**Mục lục**

Contents

[I. **Giới thiệu** 2](#_Toc74689437)

[**1.** **Lý do chọn đề tài** 2](#_Toc74689438)

[**2.** **Một số yêu cầu chức năng của đề tài** 2](#_Toc74689439)

[**I.** **Thiết kế và chương trình** 2](#_Toc74689440)

[**1.** **Cơ sở dữ liệu** 2](#_Toc74689441)

[**a.** **Bảng Account:** 3](#_Toc74689442)

[**b.** **Bảng Table Food** 3](#_Toc74689443)

[**c.** **Bảng** 4](#_Toc74689444)

[**d.** **Bảng Food** 4](#_Toc74689445)

[**e.** **Bảng Bill** 4](#_Toc74689446)

[**f.** **Bảng BillInfo** 4](#_Toc74689447)

[**2.** **Thiết kế giao diện** 5](#_Toc74689448)

[**a.** **Giao diện Login** 5](#_Toc74689449)

[**b.** **Giao diện Quản lý quán café Healer** 8](#_Toc74689450)

[**c.** **Giao diện fAdmin** 11](#_Toc74689451)

[**d.** **Giao diện Thông tin cá nhân** 23](#_Toc74689452)

[**II.** **Kết luận** 25](#_Toc74689453)

[**III.** **Tham khảo** 25](#_Toc74689454)

**QUẢN LÝ QUÁN CÀ PHÊ**

1. **Giới thiệu**
2. **Lý do chọn đề tài**

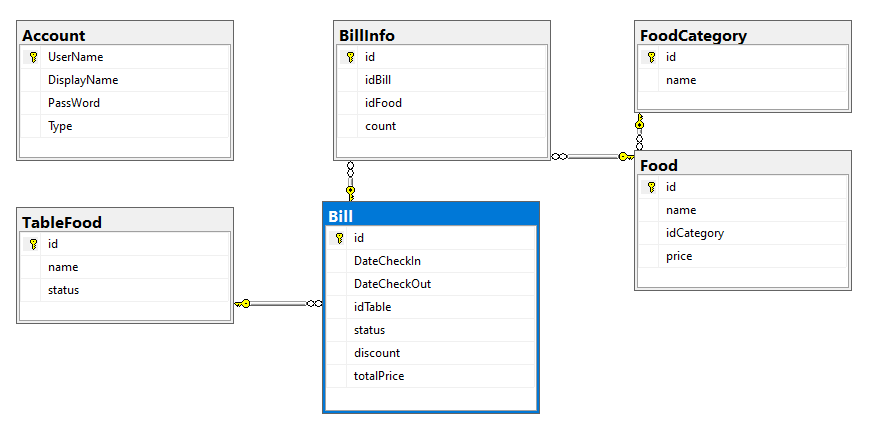
Ngày nay, với nhu cầu tăng cao của các địa điểm cà phê, trà sữa,… phục vụ nhu cầu của khách hàng không chỉ để thư giãn, tán ngẫu sau những giờ học tập làm việc căng thẳng mà còn trở thành nơi học tập làm việc một cách hiệu quả đối với một số thực khách. Việc mở một quán cà phê đòi hỏi nhiều yếu tố: kiến thức kinh doanh, tài chính, cơ sở vật chất, địa lý thuận lợi cho việc buôn bán… Bên cạnh đó thì việc quản lý quán cà phê như thế nào một cách hiệu quả cũng là một câu hỏi lớn cần được giải quyết. Với sự cạnh tranh như ngày nay thì chất lượng phục vụ là một trong những điều kiện tiên quyết để một quán cà phê duy trì hoạt động và phát triển. Thay vì mọi thứ đều được chúng ta ghi nhớ hoặc ghi chép lại một cách thủ công tốn nhiều công sức, lãng phí mà không đem lại hiệu quả cũng như độ chính xác tối ưu. Vậy thì câu hỏi đặt ra là quản lý quán cà phê như thế nào và bằng cách gì? Quản lý cái gì? Quản lý như thế nào là hiệu quả tiện lợi? Công sức bỏ ra ít nhưng lại cho kết quả một cách chính xác?

Và đó là lý do để em chọn đề tài là quản lý quán cà phê để trả lời những câu hỏi trên. Mặc dù chỉ là một ứng dụng giả lập đơn giản về cách quản lý quán cà phê tuy nhiên cũng mang lại một trải nghiệm hữu ích khi được học hỏi tìm hiểu thêm rất nhiều kiến thức khác nhau từ Windows Form, C#, ADO.NET đến việc ứng dụng cả kiến thức đã được học của môn SQL Sever. Vì chỉ là ứng dụng giả lập nên vẫn chưa thỏa mãn các yêu cầu để được mua bán hoặc áp dụng ra bên ngoài.

