

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH



LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

Báo cáo bài tập lớn:

XÂY DỰNG HỆ THỐNG BÁN HÀNG VÀ HỖ TRỢ QUẢN LÝ CHO QUÁN CÀ PHÊ

GVHD: Mai Đức Trung
SV: Phạm Quý Luân 1511899
Dinh Hoàng Kim 1711872
Nguyễn Thị Trúc Ly 1710187
Phan Thành Đạt 1710984

TP. HỒ CHÍ MINH, THÁNG 11/2018



Mục lục

1 Giới thiệu chung về đề tài	3
1.1 Tổng quan và lý do chọn đề tài	3
1.2 Mô tả ứng dụng và các công cụ thực hiện	3
2 Lược đồ lớp của hệ thống	4
2.1 Lược đồ lớp của ứng dụng Point-Of-Sale	4
2.1.1 Class Program	5
2.1.2 Class FormLogin – Form đăng nhập	5
2.1.3 Class FormStaff – Giao diện chính để bán hàng	5
2.1.4 Class FormAccountInfo – Form thông tin nhân viên	7
2.1.5 Class FormHistoryBill – Form lịch sử bán hàng	7
2.2 Lược đồ lớp của ứng dụng hỗ trợ quản lý	8
2.2.1 Lược đồ lớp của class Program	9
2.2.2 Lược đồ lớp của class fLogin - Form Đăng nhập	10
2.2.3 Lược đồ lớp của class fMain - Giao diện chính	10
2.2.4 Lược đồ lớp của class fAddItem - Giao diện thêm Item vào CSDL	11
2.2.5 Lược đồ lớp của class ctrl_item - Khung nhìn quản lý item	12
3 Sơ lược về Cơ sở dữ liệu	14
3.1 Bảng quản lý Account	14
3.2 Bảng Quản lý Item	15
3.3 Bảng Quản lý Bill	15
3.4 Bảng Quản lý BillDetail	16
3.5 Thao tác với Cơ sở dữ liệu	16
4 Các chức năng và giao diện của ứng dụng	18
4.1 Phần Point-of-Sale	18
4.1.1 Đăng nhập (FormLogin)	18
4.1.2 Giao diện chính của ứng dụng (FormStaff)	19
4.1.3 Chọn món - hiển thị lên hóa đơn	21
4.1.4 Chọn thanh toán – lưu hóa đơn	22
4.1.5 Xem lại lịch sử bán hàng - FormHistoryBill	24
4.1.6 Xem thông tin cá nhân – FormAccountInfo	24
4.2 Phần ứng dụng hỗ trợ quản lý	26
4.2.1 Đăng nhập	26
4.2.2 Màn hình thao tác chính	27
4.2.3 Màn hình thêm tài khoản / thêm item	28
4.2.4 Màn hình chi tiết hóa đơn	29
5 Tổng kết	30
6 Phân công công việc trong nhóm	31



Danh sách hình vẽ

1	Lược đồ lớp của toàn bộ ứng dụng Point-Of-Sale	4
2	Lược đồ lớp của class FormLogin	5
3	Lược đồ lớp của class FormStaff	6
4	Lược đồ lớp của class FormAccountInfo	7
5	Lược đồ lớp của class FormHistoryBill	7
6	Lược đồ lớp của toàn bộ phần ứng dụng hỗ trợ quản lý	8
7	Lược đồ lớp của class Program	9
8	Lược đồ lớp của class fLogin	10
9	Lược đồ lớp của class fMain	11
10	Lược đồ lớp của class fAddItem	12
11	Lược đồ lớp của class ctrl_item	13
12	Giao diện đăng nhập, ngõ vào của chương trình	18
13	Giao diện chính	20
14	Nhập món và hiển thị lên hóa đơn	21
15	Thông báo thanh toán thành công	23
16	Giao diện xem lịch sử bán hàng	24
17	Giao diện xem thông tin cá nhân của nhân viên	25
18	Giao diện đăng nhập tài khoản quản lý	26
19	Giao diện thao tác chính	28
20	Hai màn hình thêm dữ liệu vào CSDL	28
21	Giao diện thao tác chính	29



1 Giới thiệu chung về đề tài

Trong thời đại công nghệ thông tin đang nổi lên như hiện nay, tất cả mọi lĩnh vực đều có sự xuất hiện của công nghệ dù ít hay nhiều. Đặc biệt là trong ngành dịch vụ, quản lý và bán hàng đã có sự xuất hiện của các **phần mềm quản lý** giúp giảm sức người, tăng độ chính xác và sự tiện lợi. Lợi ích của việc sử dụng phần mềm và nhiều vô kể và trong tương lai, nó sẽ còn gia tăng và có nhiều cải tiến hơn nữa cùng với sự phát triển chóng mặt như hiện nay.

Do đó, nhóm chúng em đã lựa chọn xây dựng **một ứng dụng hỗ trợ quản lý quán Cà Phê** trong bài tập lớn này.

1.1 Tổng quan và lý do chọn đề tài

Nhóm chúng em gồm 4 thành viên, đã thảo luận và chọn thực hiện đề tài xây dựng một hệ thống hỗ trợ quản lý như vậy. Mặc dù nó ít có phần hấp dẫn về mặt đồ họa và thú vị như xây dựng một ứng dụng chơi game, nhưng xét về tính phổ biến và tính có ích của kiến thức thu nhặt được, nhóm chúng em đã chọn hiện thực đề tài này.

Ý tưởng của đề tài ban đầu là một ứng dụng đơn giản thao tác với một Cơ sở dữ liệu (CSDL) nhằm lưu trữ và xử lý các thông tin mua bán của cửa hàng. Nhưng về sau đã có cải tiến, mượn ý tưởng từ mô hình Point-Of-Sale (POS), nhóm đã có suy nghĩ đặt các POS ở khắp các chi nhánh cửa hàng, và có 1 ứng dụng quản lý chính để giúp người chủ dễ dàng quản lý các chi nhánh của mình thông qua một ứng dụng riêng biệt. Vậy, hệ thống chúng em hiện thực bao gồm 2 ứng dụng riêng biệt.

- Ứng dụng Point-Of-Sale
- Ứng dụng hỗ trợ quản lý

1.2 Mô tả ứng dụng và các công cụ thực hiện

Như đã đề cập ở phần trên, hệ thống gồm 2 phần ứng dụng riêng biệt kết hợp lại với nhau. Bên dưới sẽ mô tả bao quát ứng dụng và các công cụ được sử dụng để hiện thực

Phần ứng dụng Point-of-Sale được xây dựng bằng Window Form – tuy hiệu năng không cao, giao diện hiển thị chưa hoàn thiện, nhưng dễ tiếp cận vì cơ bản bám sát với các bài giảng, thực hành trên lớp.

Phần ứng dụng hỗ trợ quản lý được hiện thực bằng ngôn ngữ C# - được phát triển trên nền tảng của Java và C++, là một ngôn ngữ lập trình hiện đại, mục đích tổng quát, hướng đối tượng. Về giao diện, ứng dụng này được xây dựng trên Bunifu Framework [1], đây là một framework hỗ trợ xây dựng giao diện hiện đại, thân thiện với người dùng một cách dễ dàng và nhanh chóng.



