

|  |
| --- |
| **FPT Polytechnic** |
| **DỰ ÁN 1**  **PHẦN MỀM QUẢN LÝ QUÁN CÀ PHÊ** |

|  |
| --- |
| NGÀNH CÔNG NGHỆ THÔNG TIN (ỨNG DỤNG PHẦN MỀM |

|  |
| --- |
| GIẢNG VIÊN HƯỚNG DẪN: NGUYỄN ĐỖ ANH KHOA |
| SINH VIÊN THỰC HIỆN |
| 1. PS08674 – NGUYỄN QUỐC PHONG   Trưởng nhóm   1. PS08771 – HỒ TRẦN BẢO CHÂU 2. PS08676 – NGUYỄN HOÀNG MINH 3. PS08675 – TẠ HUY THỊNH 4. PS08785 – NGUYỄN HUY HOÀNG |

**MỤC LỤC**

[1 PHÂN TÍCH 3](#_Toc27973946)

[1.1 HIỆN TRẠNG 3](#_Toc27973947)

[1.2 YÊU CẦU HỆ THỐNG 3](#_Toc27973948)

[1.3 USE CASE 3](#_Toc27973949)

[2 THIẾT KẾ 6](#_Toc27973950)

[2.1 MÔ HÌNH TRIỂN KHAI 6](#_Toc27973951)

[2.2.1 Sơ đồ quan hệ thực thể 7](#_Toc27973952)

[2.2.2 Thiết kế chi tiết các thực thể 8](#_Toc27973953)

[2.3 THIẾT KẾ GIAO DIỆN 12](#_Toc27973954)

[2.3.1 Sơ đồ tổ chức giao diện 12](#_Toc27973955)

[2.3.2 Thiết kế giao diện cho các chức năng nghiệp vụ 13](#_Toc27973956)

[3 THỰC HIỆN VIẾT MÃ 40](#_Toc27973957)

[3.1.1 Tạo CSDL 41](#_Toc27973958)

[93.1.1 SQL truy vấn và thao tác bảng 42](#_Toc27973959)

[3.1.3 Thêm dữ liệu 46](#_Toc27973960)

[3.2.1 Lớp hỗ trợ 47](#_Toc27973961)

[3.2.2 Model class - Các lớp mô tả dữ liệu 52](#_Toc27973962)

[3.2.3 DAO Class - Các lớp truy xuất dữ liệu 68](#_Toc27973963)

[3.3 VIẾT MÃ CHO ỨNG DỤNG 92](#_Toc27973964)

[3.3.1 Các lớp tiện ích 92](#_Toc27973965)

[3.3.2 Viết mã cho các chức năng 96](#_Toc27973966)

[4 KIỂM THỬ 97](#_Toc27973967)

[5 ĐÓNG GÓI VÀ TRIỂN KHAI 97](#_Toc27973968)

[5.3 HƯỚNG DẪN CHUYỂN ĐỔI JAR THÀNH EXE 97](#_Toc27973969)

[5.4 HƯỚNG DẪN CÀI ĐẶT TRIỂN KHAI 97](#_Toc27973970)

[5.5 HƯỚNG DẪN SỬ DỤNG PHẦN MỀM 97](#_Toc27973971)

# PHÂN TÍCH

### HIỆN TRẠNG

##### Các quán cà phê nói chung hiện nay đang cần 1 hệ thống có thể quản lý tất cả mọi thứ cho quán minh.Do việc quản lý đơn hàng , sản phẩm, số lượng,khách hàng và doanh thu đang thực hiện thông qua excel. Hiện quán đang gặp khó khăn khi số lượng khách hàng ngày một nhiều và số lượng sản phẩm ngày càng đa dạng, dữ liệu ngày càng lớn nên việc quản lý excel gặp rất nhiều khó khăn, dễ sai sót mà không bảo mật.

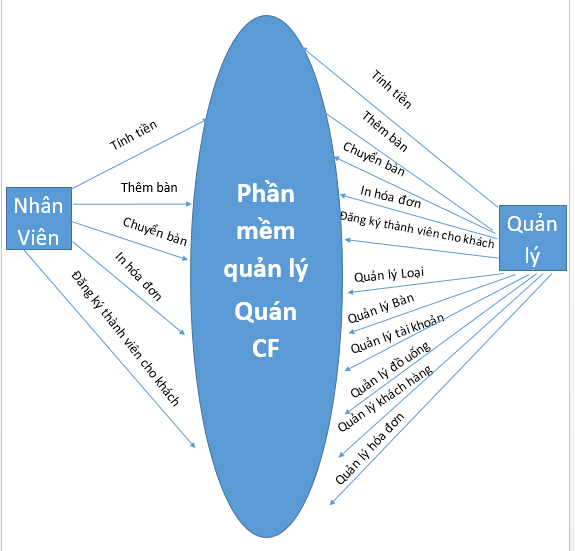
### YÊU CẦU HỆ THỐNG

Quán mong muốn xây dựng một phần mềm để giải quyết khó khăn trên.

* + - **Yêu cầu chức năng nghiệp vụ**
      * Quản lý menu
      * Quản lý nhân viên
      * Quản lý thành viên
      * Quản lý hóa đơn
      * Quản lý loại
      * Quản lý kho
      * Xem thống kê
    - **Yêu cầu về bảo mật**
      * Chỉ nhân viên và quản lý mới có thể đăng nhập hệ thống
      * Quản lý được phép thực hiện tất cả các chức năng
      * Nhân viên bị hạn chế quyền hạn
    - **Yêu cầu về môi trường 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 11
      * Hệ quản trị CSDL SQL Server 2008 trở lên

### USE CASE

Use case là sơ đồ tổng quan về mặt chức năng và phân vai trò người sử dụng. Dựa vào yêu cầu hệ thống của khách hàng, chúng ta có thể phác thảo sơ đồ use case như sau.