1. **Một số yêu cầu chức năng của đề tài**

* Chức năng chính của ứng dụng quản lý quán cà phê giúp quản lý thông tin cơ bản như:
* Quản lý doanh thu
* Quản lý thức uống
* Quản lý danh mục loại thức uống
* Quản lý bàn
* Quản lý tài khoản
* Ngoài ra ứng dụng còn có những chức năng khác cho phép người dùng dễ dàng quản lý hơn:
* Cho biết được thông tin từng bàn (Còn trống hay đã có người)
* Thêm bớt thức uống
* Chuyển bàn
* Gộp bàn
* Thực hiện việc thanh toán
* Cho biết thông tin quá trình buôn bán trong một khoảng thời gian nào đó
* Thêm sửa xóa xem tìm kiếm

1. **Thiết kế và chương trình**
2. **Cơ sở dữ liệu**



**Hình 1: Cơ sở dữ liệu Databases**

Trong đó:

1. **Bảng Account:**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| UserName | Tên đăng nhập | Khóa |
| DisplayName | Tên hiển thị |  |
| Password | Mật khẩu |  |
| Type | Loại tài khoản |  |

1. **Bảng Table Food**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã bàn | Khóa |
| name | Tên bàn |  |
| status | Trạng thái  (Trống hoặc Có người) |  |

1. **Bảng FoodCategory**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã loại thức ăn, thức uống | Khóa |
| name | Tên loại thức ăn, thức uống |  |

1. **Bảng Food**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã thức ăn, thức uống | Khóa |
| name | Tên thức ăn, thức uống |  |
| idCategory | Mã loại thức ăn, thức uống |  |
| price | Giá |  |

1. **Bảng Bill**

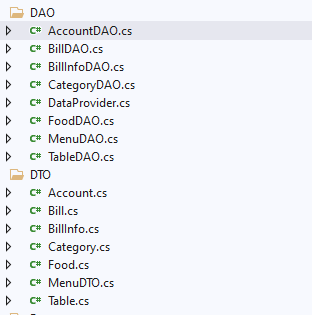
|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã hóa đơn | Khóa |
| DateCheckIn | Thời gian vào quán |  |
| DateCheckOut | Thời gian ra khỏi quán |  |
| idTable | Mã bàn |  |
| status | Trạng thái |  |
| discount | Giảm giá |  |
| totalPrice | Tổng tiền |  |

1. **Bảng BillInfo**

|  |  |  |
| --- | --- | --- |
| Tên trường thông tin | Cụ thể | Ghi chú |
| id | Mã thông tin hóa đơn | Khóa |
| idBill | Mã hóa đơn |  |
| idFood | Mã thức ăn, thức uống |  |
| count | Số lượng |  |

1. **Thiết kế giao diện**

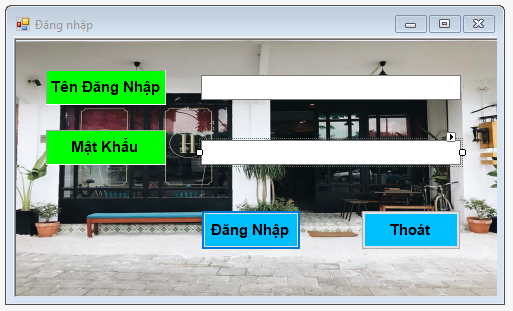
Mô hình 3 lớp dùng chung cho cả ứng dụng bao gồm: GUI, DAO, DTO



Trong đó:

* **DAO (Data Access Layer):** có nhiệm vụ chuyển đổi dữ liệu, xử lý dữ liệu giữa Database và GUI để kết nối với Database lấy dữ liệu lên rồi gửi lên GUI để show. Hoặc nhận yêu cầu từ GUI xuống để xử lý lấy dữ liệu từ Database lên.
* **DTO (Data Tranfer Object):** có thể là những class như: Account, Bill, BillInfo, Category, Food, MenuDTO, Table… dùng để lấy truyển dữ liệu lên nhiệm vụ lưu trữ dữ liệu từ phía client chuyển đổi dữ liệu từ dạng DataSet hoặc DataTable để dễ dàng sử dụng hơn.
* **GUI** là các giao diện tương tác với người dùng (xem là các form):

