sql = String.*format*("insert into loaita(tenloai) values ('%s')", l.getTenLoai());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showMessageDialog*(**null**,"Thêm thành công");

} **else** {

JOptionPane.*showMessageDialog*(**null**,"Thêm thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** Update(LoaiThucAn l) {

**try** {

String sql = String.*format*("update loaita set tenloai='%s' where maloai=%d", l.getTenLoai(), l.getMaLoai());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showMessageDialog*(**null**,"Cập nhật thành công");

} **else** {

JOptionPane.*showMessageDialog*(**null**,"Cập nhật thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** Delete(LoaiThucAn l) {

**try** {

String sql = String.*format*("delete from loaita where maloai=%d", l.getMaLoai());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showMessageDialog*(**null**,"Xoá thành công");

} **else** {

JOptionPane.*showMessageDialog*(**null**,"Xoá thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** ArrayList<LoaiThucAn> ShowTable(String key) {

ArrayList<LoaiThucAn> arrloai = **null**;

String sql;

**if** (Integer.*parseInt*(key) == 0) {

sql = "select \* from loaita";

} **else** {

sql = String.*format*("select \* from loaita where maloai= %d or tenloai like '%s'", Integer.*parseInt*(key),

'%' + key + '%');

}

**try** {

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(sql);

arrloai = **new** ArrayList<LoaiThucAn>();

**while** (rs.next()) {

LoaiThucAn lt = **new** LoaiThucAn(rs.getInt(1), rs.getString(2));

arrloai.add(lt);

}

} **catch** (SQLException ex) {

JOptionPane.*showMessageDialog*(**null**, "Đã xảy ra lỗi !");

}

**return** arrloai;

}

**public** **void** ShowLoai\_InCBB(JComboBox cbb) {

**try** {

DefaultComboBoxModel<LoaiThucAn> model = **new** DefaultComboBoxModel<>();

String sql = "select \* from loaita";

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(sql);

**while** (rs.next()) {

model.addElement(**new** LoaiThucAn(rs.getInt("MaLoai"), rs.getString("TenLoai")));

}

cbb.setModel(model);

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

}

### 4.3.2 Events

**public** **void** FillTable() {

ArrayList<LoaiThucAn> arrTable = l.ShowTable("0");

DefaultTableModel tbmodel = **new** DefaultTableModel();

tbmodel.addColumn("Mã Loại");

tbmodel.addColumn("Tên Loại");

**if** (arrTable != **null**) {

**int** soban = 0;

**for** (LoaiThucAn l : arrTable) {

soban++;

tbmodel.addRow(**new** Object[] { l.getMaLoai(), l.getTenLoai() });

}

lblTotal.setText(String.*valueOf*(soban) + " loại");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Không có bàn nào");

}

tabLoai.setModel(tbmodel);

**for** (**int** i = 0; i < tabLoai.getColumnCount(); i++) {

Class<?> col = tabLoai.getColumnClass(i);

tabLoai.setDefaultEditor(col, **null**);

}

}

**private** **void** addevents() {

btnThem.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

*name* = btnThem.getText();

JD\_QL\_NhomMon\_Edit ban = **new** JD\_QL\_NhomMon\_Edit(Run.*main*, **true**, "Thêm loại món");

ban.setVisible(**true**);

FillTable();

}

});

btnSua.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

*name* = btnSua.getText();

**if**(*maloai*>0) {

JD\_QL\_NhomMon\_Edit ban = **new** JD\_QL\_NhomMon\_Edit(Run.*main*, **true**, "Sửa loại món");

ban.setVisible(**true**);

FillTable();

}

**else** {

JOptionPane.*showMessageDialog*(**null**, "Bạn chưa chọn");

}

}

});

btnXoa.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

FillTable();

}

});

tabLoai.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

ListSelectionModel model = tabLoai.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

@Override

**public** **void** valueChanged(ListSelectionEvent e) {

**if** (e.getValueIsAdjusting()) {

**return**;

}

ListSelectionModel lsm = (ListSelectionModel) e.getSource();