* + - **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 chức 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
        + Tổng hợp 5 loại đồ uống được sử dụng nhiều nhất
        + Khách hàng thường xuyên ghé quan
      * Chức năng đăng nhập và đăng xuất
        + Đăng nhập: để sử dụng phần mềm, bắt buộc phải đăng nhập
        + Đăng xuất: đăng nhập lại với người dùng khác hoặc dừng lại đi đâu đó
    - **CHI TIẾT VỀ YÊU CẦU BẢO MẬT**

##### Tất cả phải đăng nhập mới được sử dụng các chức năng trong phần mềm

* + - * Quản lý: được phép sử dụng tất cả
      * Nhân viên:
        + Không được phép thực hiện chức năng xóa trong các chức năng quản lý
        + Không được phép xem chức năng thống kê doanh thu
    - **THÔNG TIN CÁC THỰC THỂ**

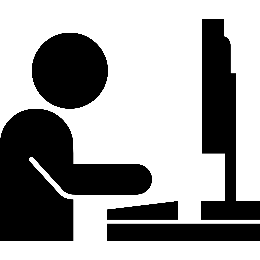
##### Order Number

* + - * + ID
        + Status
      * Bill
        + ID
        + Date Check
        + Order Number
        + Customer
        + Staff Name
        + Totalprice
      * Bill Info
        + ID Bill
        + ID Drinks
        + Count Item
        + Price
      * Account
        + Username
        + Display Name
        + PassWord
        + Type
      * Drinks
        + ID
        + Name
        + Idcategory
        + Price
      * Category
        + ID
        + Name
      * Storage
        + iddrink
        + quantity
      * Customer
        + Phone
        + Name
        + point

# THIẾT KẾ

### MÔ HÌNH TRIỂN KHAI

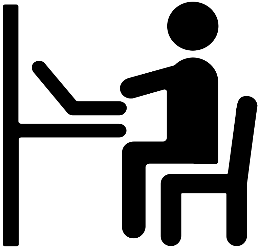
Ứng dụng phần mềm được xây dựng để phục vụ cho nhiều người dùng nhưng cơ sở dữ liệu thì lưu trữ tập trung.



Manage System



DB SQL



[This Photo](https://commons.wikimedia.org/wiki/File:Noun_project_-_Working_with_laptop.svg) by Unknown Author is licensed under [CC BY-SA](https://creativecommons.org/licenses/by-sa/3.0/)

Manage System

* + - Cần một máy cài SQL Server 2008+. Máy này cần hệ điều hành windows 7 trở lên
    - Máy dùng tính tiền phải được cài phần mềm Coffe Shop
  1. **THIẾT KẾ CSDL**

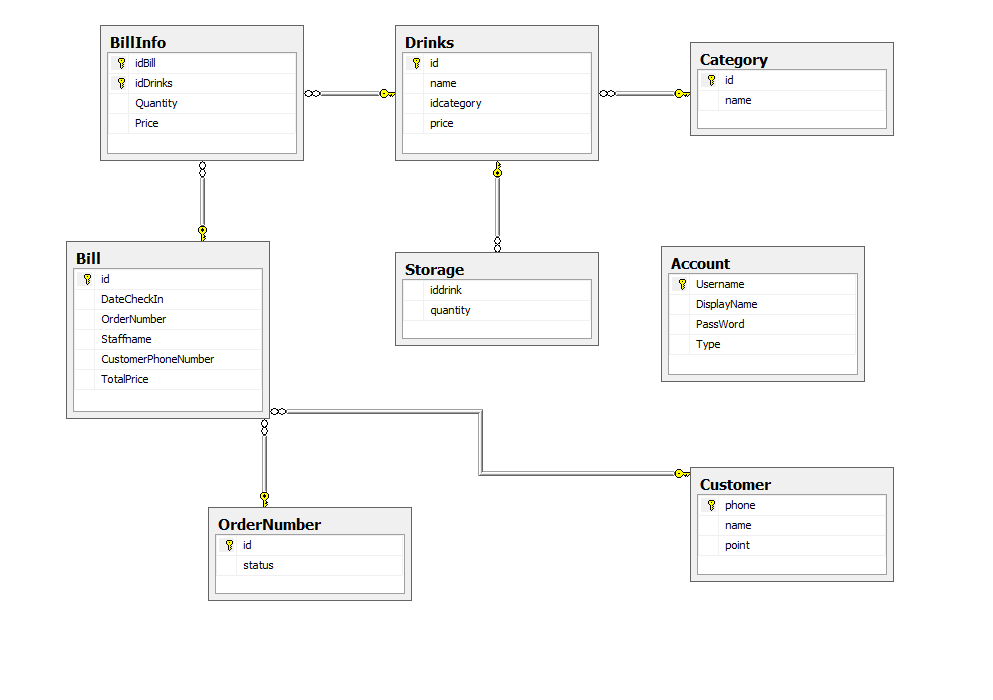
## Sơ đồ quan hệ thực thể

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

* + - * Bill: quản lý hóa đơn
      * BillInfo: Quản lý hóa đơn chi tiết quan hệ 1-1 với Bill
      * Drinks: Quản lý các đồ uống, quan hệ một-nhiều với Bill (một hóa đơn có thể nhiều đồ uống, 1 đồ uống có thể nhiều hóa đơn)
      * Category: quản lý các loại của đồ uống (một đồ uống chỉ có 1 loại, 1 loại thì có nhiều đồ uống)
      * Storage: Kho (1 kho có thể có nhiều loại nước)
      * Customer : Khách hàng (một khách hàng thì có nhiều Bill, nhưng 1 Bill chỉ 1 khách hàng)
      * OrderNumber: Số thứ tự order (1 Bill có nhiều orderNumber, 1 orderNumber chỉ có 1 Bill quan hệ 1-1)