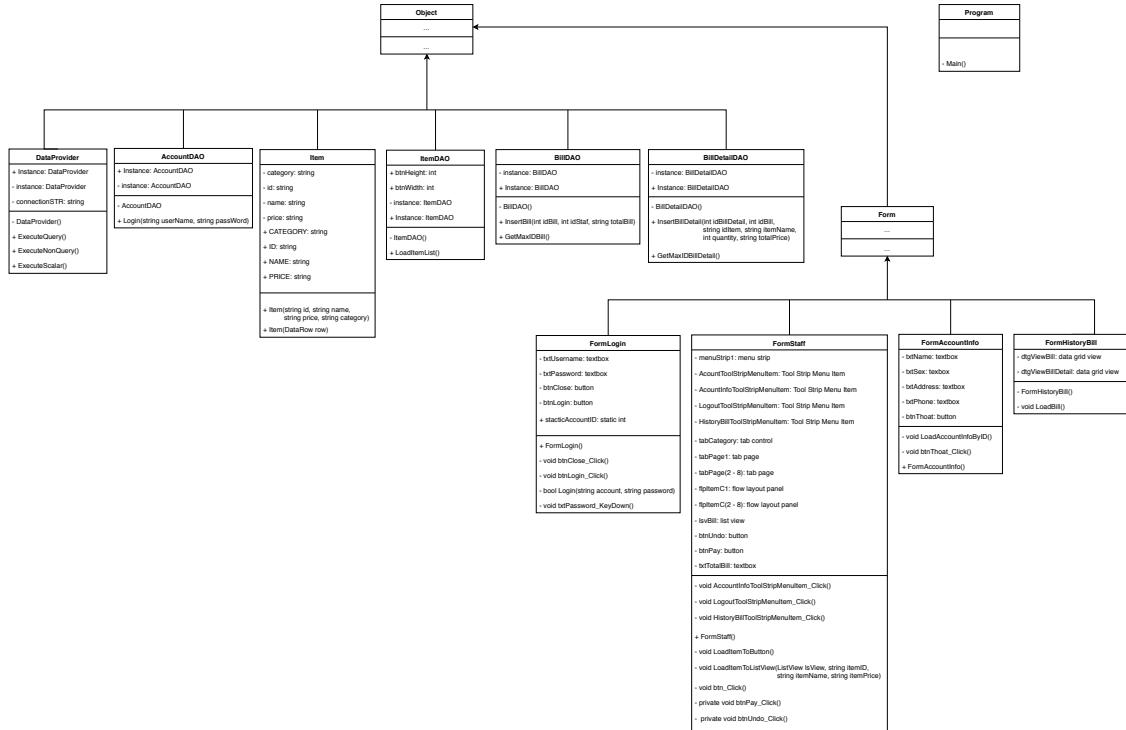
2 Lược đồ lớp của hệ thống

2.1 Lược đồ lớp của ứng dụng Point-Of-Sale

Bao gồm:

- Class Program.
- Các class hỗ trợ kết nối tới Cơ sở dữ liệu, hiện thực chương trình.
- Các form giao diện.

Hình ảnh dưới là lược đồ lớp của toàn bộ ứng dụng POS (Phóng to ra có thể thấy rõ):



Hình 1: Lược đồ lớp của toàn bộ ứng dụng Point-Of-Sale

Dưới đây là phần trình bày một số class:



2.1.1 Class Program

Đây là một static class, chứa hàm Main và do đó nó là ngõ vào của chương trình. Trong class này em chỉ khởi chạy duy nhất màn hình đăng nhập (FormLogin).

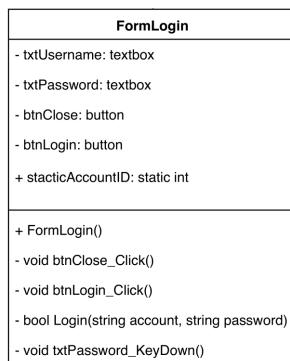
Mã hiện thực:

```
static class Program {  
    /// <summary>  
    /// The main entry point for the application.  
    /// </summary>  
    [STAThread]  
    static void Main() {  
        Application.EnableVisualStyles();  
        Application.SetCompatibleTextRenderingDefault(false);  
        Application.Run(new FormLogin());  
    }  
}
```

2.1.2 Class FormLogin – Form đăng nhập

Giao diện đăng nhập là giao diện đầu tiên hiển thị khi khởi chạy ứng dụng. Class này bao gồm 2 trường để người nhân viên nhập vào username và password (được quản lý cung cấp) Thông qua đó mới có quyền đi tiếp vào ứng dụng.

Dưới đây là class diagram cho class này:



Hình 2: Lược đồ lớp của class FormLogin

Chi tiết hiện thực các hàm chức năng trong class này sẽ được chi tiết ở phần trình bày Chức năng và Giao diện.

2.1.3 Class FormStaff – Giao diện chính để bán hàng

Class FormStaff là nơi hiện thực giao diện chính của ứng dụng, sau khi đăng nhập thành công, user sẽ được chuyển hướng sang giao diện này để thực hiện các thao tác tiếp theo. Mọi chức

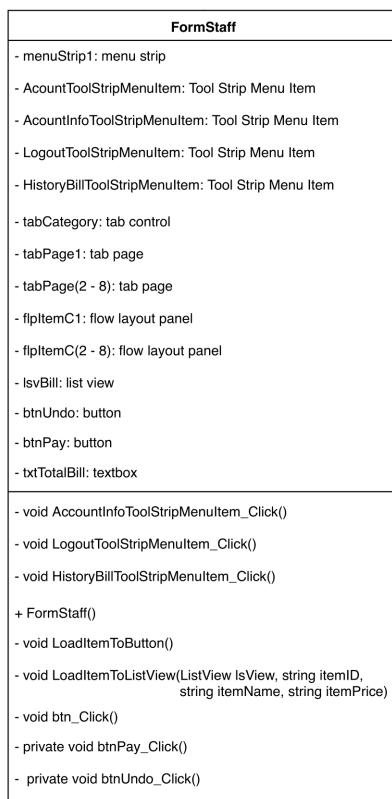


năng nhập xuất hóa đơn và thao tác của ứng dụng đều được rẽ nhánh ra từ màn hình này.

Form này bao gồm:

- Một menu strip để chuyển qua các màn hình khác: đăng xuất, thông tin cá nhân, lịch sử bán hàng.
- Một tab control để lựa chọn danh sách các nước uống.
- Các tab page con của tab control hiển thị danh sách nước uống theo danh mục.
- List view hiển thị hóa đơn.
- Nút hủy hóa đơn (Cancel) và nút thanh toán (Pay).
- Một text box hiển thị tổng tiền hóa đơn.

Dưới đây là class diagram cho class này:



Hình 3: Lược đồ lớp của class FormStaff

Chi tiết hiện thực các hàm chức năng trong class này sẽ được chi tiết ở phần trình bày Chức năng và Giao diện.



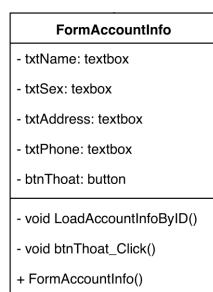
2.1.4 Class FormAccountInfo – Form thông tin nhân viên

Màn hình hiển thị khi nhân viên chọn xem AccountInfo từ màn hình chính.

Form này bao gồm:

- Các text box hiển thị thông tin nhân viên như tên, giới tính, số điện thoại, địa chỉ.
- Một nút thoát để rời khỏi giao diện xem thông tin.

Dưới đây là class diagram cho class này:



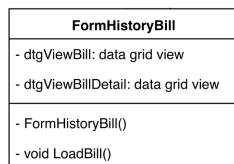
Hình 4: Lược đồ lớp của class FormAccountInfo

Chi tiết hiện thực các hàm chức năng trong class này sẽ được chi tiết ở phần trình bày Chức năng và Giao diện.

2.1.5 Class FormHistoryBill – Form lịch sử bán hàng

Màn hình hiển thị khi nhân viên chọn xem HistoryBill từ màn hình chính. Form này gồm: 2 data grid view để hiển thị hóa đơn và hóa đơn chi tiết. Dưới đây là class diagram cho class này:

Dưới đây là class diagram cho class này:



Hình 5: Lược đồ lớp của class FormHistoryBill

Chi tiết hiện thực các hàm chức năng trong class này sẽ được chi tiết ở phần trình bày Chức năng và Giao diện.