**if** (lsm.isSelectionEmpty()) {

} **else** {

**int** selectedRow = lsm.getMinSelectionIndex();

*maloai* = Integer.*parseInt*(tabLoai.getModel().getValueAt(selectedRow, 0).toString());

}

}

});

}

## 4.4 Thức uống

### 4.4.1 Controllers – Models

**public** **class** ThucAn {

// Constractor

**private** **int** maTA;

**private** String tenTA;

**private** **int** gia;

**private** String dVT;

**private** String maLoai;

**public** ThucAn(**int** maTA, String tenTA, **int** gia, String dVT, String maLoai) {

**super**();

**this**.maTA = maTA;

**this**.tenTA = tenTA;

**this**.gia = gia;

**this**.dVT = dVT;

**this**.maLoai = maLoai;

}

**public** ThucAn() {

**super**();

}

**public** **int** getMaTA() {

**return** maTA;

}

**public** **void** setMaTA(**int** maTA) {

**this**.maTA = maTA;

}

**public** String getTenTA() {

**return** tenTA;

}

**public** **void** setTenTA(String tenTA) {

**this**.tenTA = tenTA;

}

**public** **int** getGia() {

**return** gia;

}

**public** **void** setGia(**int** gia) {

**this**.gia = gia;

}

**public** String getdVT() {

**return** dVT;

}

**public** **void** setdVT(String dVT) {

**this**.dVT = dVT;

}

**public** String getMaLoai() {

**return** maLoai;

}

**public** **void** setMaLoai(String maLoai) {

**this**.maLoai = maLoai;

}

// xử lý dữ liệu

**private** Connection con = MyConnect.*con*;

**public** **void** Insert(ThucAn ta) {

**try** {

String sql = String.*format*("insert into thucdon(tenta,gia,dvt,maloai)values ('%s',%d ,'%s','%s' )",

ta.getTenTA(), ta.getGia(), ta.getdVT(), ta.getMaLoai());

PreparedStatement s = con.prepareStatement(sql);

System.***err***.println(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showMessageDialog*(**null**, "Thêm thành công");

} **else** {

JOptionPane.*showConfirmDialog*(**null**, "Thêm thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** Update(ThucAn ta) {

**try** {

String sql = String.*format*("update thucdon set tenta='%s', gia=%d, dvt='%s', maloai='%s' where mata=%d",

ta.getTenTA(), ta.getGia(), ta.getdVT(), ta.getMaLoai(), ta.getMaTA());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showMessageDialog*(**null**, "Cập nhật thành công");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Cập nhật thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** **void** Delete(ThucAn ta) {

**try** {

String sql = String.*format*("delete form thucdon where mata=%d )", ta.getMaTA());

PreparedStatement s = con.prepareStatement(sql);

**if** (s.executeUpdate() > 0) {

JOptionPane.*showMessageDialog*(**null**, "Xoá thành công");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Xoá thất bại");

}

} **catch** (SQLException ex) {

ex.printStackTrace();

}

}

**public** ArrayList<ThucAn> ShowTable(**int** key) {

ArrayList<ThucAn> arrThucAn = **null**;

String sql;

**if** (key == 0) {

sql = "select MaTA,TenTA,Gia,DVT, l.TenLoai from thucdon td, loaita l where td.MaLoai=l.MaLoai";

} **else** {

sql = "select MaTA,TenTA,Gia,DVT, l.TenLoai from thucdon td, loaita l where td.MaLoai=l.MaLoai and l.maloai=" + key;

}

**try** {

Statement st = con.createStatement();

ResultSet rs = st.executeQuery(sql);

arrThucAn = **new** ArrayList<ThucAn>();

**while** (rs.next()) {

ThucAn ta = **new** ThucAn(rs.getInt(1), rs.getString(2), rs.getInt(3), rs.getString(4), rs.getString(5));

arrThucAn.add(ta);

}

} **catch** (SQLException ex) {

JOptionPane.*showMessageDialog*(**null**, "Không lấy được danh sách thực đơn !");

}

**return** arrThucAn;

}

}

### 4.4.2 Events

**public** **void** FillTable() {

ArrayList<ThucAn> arrTA = ta.ShowTable(0);