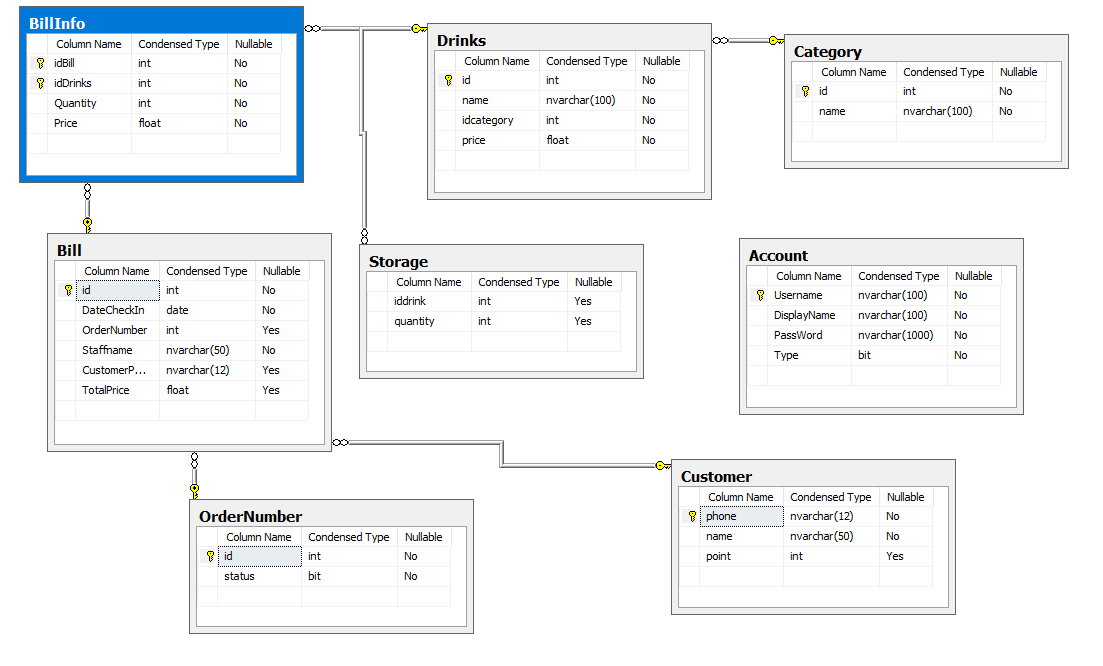
Sau khi phân tích chúng ta có thể thấy các thực thể có mối quan hệ với nhau như sơ đồ ERD sau

* + - 1. ***ERD Diagram level 1***



* + - 1. ***ERD Diagram level 2***

**Yêu cầu ràng buộc quan hệ:**



* + - * + Tất cả các relationship đều là CASCADE (cập nhật dây chuyền)

## Thiết kế chi tiết các thực thể

* + - 1. ***Account***

Bảng Account lưu thông tin quản lý ,nhân viên có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| Username | NVARCHAR(100) | PK, NOT NULL | Tên nhân viên |
| DisplayName | NVARCHAR(100) | NOT NULL | Tên hiển thị |
| PassWord | NVARCHAR(100) | NOT NULL | Mật khẩu |
| Type | BIT | NOT NULL | Kiểu dữ liệu |

* + - 1. ***Bill***

Bảng Bill lưu thông tin các Hóa đơn có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Rằng buộc** | **Ghi chú** |
| id | INT | PK, NOT NULL | ID |
| DateCheckIn | DATE | NOT NULL | Ngày đến |
| OrderNumber | INT | FK, NULL | Số thứ tự |
| Staffname | NVARCHAR(50) | NOT NULL | Tên nhân viên phụ trách |
| customer | NVARCHAR(12) | FK, NULL | Khách hàng |
| totalprice | FLOAT | NULL | Tổng hóa đơn |
|  |  |  |  |

* + - 1. ***BillInfo***

Bảng BillInfo lưu thông tin chi tiết của hóa đơn có cấu trúc như sau:

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| idBill | INT | PK,FK, NOT NULL | Id của hóa đơn |
| idDrinks | INT | PK,FK,NOT NULL | Sản phẩm đã sử dụng |
| Quantity | INT | NOT NULL | Số lượng |
| Price | FLOAT | NOT NULL | Tổng tiền |

* + - 1. ***Category***

Bảng Category lưu :

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| id | INT | PK,NOT NULL | Id |
| name | NCHAR(100) | NOT NULL | Tên loại |

* + - 1. ***Customer***

Bảng Customer lưu thông tin khách hàng có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| phone | NVARCHAR(12) | PK,NOT NULL | Số điện thoại khách hàng |
| name | NVARCHAR(50) | NOT NULL | Tên khách hàng |
| point | INT | NOT NULL | Điểm |