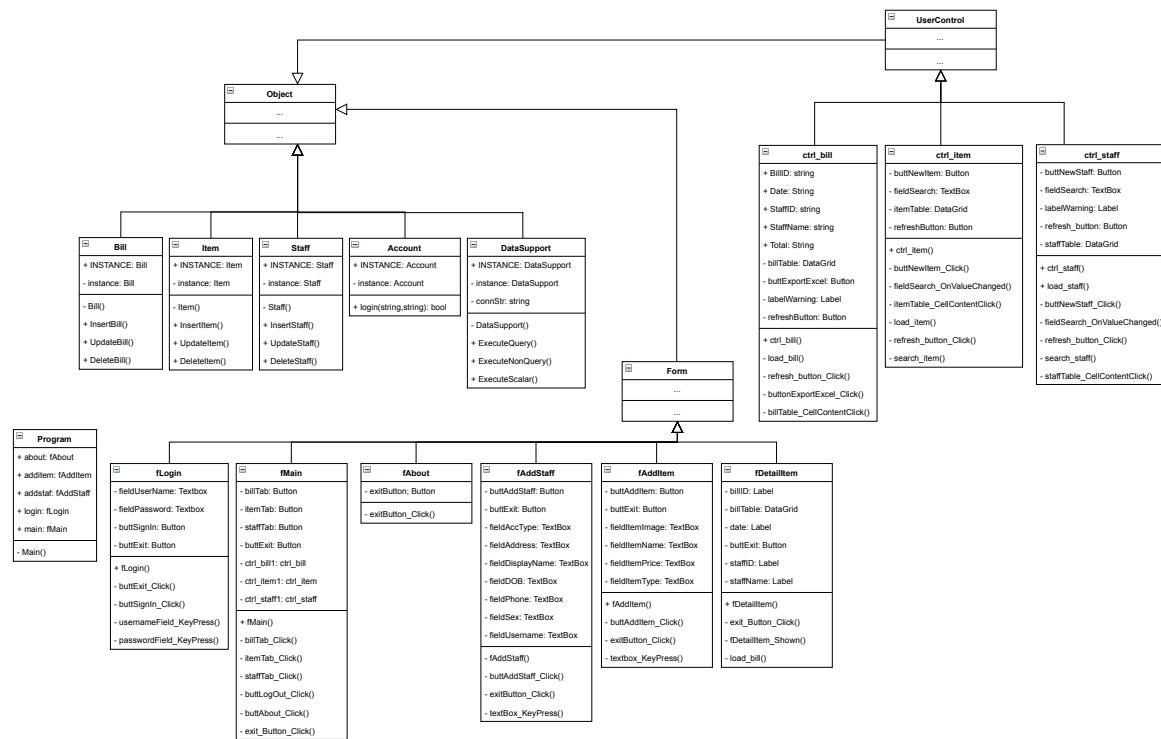
2.2 Lược đồ lớp của ứng dụng hỗ trợ quản lý

Trong lược đồ lớp này, chúng em đã bỏ bớt các thành phần không cần thiết như các label, hình ảnh trang trí thêm cho ứng dụng. Vì kích cỡ của lược đồ lớp là khá lớn và nên khi đưa vào report toàn bộ lược đồ làm kích cỡ chữ bị thu nhỏ lại (nhưng vẫn có thể zoom ra để xem). Trong các mục con, chúng em sẽ đề cập và tách riêng lược đồ lớp cho từng phần của ứng dụng.

Nhìn chung, lược đồ lớp gồm 4 phần chính:

- Class Program, ngõ vào của ứng dụng.
- Các giao diện (Form).
- Các trang nhàn (được hiện thực bằng User Control).
- Các class hỗ trợ tương tác trực tiếp với Cơ sở dữ liệu.

Hình ảnh bên dưới (Hình 6) là hình ảnh lược đồ lớp của toàn bộ ứng dụng hỗ trợ quản lý.



Hình 6: Lược đồ lớp của toàn bộ phần mềm ứng dụng hỗ trợ quản lý

Vì số lượng các class nhiều và có sự giống nhau giữa các class, nên ở phần sau, nhóm sẽ chỉ trình bày một số class đặc trưng chính: class Program, class fLogin, fMain, fAddItem và class ctrl_item.



2.2.1 Lược đồ lớp của class Program

Đây là một static class, chứa hàm Main, và do đó, nó là ngõ vào của chương trình. Class này được dùng để khởi tạo các thành phần cần thiết khi khởi động ứng dụng, và khởi chạy màn hình đầu tiên - màn hình Đăng nhập (fLogin).

Dưới đây là class diagram cho class này. Hình 7



Hình 7: Lược đồ lớp của class Program

Dưới đây là mã hiện thực cho class này.

```
public static class Program
{
    // create form here
    public static fMain      main;
    public static fLogin     login;
    public static fAddItem   additem;
    public static fAddStaff  addstaff;
    public static fAbout     about;

    /// <summary>
    /// The main entry point for the application.
    /// </summary>
    [STAThread]
    static void Main()
    {
        Application.EnableVisualStyles();
        Application.SetCompatibleTextRenderingDefault(false);

        main      = new fMain();
        login     = new fLogin();
        additem   = new fAddItem();
        addstaff = new fAddStaff();
    }
}
```

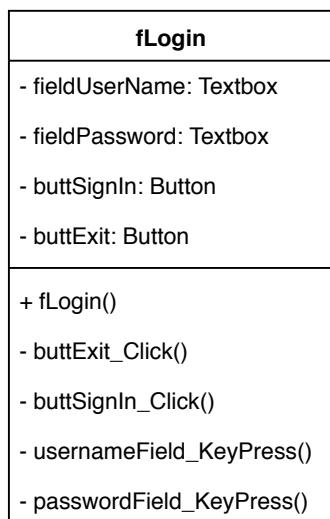


```
    about      = new fAbout();  
  
    Application.Run(login);  
}  
}
```

2.2.2 Lược đồ lớp của class fLogin - Form Đăng nhập

Giao diện đăng nhập là giao diện đầu tiên được hiển thị khi khởi chạy ứng dụng. Class này bao gồm 2 trường để người quản lý nhập vào username và password đã đăng ký trước của họ, và một nút để thực thi thao tác đăng nhập. Thông qua đó mà có quyền để đi sâu hơn vào ứng dụng.

Dưới đây là class diagram cho class này. Hình 8



Hình 8: Lược đồ lớp của class fLogin

Chi tiết hiện thực các hàm chức năng trong class này sẽ được chi tiết ở phần trình bày Chức năng và Giao diện.

2.2.3 Lược đồ lớp của class fMain - Giao diện chính

Class **fMain** là nơi hiện thực giao diện chính của ứng dụng, sau khi đăng nhập thành công, user sẽ được chuyển hướng sang giao diện này để thực hiện các thao tác tiếp theo. Mọi chức năng quản lý, và thao tác của ứng dụng đều được rẽ nhánh ra từ màn hình này.

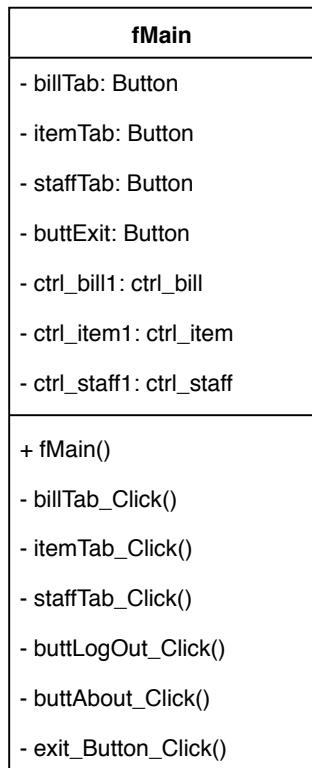
Class này bao gồm:

- Các nút để chuyển hướng sang các tính năng tương ứng.
- Các trang nhàn để hiển thị các mục quản lý tương ứng.



Các trang nhàn được hiện thực bằng (kê thừa) từ **User Control** trong C#.

Dưới đây là class diagram cho class này. Hình 9



Hình 9: Lược đồ lớp của class fMain

Chi tiết hiện thực các hàm chức năng trong class này sẽ được chi tiết ở phần trình bày Chức năng và Giao diện.

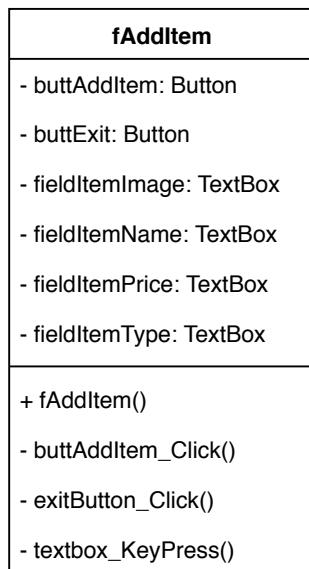
2.2.4 Lược đồ lớp của class fAddItem - Giao diện thêm Item vào CSDL

Class **fAddItem** là nơi hiện thực giao diện thêm item vào CSDL của ứng dụng.

Class này bao gồm:

- Các trường để nhập thông tin của item mới.
- Nút để thực hiện thao tác.

Dưới đây là class diagram cho class này. Hình 10



Hình 10: Lược đồ lớp của class fAddItem

Chi tiết hiện thực các hàm chức năng trong class này sẽ được chi tiết ở phần trình bày Chức năng và Giao diện.