1. **Giao diện Login**



**Hình 2.1 Màn hình Đăng nhập**

* **Các thành phần trong màn hình:**

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| Tên Đăng Nhập | Textbox | Người dùng nhập Tên Đăng Nhập |
| Mật Khẩu | Textbox | Người dùng nhập Mật Khẩu |
| Đăng Nhập | Button | Đăng Nhập khi đã điền đủ thông tin |
| Thoát | Button | Thoát khỏi màn hình đăng nhập |

* **Các chức năng liên quan đến màn hình Đăng nhập**
* Button “Đăng Nhập” dùng để đăng nhập vào hệ thống khi người dùng click vào, khi người dùng đã nhập Tên Đăng Nhập và Mật Khẩu vào ô textBox thì sẽ kiểm tra xem lại tài khoản và mật khẩu có tồn tại trong cơ sở dữ hay không nếu tài khoản tồn tại thì cho người dùng đăng nhập vào, ngược lại thì sẽ có một Messages Box thông báo Tên Đăng Nhập và Mật Khẩu không chính xác.
* Button “Thoát” cho phép người dùng thoát khỏi giao diện màn hình Đăng nhập.
* **Câu lệnh cho các chức năng trên màn hình Đăng nhập**
* **Câu lệnh cho Button Đăng Nhập**
* **fLogin**

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

{

try

{

string userName = tbUserName.Text;

string passWord = tbPassWord.Text;

if (Login(userName, passWord))

{

Account loginAccount = AccountDAO.Instance.GetAccountByUserName(userName);

fTableManager f = new fTableManager(loginAccount);

this.Hide();

f.ShowDialog();

this.Show();

}

else

{

MessageBox.Show("Tài khoản hoặc mật khẩu không đúng!", "Warming", MessageBoxButtons.OK, MessageBoxIcon.Warning);

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

bool Login(string userName, string passWord)

{

return AccountDAO.Instance.Login(userName, passWord);

}

* **Class AccountDAO**

public bool Login(string userName, string passWord)

{

byte[] temp = ASCIIEncoding.ASCII.GetBytes(passWord);

byte[] hasData = new MD5CryptoServiceProvider().ComputeHash(temp);

string hasPass = "";

foreach (byte item in hasData)

{

hasPass += item;

}

string query = "USP\_Login @userName , @passWord";

DataTable result = DataProvider.Instance.ExecuteQuery(query, new object[] { userName, hasPass});

return result.Rows.Count > 0;

}

* **SQL**

CREATE PROC USP\_Login

@userName nvarchar(100) , @passWord nvarchar(100)

AS

BEGIN

SELECT \* FROM dbo.Account WHERE UserName = @userName AND PassWord = @passWord

END

GO

* **Câu lệnh cho Button Thoát**
* **fLogin**

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

{

Application.Exit();

}

1. **Giao diện Quản lý quán café Healer**



**Hình 3 Màn hình Quản lý quán café Healer**

* **Các thành phần trong màn hình**

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
|  | Combobox | Chọn loại đồ dùng |
|  | Combobox | Chọn đồ dùng |
|  | NumericUpDown | Số lượng |
| Thêm món | Button | Thêm món vào hóa đơn của bàn được chọn |
|  | DataGridView | Hiển thị tên món số lượng đơn giá thành tiền |
|  | Combobox | Chọn số bàn cần chuyển |
| Chuyển bàn | Button | Thực hiển chuyển thông tin từ bàn này sang bàn khác |
| Gộp bàn | Button | Thực hiện gộp 2 bàn thành 1 bàn |
|  | NumericUpDown | Giảm giá |
| Thanh toán | Button | Thực hiện thanh toán hóa đơn |
|  | Textbox | Hiển thị thông tin tổng tiền thanh toán |
| Admin | ToolStripMenuItem | Truy cập vào màn hình Admin |
| Thông tin tài khoản | ToolStripMenuItem | Truy cập vào màn hình Thông tin cá nhân |
| Chức năng | ToolStripMenuItem | Thức hiện thêm món, thanh toán bằng phím tắt |

* **Các chức năng liên quan đến màn hình**
* Button “Thêm món” để thêm món vào hóa đơn của bàn đã chọn
* Button “Chuyển bàn” để thực hiển chuyển thông tin từ bàn này sang bàn khác
* Button “Gộp bàn” để thực hiện gộp 2 bàn thành 1 bàn
* Button “Thanh toán” để thực hiển tính toán cho ra tổng tiền cần thanh toán của bàn
* **Câu lệnh cho các chức năng trên màn hình**
* **Câu lệnh cho chức năng thêm món**

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

{

Table table = lsvTable.Tag as Table;

if (table == null)

{

MessageBox.Show("Hãy chọn bàn");

return;