* + - 1. ***Drinks***

Bảng Drinks lưu thông tin các loại nước uống có cấu trúc như sau

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| id | INT | PK, NOT NULL | ID của loại nước |
| name | NVARCHAR(100) | NOT NULL | Tên từng loại nước |
| idcategory | INT | FK,NOT NULL |  |
| price | FLOAT | NOT NULL | Giá |

* + - 1. ***OrderNumber***

Bảng Import :

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| Id | INT | PK,NOT NULL | Id |
| status | BIT | NOT NULL | Trạng thái |

* + - 1. ***Storage***

Bảng Storage :

|  |  |  |  |
| --- | --- | --- | --- |
| **Tên cột** | **Kiểu** | **Ràng buộc** | **Ghi chú** |
| iddrink | INT | FK, NULL |  |
| quantity | INT | NULL |  |

### THIẾT KẾ GIAO DIỆN

## Sơ đồ tổ chức giao diện

Giao diện chính là một cửa sổ chứa menu chính và toolbar. Thông qua đó để đi đến các giao diện thành viên để thực hiện các chức năng trong hệ thống.

Theo yêu cầu thì mọi nhân viên phải đăng nhập trước khi sử dụng ứng dụng nên form đăng nhập xuất hiện trước để yêu cầu đăng nhập.

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

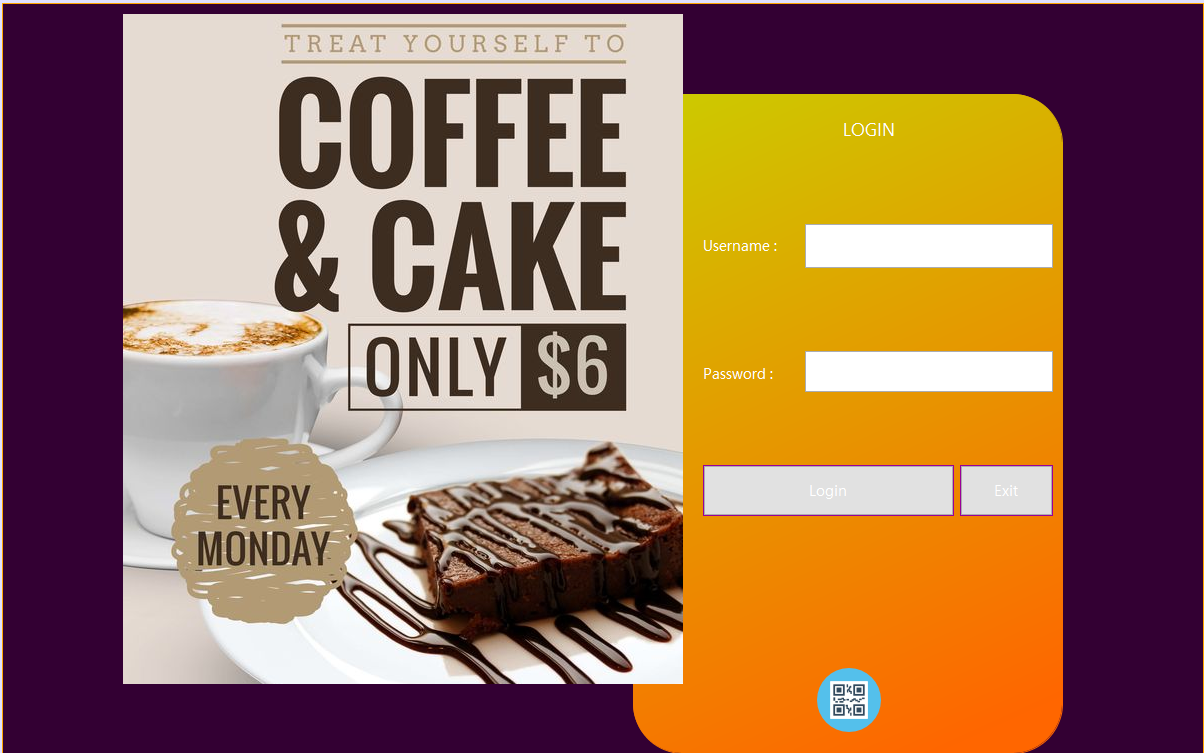
Trong phần thiết kế giao diện các bạn nên sử dụng gói hình ảnh được cung cấp sẵn trên tài nguyên học tập của LMS. Nếu các bạn không muốn sử dụng gói hình ảnh này thì bạn có thể chọn gói hình ảnh phù hợp của các bạn.

Lưu ý quan trọng:

* + - * Các bạn phải đặt tên đúng vì việc đặt tên sẽ ảnh hưởng tới việc viết mã sau này
      * Nếu các bạn đặt tên khác thì phải chỉnh mã sau này theo tên mà các bạn đã đặt.
      * Trong lập trình giao diện người ta thường qui ước tiếp đầu ngữ của tên các thành phần dao diện như sau:

|  |  |  |
| --- | --- | --- |
| **Control** | **Prefix** | **Mô tả** |
| JLabel | lbl | Nhãn |
| JButton | btn | Nút |
| JTextField, JTextArea, JPassword, JTextPane | txt | Các loại ô nhập |
| JRadioButton | rdo | Hộp chọn |

|  |  |  |
| --- | --- | --- |
| JcheckBox | chk | Hộp kiểm |
| ButtonGroup | bgr | Nhóm các button |
| JcomboBox | cbo | Hộp chọn xổ xuống |
| JlistBox | lst | Hộp chọn nhiều |
| Jtree | tree | Cây |
| Jmenu | mnu | Thực đơn |
| JmenuItem | mni | Mục menu |
| Jpanel | pnl | Khung chứa |
| JtabbedPane | tab |  |
| Jtable | tbl |  |
| … |  |  |