2.2.5 Lược đồ lớp của class ctrl_item - Khung nhìn quản lý item

Class **ctrl_item** là nơi hiện thực khung nhìn quản lý item, là 1 phần cấu thành giao diện chính.

Class này bao gồm:

- Bảng hiển thị danh sách Item.
- Các nút thêm, xóa, chỉnh sửa.
- Trường tìm kiếm.

Dưới đây là class diagram cho class này. Hình 11



ctrl_item
- buttNewItem: Button - fieldSearch: TextBox - itemTable: DataGrid - refreshButton: Button
+ ctrl_item() - buttNewItem_Click() - fieldSearch_OnValueChanged() - itemTable_CellContentClick() - load_item() - refresh_button_Click() - search_item()

Hình 11: Lược đồ lớp của class ctrl_item

Chi tiết hiện thực các hàm chức năng trong class này sẽ được chi tiết ở phần trình bày Chức năng và Giao diện.



3 Sơ lược về Cơ sở dữ liệu

Vì phần lớn thành viên trong nhóm chưa được học qua môn CSDL và các khái niệm cơ bản của CSDL cần thiết. Nên nhóm đã quyết định hiện thực CSDL một cách đơn giản như sau:

- Các trường ID được đánh số nguyên tự động tăng.
- Các trường Tên, Giá cả,... đều sử dụng kiểu chuỗi nvarchar.
- Các trường liên quan đến ngày tháng sử dụng kiểu Date.
- Không có khóa ngoại và các ràng buộc trên CSDL. Mọi ràng buộc sẽ được hiện thực trong code.

Nhóm đã tạo các bảng sau để quản lý hệ thống:

- Account: Quản lý thông tin đăng nhập.
- AccountInfo: Quản lý thông tin tài khoản.
- Bill: Quản lý các đơn hàng.
- BillDetail: Quản lý thông tin chi tiết của các đơn hàng.
- Item: Quản lý thông tin các mặt hàng (ở đây là các loại đồ uống).
- Category: Quản lý thông tin các loại mặt hàng.

Cơ sở dữ liệu này được hiện thực trên Microsoft SQL Server 2014. Vì số lượng các bảng khá lớn và có sự tương tự nhau nên nhóm sẽ chỉ trình bày một số bảng: Account, Item, Bill, BillDetail và cách tương tác giữa code và CSDL.

3.1 Bảng quản lý Account

Bảng Account bao gồm các field sau:

- AccountID: Số duy nhất định danh cho tài khoản.
- username: username dùng để đăng nhập.
- displayname: tên hiển thị của tài khoản.
- password: mật khẩu đăng nhập.
- type: loại tài khoản (staff hay admin).

Dưới đây là mã SQL khởi tạo bảng này.

```
CREATE TABLE Account
(
    AccountID int identity primary key,
    username nvarchar(100) not null,
    displayname nvarchar(100) not null,
    password nvarchar(100) not null,
    type int not null
)
Go
```



3.2 Bảng Quản lý Item

Bảng Item bao gồm các field sau:

- ItemID: Số duy nhất định danh cho Item.
- ItemName: Tên Item.
- ItemPrice: Giá của Item.
- ItemCategory: Loại của Item, mặc định có 8 loại.
- ItemImage: Hình ảnh của Item.

Dưới đây là mã SQL khởi tạo bảng này.

```
CREATE TABLE Item
(
    ItemID int identity primary key,
    ItemName nvarchar(100) not null,
    ItemPrice nvarchar(50) not null,
    ItemCategory nvarchar(50) not null,
    ItemImage nvarchar(50) not null
)
Go
```

3.3 Bảng Quản lý Bill

Bảng Bill bao gồm các field sau:

- BillID: Số duy nhất định danh cho đơn hàng.
- BillStaffID: ID của nhân viên chịu trách nhiệm đơn hàng này.
- BillStaffName: tên của nhân viên chịu trách nhiệm đơn hàng này.
- BillDate: thời điểm đơn hàng được tạo.
- BillPrice: giá tiền tổng cộng của đơn hàng.

Dưới đây là mã SQL khởi tạo bảng này.

```
CREATE TABLE Bill
(
    BillID int identity primary key,
    BillStaffID int not null,
    BillStaffName nvarchar(100) not null,
    BillDate date not null,
    BillPrice nvarchar(100) not null
)
Go
```



3.4 Bảng Quản lý BillDetail

Vì một hóa đơn có thể bao gồm nhiều item nên bảng BillDetail ra đời để quản lý các item trong Bill. Bảng BillDetail bao gồm các field sau:

- BillDetailID: Số duy nhất định danh cho món hàng trong đơn hàng.
- BillID: ID của đơn hàng.
- ItemID: ID của đồ uống / đồ ăn.
- ItemName: Tên của đồ uống / đồ ăn.
- ItemQuantity: Số lượng.
- BillDetailPrice: tổng tiền cho riêng món này.

Dưới đây là mã SQL khởi tạo bảng này.

```
CREATE TABLE BillDetail
(
    BillDetailID int identity primary key,
    BillID int not null,
    ItemID int not null,
    ItemName nvarchar(100) not null,
    ItemQuantity int not null,
    BillDetailPrice nvarchar(100) not null
)
Go
```

3.5 Thao tác với Cơ sở dữ liệu

Để thao tác trực tiếp với CSDL, nhóm đã tạo ra 1 class `DataSupport` dùng chung cho tất cả những câu truy vấn với Account, Bill, Item,... Đầu vào là một câu truy vấn SQL và các tham số (nếu có), đầu ra của nó sẽ là một bảng dữ liệu được trả về từ CSDL. Chi tiết class như sau:

```
public class DataSupport{
    private static DataSupport instance;
    public static DataSupport INSTANCE {
        get {
            if (instance == null) instance = new DataSupport();
            return DataSupport.instance;
        }
        set {
            DataSupport.instance = value;
        } // Outside cannot set value.
    }

    private DataSupport() { }

    private string connStr = "Data Source=LUAN-PC\\SQLEXPRESS;
```



```
Initial Catalog=CoffeeShop;Integrated Security=True";  
  
public DataTable ExecuteQuery(string query, object[] parameter = null) {  
    DataTable data = new DataTable();  
  
    using (SqlConnection conn = new SqlConnection(connStr)) {  
        conn.Open();  
        SqlCommand sqlCommand = new SqlCommand(query, conn);  
        SqlDataAdapter adapter = new SqlDataAdapter(sqlCommand);  
        adapter.Fill(data);  
        conn.Close();  
    }  
  
    return data;  
}  
}
```



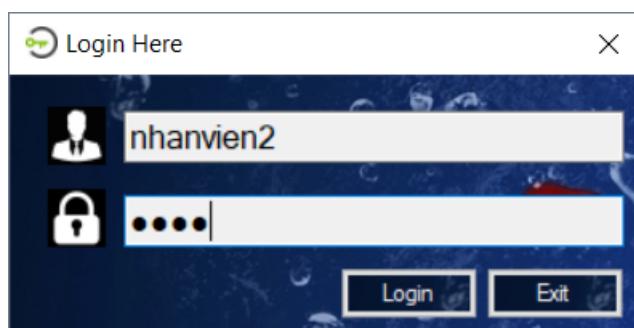
4 Các chức năng và giao diện của ứng dụng

4.1 Phần Point-of-Sale

Phần này em sẽ trình bày theo thứ tự thực tế khi mở ứng dụng, đồng thời mô tả mã hiện thực của một số Form, Class và hàm sử dụng để minh họa.

4.1.1 Đăng nhập (FormLogin)

Đây là màn hình đầu tiên khi khởi chạy chương trình, nó yêu cầu người dùng nhập tài khoản và mật khẩu chính xác (được lưu trong cơ sở dữ liệu) để tiếp tục chương trình.



Hình 12: Giao diện đăng nhập, ngõ vào của chương trình

Khi click nút Login, chương trình sẽ gọi hàm bool Login(string account, string password), account và password lấy từ 2 text box trên Form, nếu hàm trả về true thì cho phép truy cập, ngược lại thông báo nhập lại tài khoản, mật khẩu.

Để so sánh tài khoản, mật khẩu với cơ sở dữ liệu, em sử dụng class AccountDAO, trong đó hiện thực hàm AccountDAO.Instance.Login(account, password), hàm này trả về giá trị trong hàm Login của FormLogin.