}

int idBill = BillDAO.Instance.GetUncheckBillIDByTableID(table.ID);

int foodID = (cbbFood.SelectedItem as Food).ID;

int count = (int)nmFoodCount.Value;

if (idBill == -1)

{

BillDAO.Instance.InsertBill(table.ID);

BillInfoDAO.Instance.InsertBillInfo(BillDAO.Instance.GetMaxIDBill(), foodID, count);

}

else

{

BillInfoDAO.Instance.InsertBillInfo(idBill, foodID, count);

}

ShowBill(table.ID);

loadTable();

}

* **Câu lệnh cho chức năng chuyển bàn**

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

{

int id1 = (lsvTable.Tag as Table).ID;

int id2 = (cbbSwitchTable.SelectedItem as Table).ID;

if (MessageBox.Show(string.Format("Bạn có thật sự muốn chuyển bàn {0} qua bàn {1}", (lsvTable.Tag as Table).Name, (cbbSwitchTable.SelectedItem as Table).Name), "Thông báo", MessageBoxButtons.OKCancel) == System.Windows.Forms.DialogResult.OK)

{

TableDAO.Instance.SwitchTable(id1, id2);

loadTable();

}

}

* **Câu lệnh cho chức năng gộp bàn**

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

{

int id1 = (lsvTable.Tag as Table).ID;

int id2 = (cbbGopBan.SelectedItem as Table).ID;

if (MessageBox.Show(string.Format("Bạn có thật sự muốn gộp bàn {0} qua bàn {1}", (lsvTable.Tag as Table).Name, (cbbGopBan.SelectedItem as Table).Name), "Thông báo", MessageBoxButtons.OKCancel) == System.Windows.Forms.DialogResult.OK)

{

TableDAO.Instance.GopBan(id1, id2);

loadTable();

}

}

* **Câu lệnh cho chức năng thanh toán**

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

{

Table table = lsvTable.Tag as Table; //để lấy dữ liệu từ cái lsvTable;

int idBill = BillDAO.Instance.GetUncheckBillIDByTableID(table.ID);

int discount = (int)nmDiscount.Value;

double totalPrice = Convert.ToDouble(tbTotalPrice.Text.Split(',')[0].Replace(".", ""));

double finalTotalPrice = totalPrice - (totalPrice / 100) \* discount;

if (idBill != -1)

{

if (MessageBox.Show(string.Format("Bạn có chắc thanh toán hóa đơn cho bàn {0}\nTổng tiền - (Tổng tiền / 100) x Giảm giá\n=> {1} - ({1} / 100) x {2} = {3}", table.Name, totalPrice, discount, finalTotalPrice), "Thông báo", MessageBoxButtons.OKCancel) == System.Windows.Forms.DialogResult.OK)

{

BillDAO.Instance.CheckOut(idBill, discount, (float)finalTotalPrice);

ShowBill(table.ID);

loadTable();

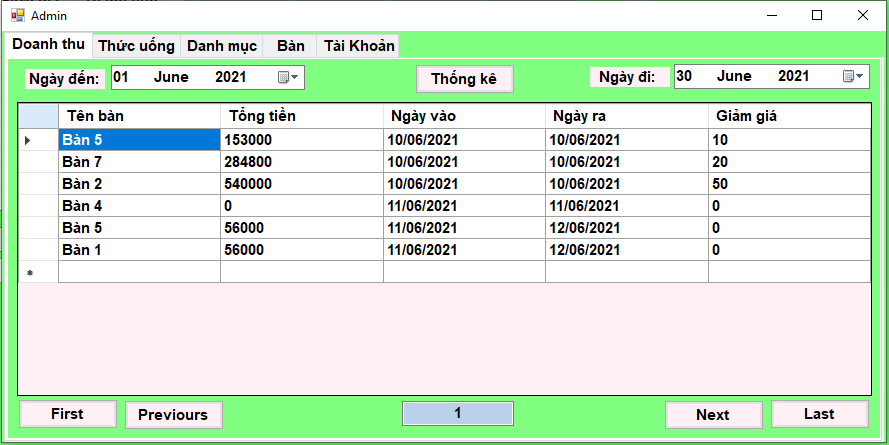
}

}

}

1. **Giao diện fAdmin**

* **Form này gồm có 5 TabControl như sau**:
* **Tab Doanh thu**



**Hình 4 Màn hình Doanh thu**

* **Các thành phần trong tab**

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| Ngày đến | DateTimePicker | Người dùng nhập vào ngày tháng năm |
| Ngày đi | DateTimePicker | Người dùng nhập vào ngày tháng năm |
| Thông kê | Button | Hiển thị danh sách doanh thu theo ngày tháng năm đã điền |
| First | Button | Hiển thị danh sách doanh thu ở trang đầu |
| Previours | Button | Hiển thị trang trước của danh sách doanh thu |
| Next | Button | Hiển thị trang tiếp theo của danh sách doanh thu |
| Last | Button | Hiển thị trang cuối cùng của danh sách doanh thu |
|  | Textbox | Hiển thị số trang hiển tại của danh sách doanh thu |