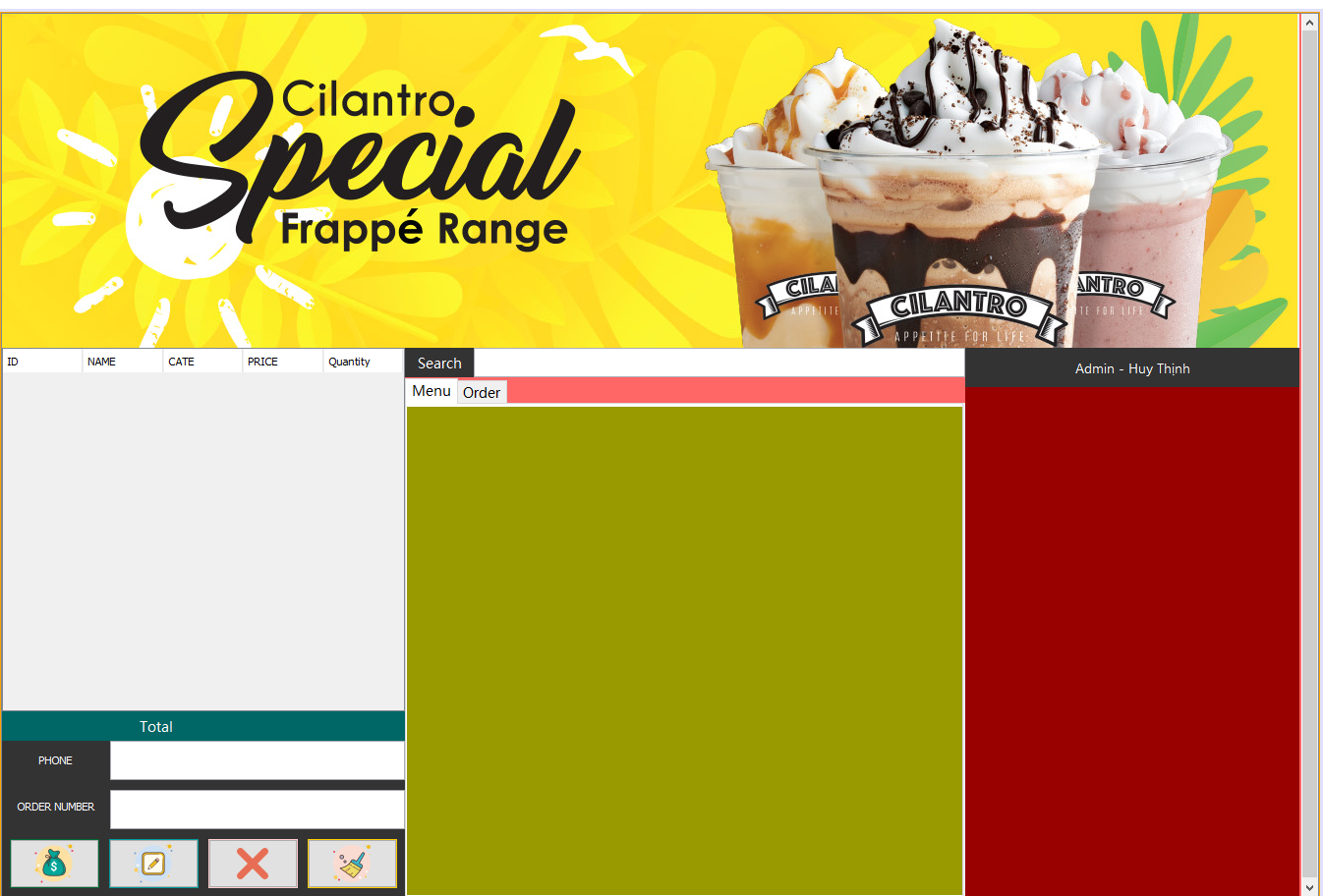
* + - 1. ***Login***

Các thành phần giao diện:

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | Jpanel1 | JLabel | Background:[51,0,51]  Border: No Border  Foreground :[0,0,0] |
| 2 | JLabel2 | JLable | Icon:a1ce2ecbf64622ccf492fe19548b7d4d.jpg  Background: [240,240,240] |