Khi click nút Cancel, chương trình sẽ thoát.

Ngoài ra em còn sử dụng một biến static int staticAccountID để dùng cho việc xuất hóa đơn sau này.

Mã hiện thực hàm btnLogin_Click và Login của FormLogin:

```
private void btnLogin_Click(object sender, EventArgs e) {
    string username = txtUsername.Text;
    string password = txtPassword.Text;
    if (Login(username, password)) {
        FormStaff f = new FormStaff();
        this.Hide();
        f.ShowDialog();
        this.Show();
    }
}
```



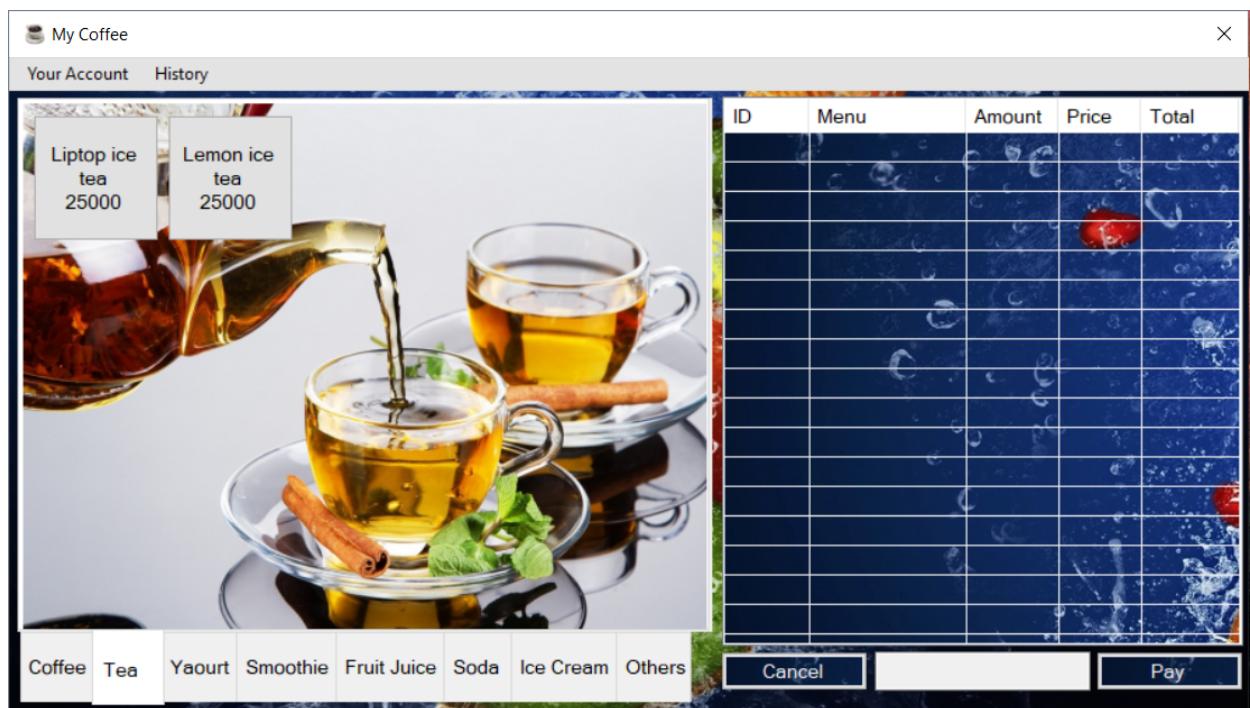
```
    } else {
        MessageBox.Show("The username or password you entered is
incorrect!");
    }
}
bool Login(string account, string password) {
    return AccountDAO.Instance.Login(account, password);
}
```

Mã hiện thực hàm Login của class AccountDAO:

```
public bool Login(string userName, string passWord) {
    string query = "SELECT * FROM dbo.Account WHERE userName =
N'" + userName + "' AND passWord = N'" + passWord + "'";
    DataTable result = DataProvider.Instance.ExecuteNonQuery(query);
    //lấy AccountID
    foreach (DataRow item in result.Rows) {
        FormLogin.staticAccountId = Convert.ToInt32(item[0].ToString());
    }
    return result.Rows.Count > 0;
}
```

4.1.2 Giao diện chính của ứng dụng (FormStaff)

Sau khi đăng nhập thành công, màn hình chính sẽ hiển thị, đây là nơi nhân viên duyệt món, nhập hóa đơn khi khách hàng gọi món, cũng từ đây nhân viên có thể xem lại lịch sử bán hàng, cũng như thông tin cá nhân của mình (mục 4.1.5 và 4.1.6).



Hình 13: Giao diện chính

Trước tiên em lấy danh sách các món nước trong cơ sở dữ liệu (hàm LoadItemList() trong class ItemDAO)

Để hiển thị các nước uống lên các tab page em chuyển chúng thành các button (hàm LoadItemToButtons()) và sử dụng các flow layout panel để sắp xếp chúng.

Mã hiện thực hàm LoadItemList() trong class ItemDAO:

```
public List<Item> LoadItemList() {
    List<Item> ListItem = new List<Item>();
    string query = "SELECT * FROM dbo.Item";
    DataTable data = DataProvider.Instance.ExecuteNonQuery(query);
    foreach (DataRow item in data.Rows) {
        Item itemData = new Item(item);
        ListItem.Add(itemData);
    }
    return ListItem;
}
```

Mã hiện thực hàm LoadItemToButtons() trong FormStaff (đã rút gọn):

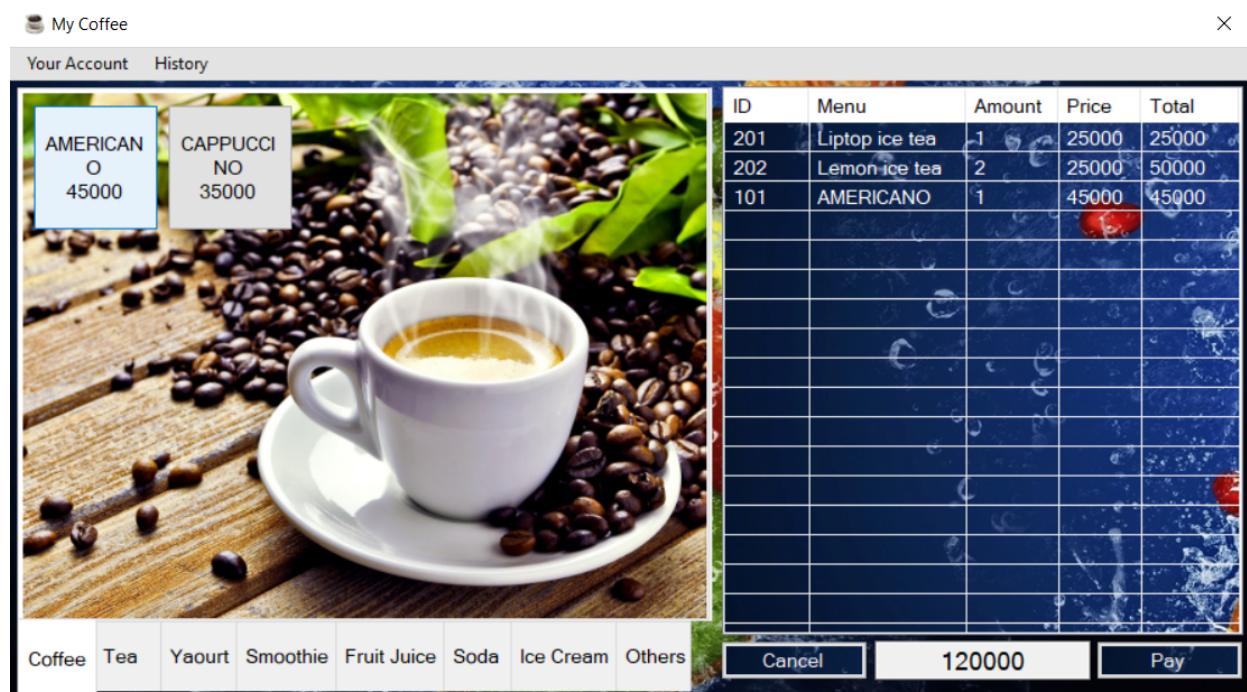
```
'void LoadItemListToButtons() {
//-----Lấy danh sách tất cả Item
```



```
List<Item> ItemList = ItemDAO.Instance.LoadItemList();
//-----Tab1 = danh sách item theo phân loại thứ I
foreach (Item item in ItemList) {
    if (item.CATEGORY == "C1") {
        Button btn = new Button() { Width = ItemDAO.btnWidth, Height = ItemDAO.btnHeight };
        btn.Text = item.NAME + Environment.NewLine + item.PRICE;
        btn.Click += btn_Click; //Hàm btn_Click (thêm item vào bill)
        btn.Tag = item; //lưu lại để lấy thông tin Item từ button.tag
        dùng sau này (btn_Click)
        flpItemC1.Controls.Add(btn);
    }
}
//---Tab2 = danh sách item theo phân loại thứ II, tương tự với các phân
loại khác
}
```