* **Các chức năng liên quan đến màn hình**
* Button “Thông kê” sau khi click sẽ hiển thị danh sách doanh thu của quán trong một khoản thời gian nào đó.
* Button “First” sau khi click sẽ hiển thị trang đầu tiên của danh sách doanh thu
* Button “Last” sau khi click sẽ hiển thị trang cuối cùng của danh sách doanh thu
* Button “Previours” sau khi click sẽ hiển thị trang trước của danh sách doanh thu
* Button “Next” sau khi click sẽ hiển thị trang tiếp theo của danh sách doanh thu
* **Câu lệnh cho các chức năng trên màn hình**
* **Câu lệnh cho chức năng Thống kê**

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

{

LoadListBillByDate(dtpkFromDate.Value, dtpkToDate.Value);

}

* **Câu lệnh cho chức năng First**

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

{

tbPageBill.Text = "1";

}

* **Câu lệnh cho chức năng Last**

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

{

int sumRecord = BillDAO.Instance.GetNumBillListByDate(dtpkFromDate.Value, dtpkToDate.Value);

int lastPage = sumRecord / 10;

if (sumRecord % 10 != 0)

lastPage++;

tbPageBill.Text = lastPage.ToString();

}

* **Câu lệnh cho chức năng Previours**

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

{

int page = Convert.ToInt32(tbPageBill.Text);

if (page > 1)

page--;

tbPageBill.Text = page.ToString();

}

* **Câu lệnh cho chức năng Next**

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

{

int page = Convert.ToInt32(tbPageBill.Text);

int sumRecord = BillDAO.Instance.GetNumBillListByDate(dtpkFromDate.Value, dtpkToDate.Value);

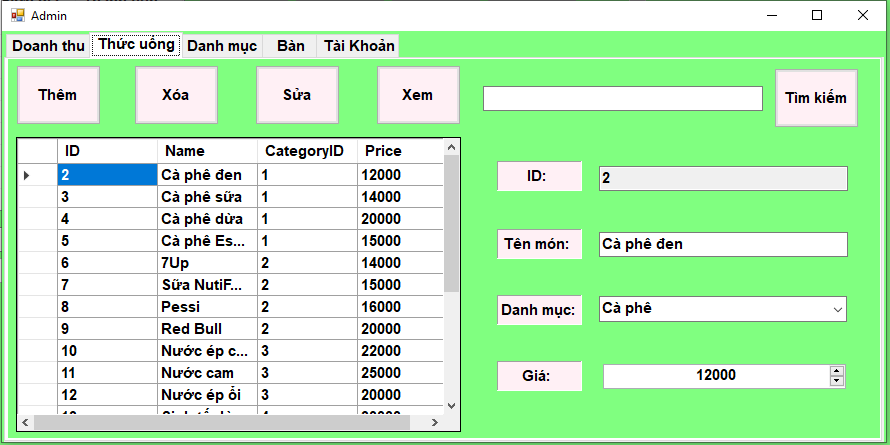
if (page < sumRecord)

page++;

tbPageBill.Text = page.ToString();

}

* **Tab Thức uống**



**Hình 4 Màn hình Thức uống**

* **Các thành phần trong tab**

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| ID | Textbox (ReadOnly) | ID được binding từ DataSource |
| Tên món | Textbox | Người dùng có thể xem và nhập vào tên món |
| Danh mục | Combobox | Người dùng có thể xem và chọn danh mục |
| Gía | NumericUpDown | Người dùng có thể xem và nhập giá tiền |
|  | DataGridView | Hiển thị thông tin thức uống |
| Thêm | Button | Thêm thức uống vào danh sách thức uống |
| Xóa | Button | Xóa thức uống đó ra khỏi danh sách thức uống |
| Sửa | Button | Cập nhật lại thông tin thức uống |
| Xem | Button | Xem danh sách thức uống |
|  | Textbox | Nhập vào thức uống cần tìm |
| Tìm kiếm | Button | Tìm thức uống vừa nhập |