3

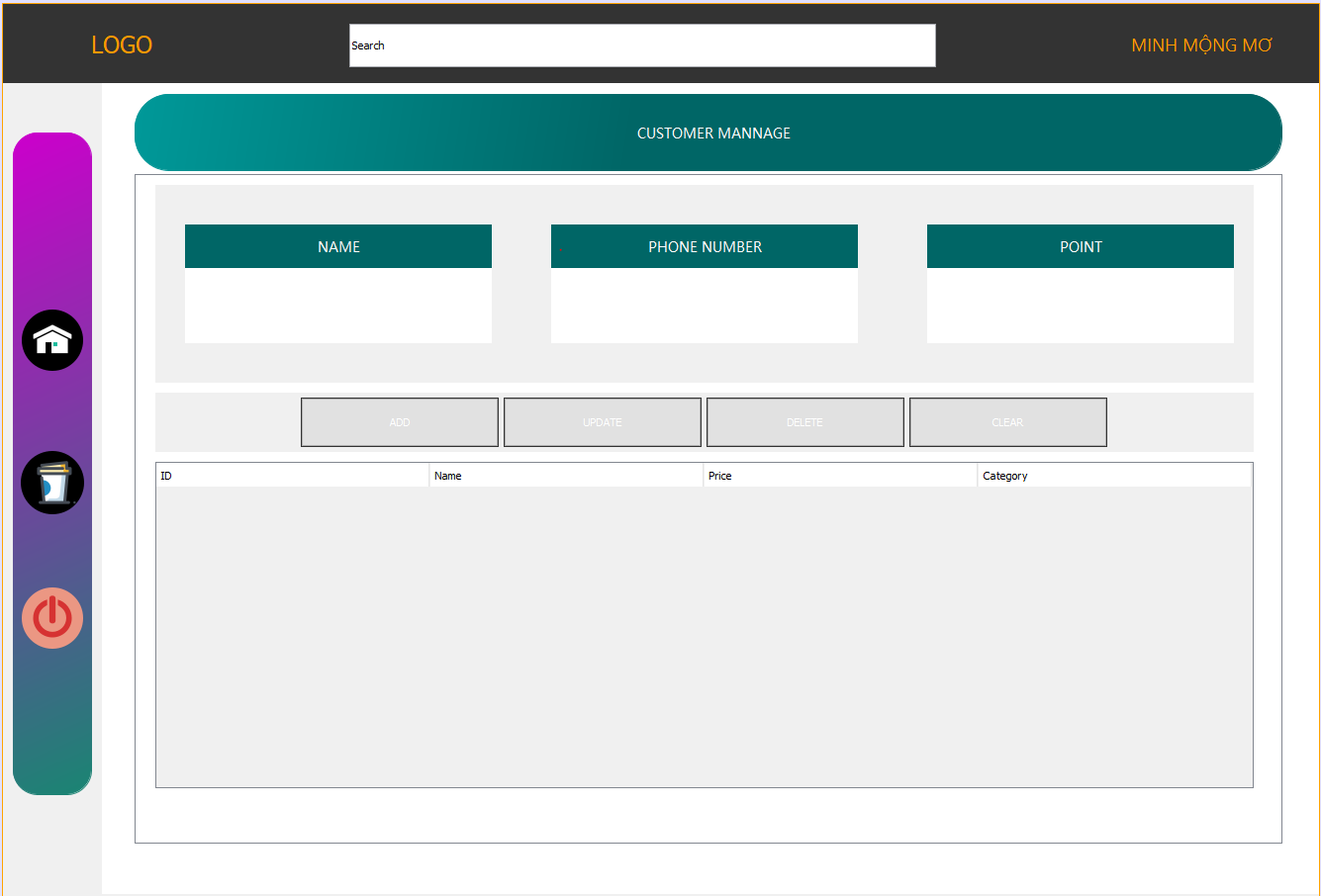
|  |  |  |  |
| --- | --- | --- | --- |
| 3 | Jlabel1 | JLable | Font : Segoe Ui 18 Plain  Foreground : [255,255,255]  Text : LOGIN  VerticalAlignment : CENTER |
| 4 | Jlabel3 | JLable | Font : Microsoft YaHei 14 Plain  Foreground : [255,255,255]  Text : Username :  VerticalAlignment : CENTER |
| 5 | Jlabel4 | JLable | Font : Microsoft YaHei 14 Plain  Foreground : [255,255,255]  Text : Password:  VerticalAlignment : CENTER |
| 6 | Jlabel5 | JLable | Font : Tahoma 11 Plain  Foreground : [0,0,0]  VerticalAlignment : CENTER |
| 7 | jButton1 | JButon | Backgound : [153,0,153]  Foreground : [255,255,255]  Text : Login |
| 8 | jButton2 | JButon | Backgound : [153,0,153]  Foreground : [255,255,255]  Text : Exit |
| 9 | kGradientPanel1 | kGradientPanel | Backgound : [51,0,51]  Foreground : [255,255,255]  Border: No Border |