4.1.3 Chọn món - hiển thị lên hóa đơn

Khi click chọn món (các button) sẽ hiển thị lên hóa đơn bên cạnh:



Hình 14: Nhập món và hiển thị lên hóa đơn

Lúc này chương trình gọi hàm `btn_Click` để lấy thông tin Item từ button đó, sau đó gọi hàm `LoadItemToListView` để hiển thị chúng lên hóa đơn cũng như tổng tiền hóa đơn. Nếu click nút

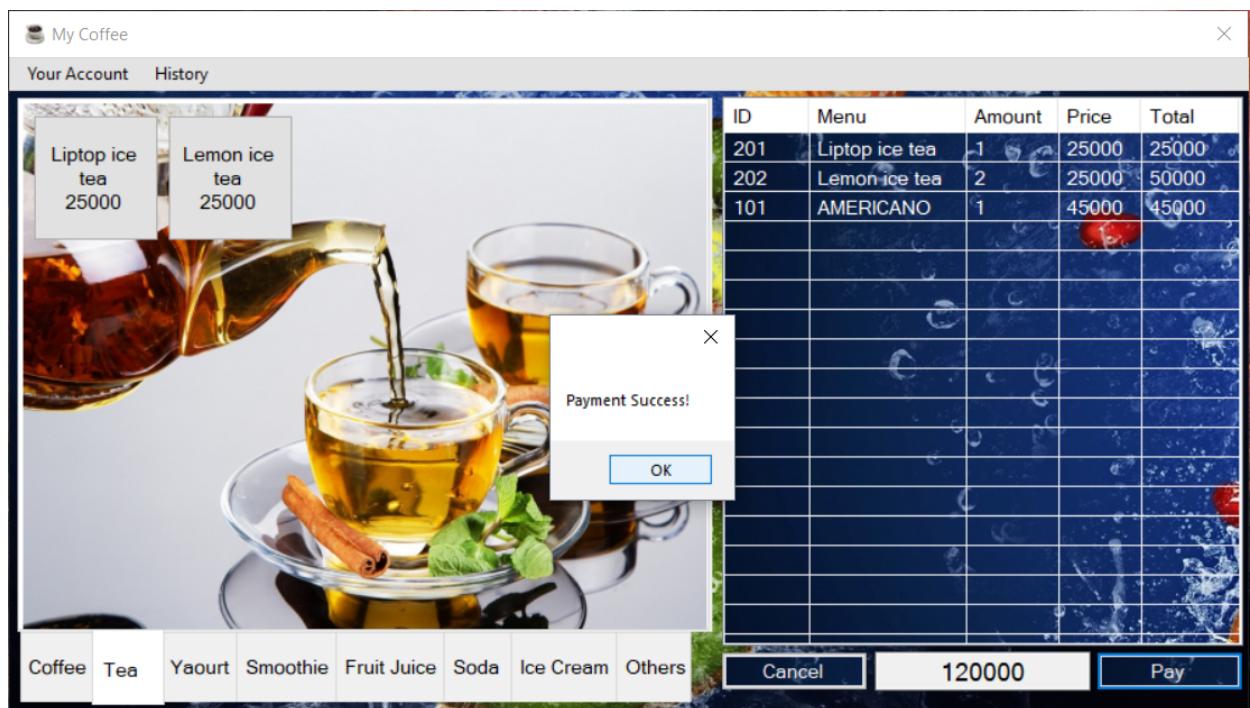


Cancel thì hóa đơn sẽ được hủy. Mã hiện thực hàm btn_Click:

```
void btn_Click(object sender, EventArgs e) {
    try
    {
        //I_lấy thông tin item từ button.tag đã lưu
        string itemID = ((sender as Button).Tag as Item).ID;
        string itemName = ((sender as Button).Tag as Item).NAME;
        string itemPrice = ((sender as Button).Tag as Item).PRICE;
        //II_thêm món vào hóa đơn
        LoadItemToListView(lsvBill, itemID, itemName, itemPrice);
        //III_tổng tiền của hóa đơn
        int totalBill = 0;
        for (int nItem = 0; nItem < lsvBill.Items.Count; nItem++)
        {
            int totalPrice =
                Convert.ToInt32(lsvBill.Items[nItem].SubItems[4].Text.ToString());
            totalBill = totalBill + totalPrice;
        }
        txtTotalBill.Text = totalBill.ToString();
    }
    catch
    {
        return;
    }
}
```

4.1.4 Chọn thanh toán – lưu hóa đơn

Khi click Pay hóa đơn sẽ được thanh toán, đồng thời nó được lưu vào cơ sở dữ liệu



Hình 15: Thông báo thanh toán thành công

Lúc này chương trình gọi hàm `btnPay_Click`, trong đó gọi 2 hàm `InsertBill` và `InsertBillDetail` để lưu hóa đơn, hóa đơn chi tiết vào cơ sở dữ liệu.

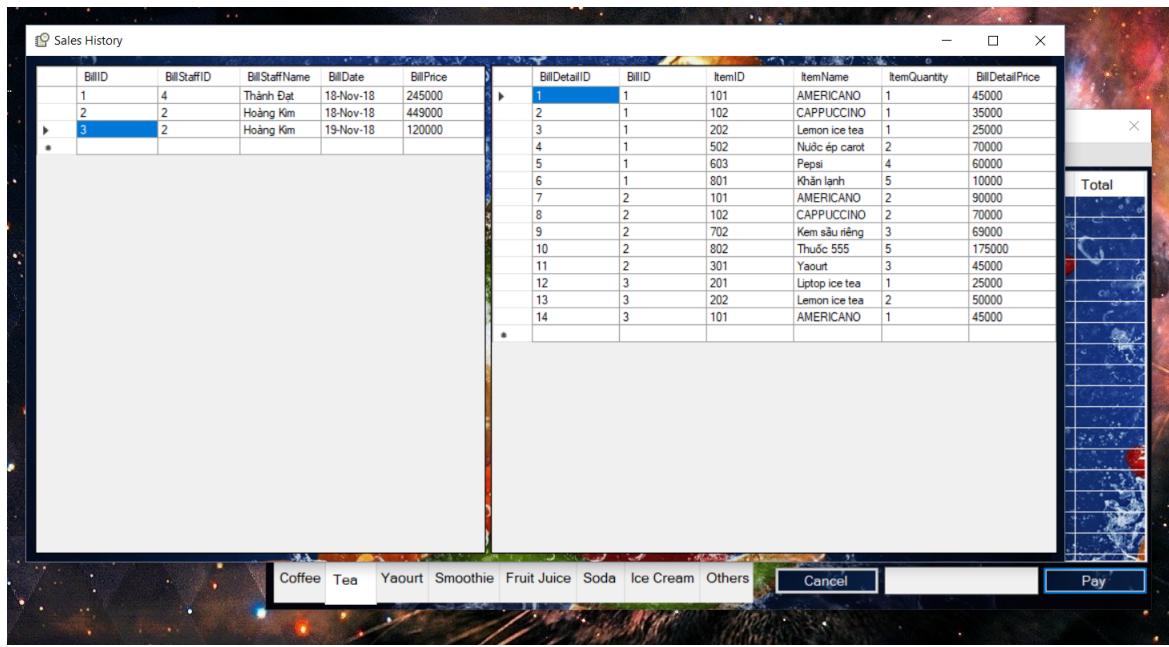
Mã nhân viên bán hàng (`idStaff`) lúc này chính là biến `staticAccountID` đã lấy từ `FormLogin`. Mã hiện thực hàm `InsertBill` của class `BillDAO` (InsertBillDetail tương tự):

```
public void InsertBill(int idBill, int idStaf, string totalBill) {
    string staffName = "null";
    //lấy NameStaff từ AccountID
    string query = "SELECT AccountName FROM dbo.AccountInfo WHERE
    AccountID = " + FormLogin.staticAccountID;
    DataTable nameStaff = DataProvider.Instance.ExecuteNonQuery(query);
    foreach (DataRow item in nameStaff.Rows)
    {
        staffName = item[0].ToString();
    }
    //thêm bill mới vào Database
    DataProvider.Instance.ExecuteNonQuery("EXEC dbo.USP_InsertBill
    @idBill , @idStaff , @staffName , @priceBill ", new object[] { idBill, idStaf,
    staffName, totalBill });
}
```