* **Các chức năng liên quan đến màn hình**
* Button “Thêm” thêm thông tin thức uống đã nhập vào danh sách thức uống
* Button “Xóa” xóa thông tin thức uống được chọn ra khỏi danh sách thức uống
* Button “Sửa” cập nhật lại thông tin thức uống
* Button “Xem” xem danh sách thức uống
* Button “Tìm kiếm” tìm kiếm thức uống theo tên thức uống một cách gần đúng
* **Câu lệnh cho các chức năng trên màn hình**
* **Câu lệnh cho chức năng Thêm**

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

{

string name = tbFoodName.Text;

int categoryID = (cbbFoodCategory.SelectedItem as Category).ID;

float price = (float)nmFoodPrice.Value;

if (FoodDAO.Instance.InsertFood(name, categoryID, price))

{

MessageBox.Show("Thêm món thành công");

LoadListFood();

if (insertFood != null)

insertFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi thêm thức ăn");

}

}

* **Câu lệnh cho chức năng Xóa**

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

{

int id = Convert.ToInt32(tbFoodID.Text);

if (FoodDAO.Instance.DeleteFood(id))

{

MessageBox.Show("Xóa món thành công");

LoadListFood();

if (deleteFood != null)

deleteFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi xóa thức ăn");

}

}

* **Câu lệnh cho chức năng Sửa**

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

{

string name = tbFoodName.Text;

int categoryID = (cbbFoodCategory.SelectedItem as Category).ID;

float price = (float)nmFoodPrice.Value;

int id = Convert.ToInt32(tbFoodID.Text);

if (FoodDAO.Instance.UpdateFood(id, name, categoryID, price))

{

MessageBox.Show("Sửa món thành công");

LoadListFood();

if (updateFood != null)

updateFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi sửa thức ăn");

}

}

* **Câu lệnh cho chức năng Xem**

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

{

LoadListFood();

}

* **Câu lệnh cho chức năng Tìm kiếm**

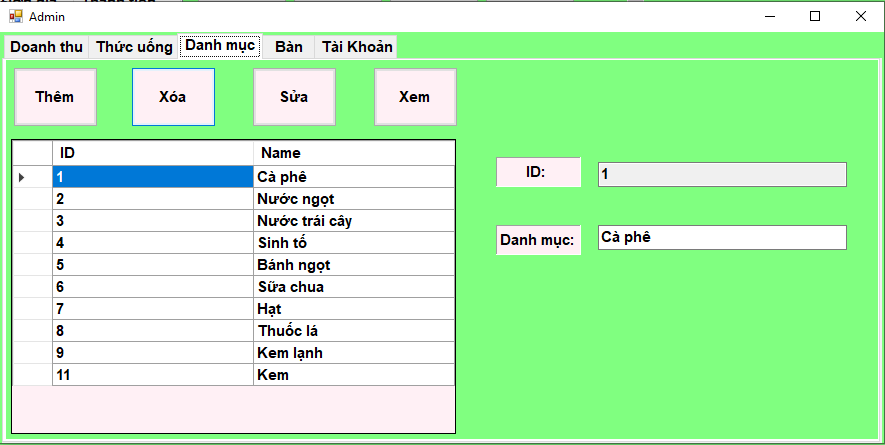
private void btnSearchFood\_Click(object sender, EventArgs e)

{

foodList.DataSource = SearchFoodByName(tbSearchFoodName.Text);

}

* **Tab Danh mục**



**Hình 5 Màn hình Danh mục**

* **Các thành phần trong tab**

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| ID | Textbox (ReadOnly) | ID được binding từ DataSource |
| Danh mục | Combobox | Người dùng có thể xem và nhập danh mục |
|  | DataGridView | Hiển thị thông tin loại thức uống |
| Thêm | Button | Thêm loại thức uống vào danh sách loại thức uống |
| Xóa | Button | Xóa loại thức uống đó ra khỏi danh sách loại thức uống |
| Sửa | Button | Cập nhật lại thông tin loại thức uống |
| Xem | Button | Xem danh sách loại thức uống |

* **Các chức năng liên quan đến màn hình**
* Button “Thêm” thêm thông tin loại thức uống đã nhập vào danh sách loại thức uống
* Button “Xóa” xóa thông tin loại thức uống được chọn ra khỏi danh sách loại thức uống
* Button “Sửa” cập nhật lại thông tin loại thức uống
* Button “Xem” xem danh sách loại thức uống
* **Câu lệnh cho các chức năng trên màn hình**
* **Câu lệnh cho chức năng Thêm**

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

{

string name = tbCategory.Text;

if (CategoryDAO.Instance.InsertFoodCategory(name))

{

MessageBox.Show("Thêm loại thức ăn thành công");

LoadListFoodCategory();

if (insertFoodCategory != null)

insertFoodCategory(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi thêm loại thức ăn");