* + - 1. ***Menu***

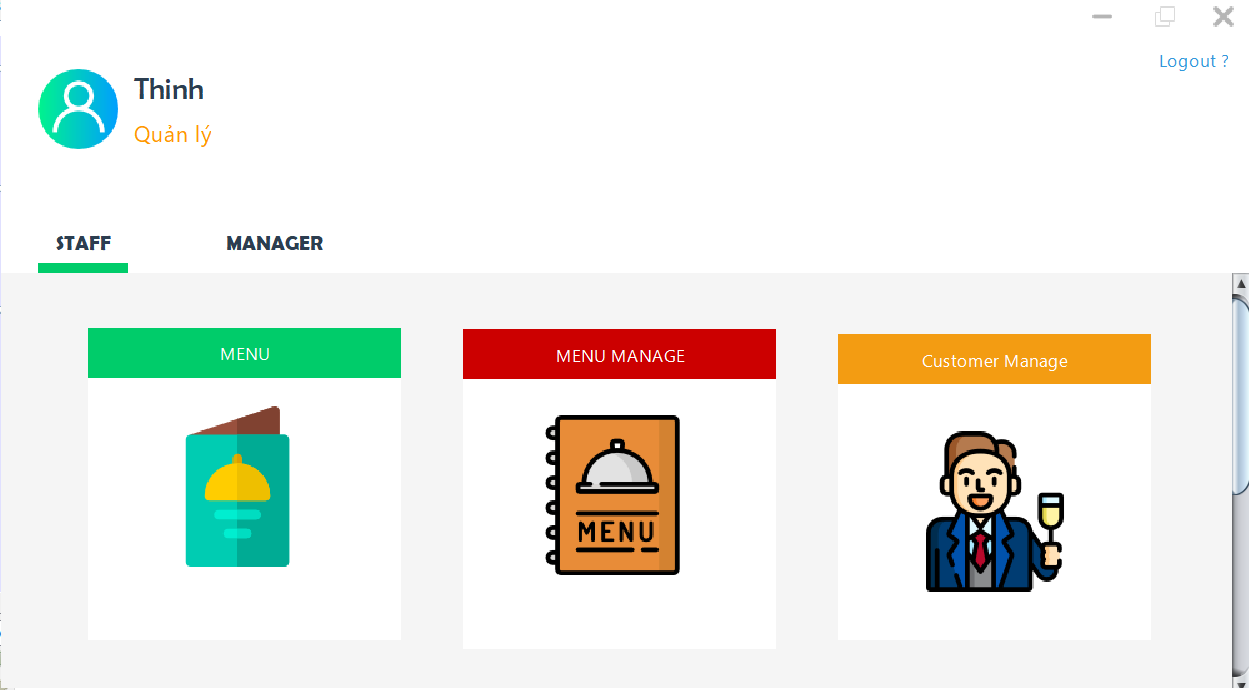
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | pnlCate | JPanel | Background: [153,0,0]  Border: No Border  Foreground :[0,0,0] |
| 2 | pnlMenu | JPanel | Background: [153,0,0]  Border: No Border  Foreground :[0,0,0] |
| 3 | tblBill | JTable | Background:[255,255,255]  Border: No Border  Font: Tahoma 11 Plain  Foreground :[0,0,0] |
| 4 | tabOrder | JTabbedPane | Background:[240,240,240]  Border: No Border  Foreground :[0,0,0]  tabLayoutPolicy: WRAP\_TAB\_LAYOUT  tabPlacement : TOP |
| 5 | lblname | JLable | Font : Segoe Ui 18 Plain  Foreground : [255,255,255]  Text : Admin - Huy Thịnh  Background: [240,240,240] |
| 6 | Jlable4 | JLable | Font : Micrisoft Yahei 14 Plain  Foreground : [255,255,255]  Text : Search  Background: [240,240,240] |
| 7 | Jlable3 | JLable | Font : Tahoma 11 Plain  Foreground : [0,0,0]  Icon: Website-1.jpg  Background: [255,204,0] |