4.1.5 Xem lại lịch sử bán hàng - FormHistoryBill

Màn hình lịch sử bán hàng sẽ hiển thị khi người dùng chọn mục History trong menu strip:



Hình 16: Giao diện xem lịch sử bán hàng

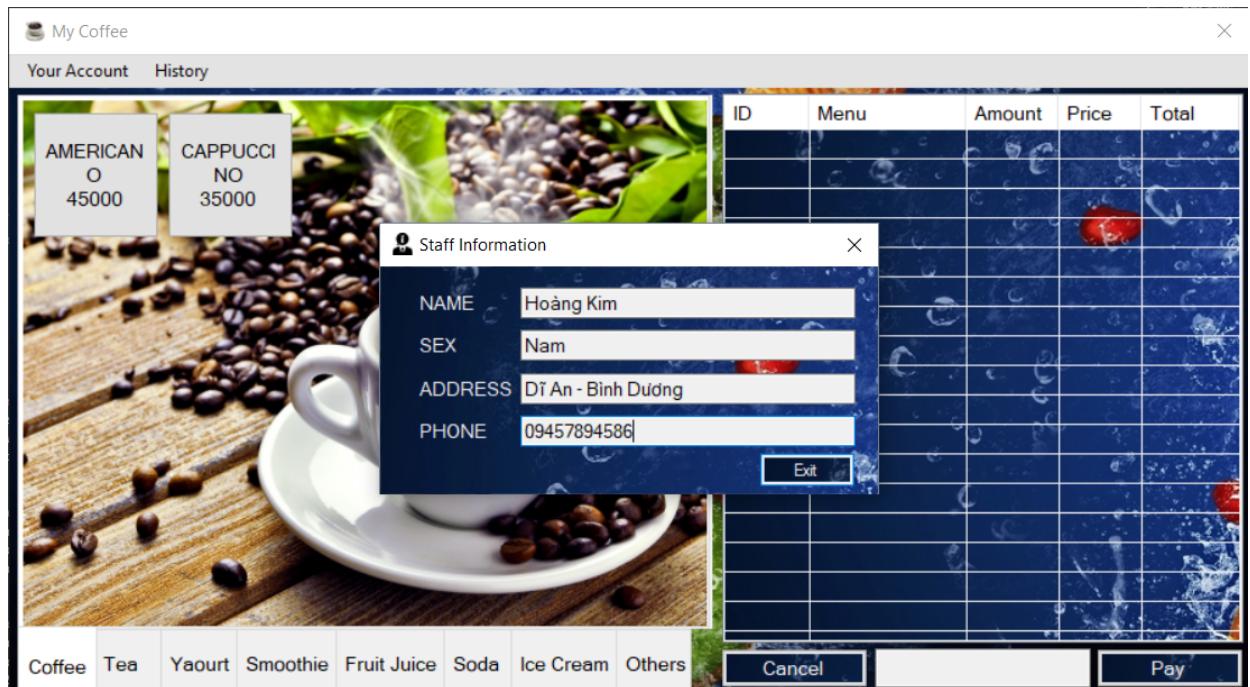
Form này sử dụng 2 data grid view để hiện thị 2 bảng Bill và BillDetail trong cơ sở dữ liệu.

Mã hiện thực của FormHistoryBill:

```
public partial class FormHistoryBill : Form {
    public FormHistoryBill() {
        InitializeComponent();
        LoadBill();
    }
    void LoadBill() {
        string query = "SELECT * FROM dbo.Bill";
        dtgViewBill.DataSource = DataProvider.Instance.ExecuteNonQuery(query);
        string query2 = "SELECT * FROM dbo.BillDetail";
        dtgViewBillDetail.DataSource =
            DataProvider.Instance.ExecuteNonQuery(query2);
    }
}
```

4.1.6 Xem thông tin cá nhân – FormAccountInfo

Khi muốn xem lại thông tin của mình, người dùng click chọn mục Your Account > Information trong menu strip:



Hình 17: Giao diện xem thông tin cá nhân của nhân viên

Form này gồm 4 text box để hiển thị lần lượt thông tin về tên, giới tính, địa chỉ, số điện thoại của nhân viên và nút exit để đóng form.

Lúc này em sử dụng tiếp biến staticAccountID ở FormLogin để lấy thông tin của nhân viên, sau đó gán từng trường vào các text box tương ứng.

Mã hiện thực của hàm LoadAccountInfoByID:

```
void LoadAccountInfoByID() {
    string query = "SELECT AccountName FROM dbo.AccountInfo WHERE
    AccountID = " + FormLogin.staticAccountID.ToString();
    DataTable data = DataProvider.Instance.ExecuteNonQuery(query);
    foreach (DataRow item in data.Rows) {
        txtName.Text = item[0].ToString();
    }
    string query2 = "SELECT AccountSex FROM dbo.AccountInfo WHERE
    AccountID = " + FormLogin.staticAccountID.ToString();
    DataTable data2 = DataProvider.Instance.ExecuteNonQuery(query2);
    foreach (DataRow item in data2.Rows) {
        txtSex.Text = item[0].ToString();
    }
    string query3 = "SELECT AccountAddress FROM dbo.AccountInfo
    WHERE AccountID = " + FormLogin.staticAccountID.ToString();
    DataTable data3 = DataProvider.Instance.ExecuteNonQuery(query3);
```

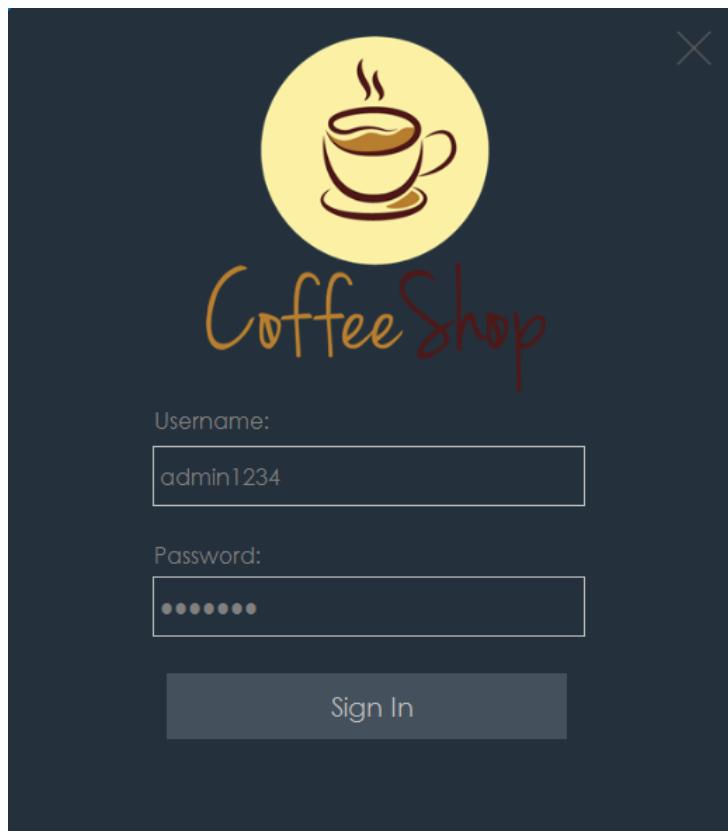


```
foreach (DataRow item in data3.Rows) {  
    txtAddress.Text = item[0].ToString();  
}  
string query4 = "SELECT AccountPhone FROM dbo.AccountInfo WHERE  
AccountID = " + FormLogin.staticAccountID.ToString();  
DataTable data4 = DataProvider.Instance.ExecuteQuery(query4);  
foreach (DataRow item in data4.Rows)  
{  
    txtPhone.Text = item[0].ToString();  
}  
}
```