}

}

* **Câu lệnh cho chức năng Xóa**

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

{

int id = Convert.ToInt32(tbCategoryID.Text);

if (CategoryDAO.Instance.DeleteFoodCategory(id))

{

MessageBox.Show("Xóa loại thức ăn thành công");

LoadListFoodCategory();

if (deleteFoodCategory != null)

deleteFoodCategory(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi xóa loại thức ăn");

}

}

* **Câu lệnh cho chức năng Sửa**

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

{

string name = tbCategory.Text;

int id = Convert.ToInt32(tbCategoryID.Text);

if (CategoryDAO.Instance.UpdateFoodCategory(id, name))

{

MessageBox.Show("Sửa món thành công");

LoadListFoodCategory();

if (updateFoodCategory != null)

updateFoodCategory(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi sửa thức ăn");

}

}

* **Câu lệnh cho chức năng Xem**

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

{

LoadListFoodCategory();

}

* **Tab Bàn**



**Hình 6 Màn hình Bàn**

* **Các thành phần trong tab**

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| ID | Textbox (ReadOnly) | ID được binding từ DataSource |
| Tên bàn | Textbox | Người dùng có thể xem và nhập vào tên bàn |
| Trạng thái | Combobox | Người dùng có thể xem và chọn trạng thái |
|  | DataGridView | Hiển thị thông tin bàn |
| Thêm | Button | Thêm một bàn vào danh sách bàn |
| Xóa | Button | Xóa một bàn ra khỏi danh sách bàn |
| Sửa | Button | Cập nhật lại thông tin một bàn |
| Xem | Button | Xem danh sách bàn |

* **Các chức năng liên quan đến màn hình**
* Button “Thêm” thêm bàn đã nhập vào danh sách bàn
* Button “Xóa” xóa bàn được chọn ra khỏi danh sách bàn
* Button “Sửa” cập nhật lại bàn
* Button “Xem” xem danh sách bàn
* **Câu lệnh cho các chức năng trên màn hình**
  + **Câu lệnh cho chức năng Thêm**

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

{

string name = tbTableName.Text;

string status = cbTableStatus.Text;

if (TableDAO.Instance.InsertTableFood(name, status))

{

MessageBox.Show("Thêm bàn ăn thành công");

LoadListTableFood();

if (insertTableFood != null)

insertTableFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi thêm bàn ăn");

}

}

* + **Câu lệnh cho chức năng Xóa**

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

{

int id = Convert.ToInt32(tbTableID.Text);

if (TableDAO.Instance.DeleteTableFood(id))

{

MessageBox.Show("Xóa bàn ăn thành công");

LoadListTableFood();

if (deleteTableFood != null)

deleteTableFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi xóa thức ăn");

}

}

* + **Câu lệnh cho chức năng Sửa**

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

{

int id = Convert.ToInt32(tbTableID.Text);

string name = tbTableName.Text;

string status = cbTableStatus.Text;

if (TableDAO.Instance.UpdateTableFood(id, name, status))

{

MessageBox.Show("Sửa bàn ăn thành công");

LoadListTableFood();

if (updateTableFood != null)

updateTableFood(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi sửa bàn ăn");

}

}

* + **Câu lệnh cho chức năng Xem**

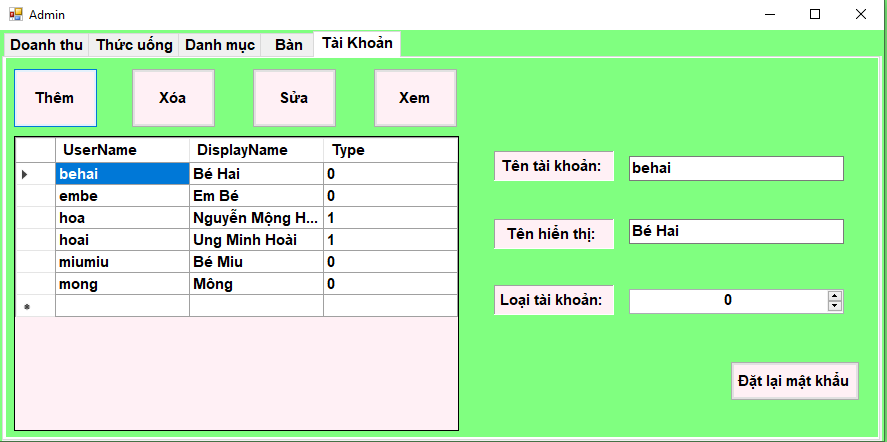
private void btnShowTable\_Click(object sender, EventArgs e)

{

LoadListTableFood();