DefaultTableModel tbmodel = **new** DefaultTableModel();

tbmodel.addColumn("Mã thức uống");

tbmodel.addColumn("Tên thức uống");

tbmodel.addColumn("Giá tiền");

tbmodel.addColumn("Đơn vị tính");

tbmodel.addColumn("Loại");

**if** (arrTA != **null**) {

**int** sotk = 0;

**for** (ThucAn ta : arrTA) {

sotk++;

tbmodel.addRow(**new** Object[] { ta.getMaTA(), ta.getTenTA(), ta.getGia(), ta.getdVT(), ta.getMaLoai() });

}

lblTotal.setText(String.*valueOf*(sotk) + " món");

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Không có thức uống nào");

}

tabThucDon.setModel(tbmodel);

**for** (**int** i = 0; i < tabThucDon.getColumnCount(); i++) {

Class<?> col = tabThucDon.getColumnClass(i);

tabThucDon.setDefaultEditor(col, **null**);

}

}

**private** **void** addevents() {

btnThem.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

*name* = btnThem.getText();

JD\_QL\_ThucDon\_Edit ban = **new** JD\_QL\_ThucDon\_Edit(Run.*main*, **true**, "Thêm thức uống mới");

ban.setVisible(**true**);

FillTable();

}

});

btnSua.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

*name* = btnSua.getText();

**if** (*maTD* > 0) {

JD\_QL\_ThucDon\_Edit ban = **new** JD\_QL\_ThucDon\_Edit(Run.*main*, **true**, "Sửa thức uống");

ban.setVisible(**true**);

FillTable();

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Bạn chưa chọn!");

}

}

});

btnXoa.addActionListener(**new** ActionListener() {

**public** **void** actionPerformed(ActionEvent e) {

**if** (*maTD* > 0) {

ta.setMaTA(*maTD*);

ta.Delete(ta);

FillTable();

} **else** {

JOptionPane.*showMessageDialog*(**null**, "Bạn chưa chọn!");

}

}

});

cbbNhom.addItemListener(**new** ItemListener() {

**public** **void** itemStateChanged(ItemEvent e) {

LoaiThucAn selected = (LoaiThucAn) cbbNhom.getSelectedItem();

}

});

tabThucDon.setSelectionMode(ListSelectionModel.***SINGLE\_SELECTION***);

ListSelectionModel model = tabThucDon.getSelectionModel();

model.addListSelectionListener(**new** ListSelectionListener() {

@Override

**public** **void** valueChanged(ListSelectionEvent e) {

**if** (e.getValueIsAdjusting()) {

**return**;

}

ListSelectionModel lsm = (ListSelectionModel) e.getSource();

**if** (lsm.isSelectionEmpty()) {

} **else** {

**int** selectedRow = lsm.getMinSelectionIndex();

*maTD* = Integer.*parseInt*(tabThucDon.getModel().getValueAt(selectedRow, 0).toString());

}

}

});

}

# TÀI LIỆU THAM KHẢO

[1]. Giáo trình Phân tích và thiết kế hệ thống thông tin. Khoa Công nghệ thông tin, Trường Đại học Nam Cần Thơ.

[2]. Phước Quang(2020), “*MySQL là gì?* ”, <https://viblo.asia/p/sql-la-gi-nhung-dieu-can-biet-ve-sql-GrLZDyjJlk0>

[3]. Got It Vietnam (2021), “*Eclipse là gì?*”, <https://vn.got-it.ai/blog/eclipse-la-gi-huong-dan-cai-dat-eclipse-chi-tiet-nhat#:~:text=Eclipse%20l%C3%A0%20m%E1%BB%99t%20c%C3%B4ng%20c%E1%BB%A5,c%C3%A1ch%20ch%C3%A8n%20plugins%20cho%20project.>

[4]. Vi.wikipedia(2022), “*Java là gì?*” <https://quantrimang.com/java-la-gi-tai-sao-nen-chon-java-143130>

# PHÂN CÔNG CÔNG VIỆC

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ Tên | Lớp | Nội dung công việc tham gia thực hiện |
| 180886 | Nguyễn Đình Nguyên | DH18TIN02 | Code |
| 188525 | Quách Hoàng Luân | DH18TIN02 | Word |