4.2 Phần ứng dụng hỗ trợ quản lý

4.2.1 Đăng nhập

Đây là màn hình đầu tiên hiện ra khi khởi động ứng dụng, cho phép người dùng nhập tài khoản của họ để truy cập vào ứng dụng:



Hình 18: Giao diện đăng nhập tài khoản quản lý



Dưới đây là mã được kích chạy khi click vào nút Sign In:

```
string username = fieldUserName.Text;
string password = fieldPassword.Text;

// Check empty username && password
if (username == "" || password == "")
{
    labelWarning.Text = "* NOTICE: You must enter both username
                        and password to login!";
    labelWarning.ForeColor = System.Drawing.Color.OrangeRed;
    fieldUserName.Text = "";
    fieldPassword.Focus();
}

// If the login is successs
else if (Account.INSTANCE.login(username, password))
{
    Program.login.Hide();
    Program.main.Show();
}

// login fail
else
{
    labelWarning.Text = "Login Failed!";
    labelWarning.ForeColor = System.Drawing.Color.OrangeRed;
    fieldPassword.Focus();
}
```

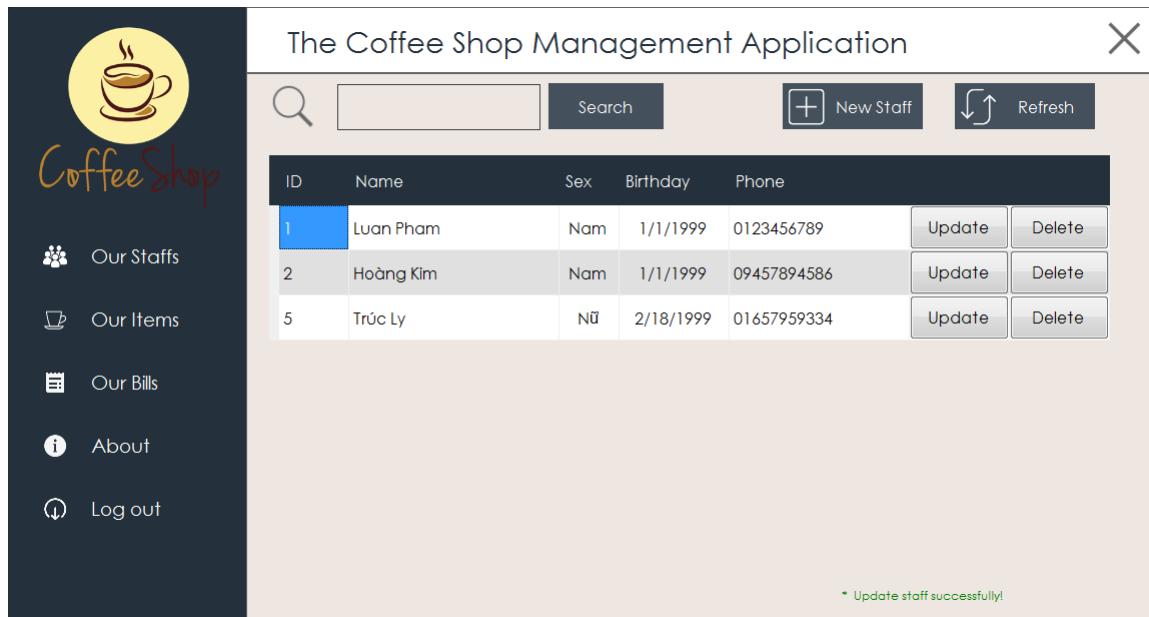
Trong đoạn mã trên, sau khi kiểm tra username và password, chương trình gọi một đoạn mã khác "Account.INSTANCE.login(username, password)", dưới đây là mã của hàm chức năng này.

```
public bool login(string username, string password){
    string query = "SELECT * FROM ACCOUNT WHERE USERNAME = '" +
                   + username + "' AND PASSWORD = '" + password + "'";
    DataTable result = DataSupport.INSTANCE.ExecuteQuery(query,
        new object[] { username, password });
    return result.Rows.Count > 0;
}
```

Đoạn mã trên sẽ tạo ra câu Query thao tác với CSDL, và gọi hàm chức năng đã tạo từ trước trong class DataSupport.

4.2.2 Màn hình thao tác chính

Màn hình thao tác chính bao gồm 1 sidebar bên trái bao gồm logo và các nút bấm redirect sang các trang nhìn và nút Logout, header là title của ứng dụng, phía dưới là panel chứa các trang nhìn. Trong hình dưới là trang nhìn quản lý nhân viên.



Hình 19: Giao diện thao tác chính

4.2.3 Màn hình thêm tài khoản / thêm item

(a) Màn hình thêm item

(b) Màn hình đăng ký tài khoản

Hình 20: Hai màn hình thêm dữ liệu vào CSDL



4.2.4 Màn hình chi tiết hóa đơn

Detail Infomations for Bill ID #1 X			
Bill ID #1			
Staff ID #2			
Staff Name : Hoàng Kim			
Date : 11/13/2018 12:00:00 AM			
Item ID	Item Name	Quantity	Amount
2	CAPPUCCINO	1	35000
7	BLUBERRY SMOOTHIE	1	59000
6	PEACH TEA MANIA	1	45000
Total : 139000 vnd			

Hình 21: Giao diện thao tác chính



5 Tổng kết

Qua bài tập lớn này, nhóm đã nâng cao được kỹ năng lập trình C# và thao tác với Windows Form trên Visual Studio, đồng thời cũng có thêm kiến thức về Microsoft SQL Server để xây dựng CSDL, Framework Bunifit để xây dựng giao diện. Ngoài ra nhóm cũng nâng cao được kỹ năng làm việc và quản lý nhóm.

Tuy vậy vì lượng thời gian và khả năng của nhóm có hạn, do còn nhiều môn học khác và kiến thức chưa đủ, nên hệ thống vẫn còn nhiều nhược điểm có thể được cải thiện sau này để hệ thống có thể tốt hơn:

- Lúc kiểm tra username, password, chưa validate tránh SQL Injection.
- Lúc kiểm tra cái field email, sđt, hình ảnh url, chưa kiểm tra chặt tính đúng đắn của nó.
- Nhóm ban đầu tính load hình ảnh từ url, hoặc upload ảnh trực tiếp trên CSDL, nhưng về sau vì không có đủ thời gian nên tính năng này không được hiện thực.

Một số khó khăn gặp phải trong khi làm việc và quản lý nhóm là vẫn chưa sử dụng các công cụ quản lý nhóm và code hiệu quả như Git, Trello,... Nhóm liên lạc và quản lý nhau chủ yếu trên Facebook Group và Facebook Messenger. Ngoài ra vì lịch học của các thành viên khác nhau, một số thành viên có đi làm thêm ở các khung giờ khác nhau, nên số lượng các buổi gặp mặt trực tiếp để cùng làm với nhau không thật sự nhiều.



6 Phân công công việc trong nhóm

Thành viên	Phản tinh hiểu
Phạm Quý Luận	<ol style="list-style-type: none">Góp ý và chọn đề tài.Xây dựng và phát triển CSDL.Xây dựng ứng dụng hỗ trợ quản lý phía admin.Viết report.Làm video.
Dinh Hoàng Kim	<ol style="list-style-type: none">Góp ý và chọn đề tài.Cập nhật và phát triển CSDL.Xây dựng ứng dụng Point-Of-Sale.Viết report.Làm video.
Nguyễn Thị Trúc Ly	<ol style="list-style-type: none">Góp ý và chọn đề tài.Xây dựng ứng dụng Point-Of-Sale.Viết report.Làm slide.
Phan Thành Đạt	<ol style="list-style-type: none">Góp ý và chọn đề tài.Hỗ trợ góp ý và xây dựng CSDL.



Tài liệu tham khảo

- [1] Bunifu Framework ([Link Homepage](#))
- [2] C# Documentations ([Link Homepage](#))
- [3] Windows Form Documentations ([Link Homepage](#))
- [4] Icon8 for Flat Icons ([Link Homepage](#))
- [5] Just Color Picker ([Link Homepage](#))