}

* **Tab Tài khoản**



**Hình 7 Màn hình Tài khoản**

* **Các thành phần trong tab**

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| Tên tài khoản | Textbox | Người dùng có thể xem và nhập tên tài khoản |
| Tên hiển thị | Textbox | Người dùng có thể xem và nhập vào tên hiển thị |
| Loại tài khoản | NumericUpDown | Người dùng có thể xem và chọn loại tài khoản |
|  | DataGridView | Hiển thị thông tin tài khoản |
| Thêm | Button | Thêm một tài khoản vào danh sách tài khoản |
| Xóa | Button | Xóa một tài khoản ra khỏi danh sách tài khoản |
| Sửa | Button | Cập nhật lại thông tin một tài khoản |
| Xem | Button | Xem danh sách tài khoản |
| Đặt lại mật khẩu | Button | Cập nhật lại thành mật khẩu mặc định |

* **Các chức năng liên quan đến màn hình**
* Button “Thêm” thêm thông tin tài khoản đã nhập vào danh sách tài khoản
* Button “Xóa” xóa thông tin tài khoản được chọn ra khỏi danh sách tài khoản
* Button “Sửa” cập nhật lại thông tin tài khoản
* Button “Xem” xem danh sách tài khoản
* Button “Đặt lại mật khẩu” đặt mật khẩu về giá trị mặc định
* **Câu lệnh cho các chức năng trên màn hình**
* **Câu lệnh cho chức năng Thêm**

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

{

string userName = tbUserName.Text;

string displayName = tbDisplayName.Text;

int type = (int)nmAccountType.Value;

AddAccount(userName, displayName, type);

}

* + **Câu lệnh cho chức năng Xóa**

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

{

string userName = tbUserName.Text;

DeleteAccount(userName);

}

* + **Câu lệnh cho chức năng Sửa**

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

{

string userName = tbUserName.Text;

string displayName = tbDisplayName.Text;

int type = (int)nmAccountType.Value;

EditAccount(userName, displayName, type);

}

* + **Câu lệnh cho chức năng Xem**

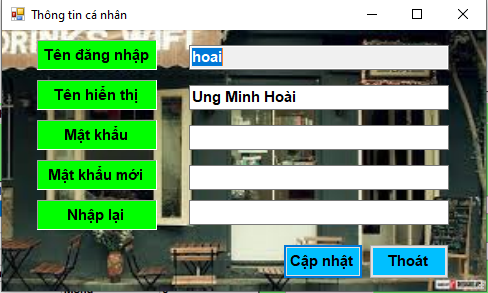
private void btnShowAccount\_Click(object sender, EventArgs e)

{

LoadAccount();

}

1. **Giao diện Thông tin cá nhân**



**Hình 8 Màn hình Thông tin tài khoản**

* **Các thành phần trong màn hình**

|  |  |  |
| --- | --- | --- |
| Tên | Kiểu | Mô tả |
| Tên đăng nhập | Textbox | Người dùng nhập vào tên đăng nhập |
| Tên hiển thị | Textbox | Người dùng nhập vào tên hiển thị |
| Mật khẩu | Textbox | Người dùng nhập vào mật khẩu |
| Mật khẩu mới | Textbox | Người dùng nhập vào mật khẩu mới |
| Nhập lại | Textbox | Người dùng nhập lại vào mật khẩu mới để xác minh |
| Cập nhật | Button | Cập nhật lại thông tin một tài khoản |
| Thoát | Button | Xem danh sách tài khoản |

* **Các chức năng liên quan đến màn hình**
* Button “Cập nhật” cập nhật lại thông tin cá nhân
* Button “Thoát” thoát khỏi màn hình thông tin cá nhân
* **Câu lệnh cho các chức năng trên màn hình**
  + **Câu lệnh cho chức năng Cập nhật**

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

{

UpdateAccountInfo();

}

* + **Câu lệnh cho chức năng Thoát**

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

{

this.Close();

}

1. **Kết luận**

Qua việc thực hành Project em đã nắm được thêm những kiến thức về cách tổ chức cơ bản của một ứng dụng. Học thêm được một ngôn ngữ lập trình C#, làm theo mô hình 3 lớp giúp thuận tiện hơn trong việc code, giúp rút ngắn lượng code phải dùng lặp lại và ứng dụng cũng được trình bày một cách logic hơn. Ngoài ra, cải thiện và nâng cao khả năng tự học của bản thân.

1. **Tham khảo**
2. Lập trình phần mềm Quản lý quán café C# Winform – Serial HowKteam
3. https://www.youtube.com/playlist?list=PL33lvabfss1xnPhBJHjM0A8TEBBcGCTsf