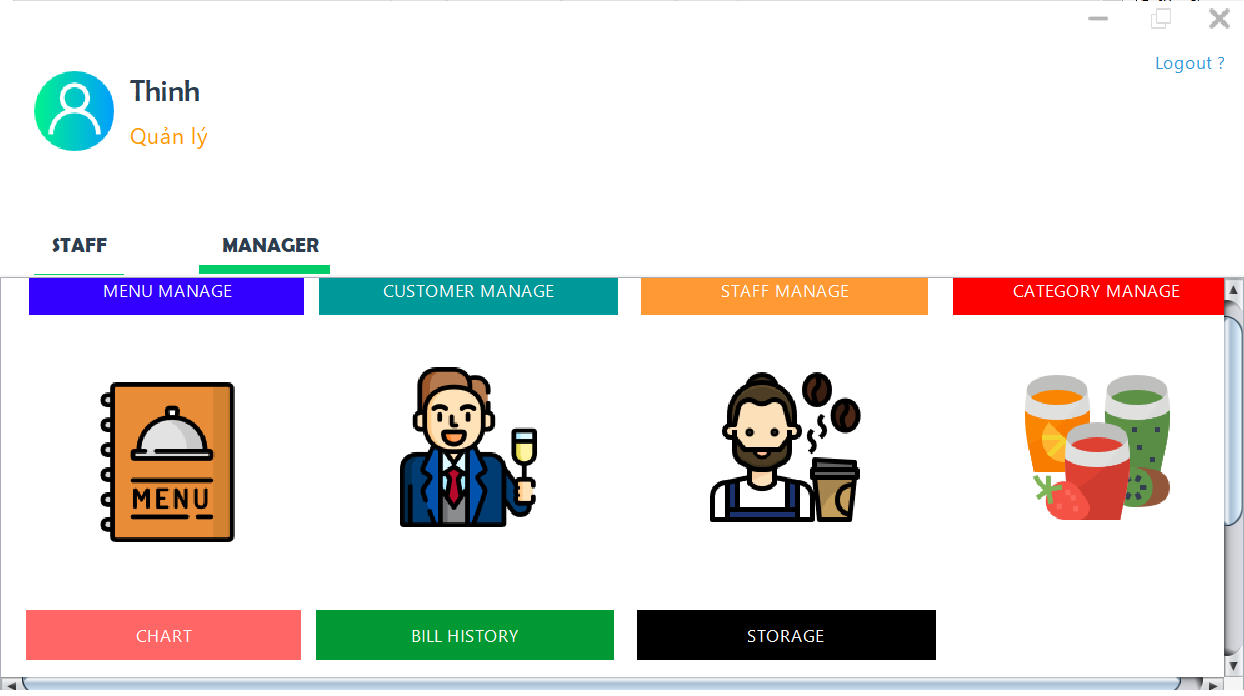
3

|  |  |  |  |
| --- | --- | --- | --- |
| 8 | lbl22222 | JLable | Font : Microsoft YaHei 14 Plain  Foreground : [255,255,255]  Text : Total  Background: [240,240,240] |
| 9 | txtOrderNumber | JTextField | Font : Tahoma 11 Plain  Foreground : [0,0,0]  Background: [255,255,255] |
| 10 | txtCustomerPhone | JTextField | Font : Tahoma 11 Plain  Foreground : [0,0,0]  Background: [255,255,255] |
| 11 | Jlabel5 | JLable | Font : Tahoma 11 Plain  Foreground : [255,255,255]  Background: [240,240,240] |
| 12 | Jlabel6 | JLable | Font : Tahoma 11 Plain  Foreground : [255,255,255]  Background: [240,240,240] |
| 13 | btnCharge | JButon | Font : Microsoft YaHei 14 Plain  Foreground : [0,0,0]  Icon : icons8-money-bag-50.pngBackground: [0,102,51] |
| 14 | jButton2 | JButon | Font : Microsoft YaHei 14 Plain  Foreground : [0,0,0]  Icon: icons8-edit-50.pngBackground: [0,153,153] |
| 15 | btnDelete | JButon | Font : Microsoft YaHei 14 Plain  Foreground : [0,0,0]  Icon: clear.pngBackground: [255,204,204] |
| 16 | btnClear | JButon | Font : Microsoft YaHei 14 Plain  Foreground : [0,0,0]  Icon: icons8-broom-50.pngBackground: [255,204,0] |

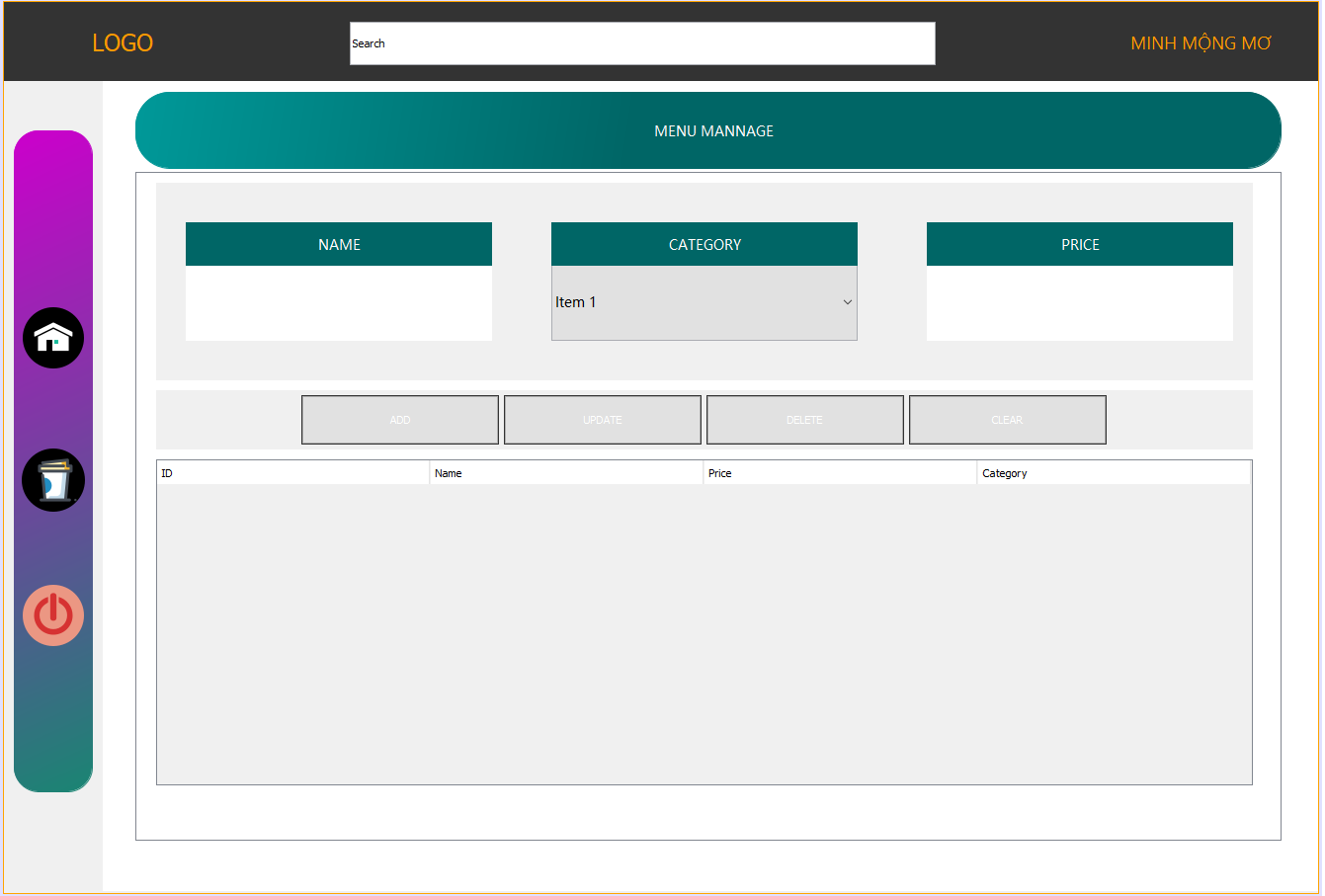
* + - 1. ***CustomerManager***

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | kGradientPane3 | kGradientPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 2 | kGradientPanel4 | kGradientPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 3 | Jpanel1 -> Jpanel6 | JPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 4 | Jlable1 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: NAME  Font: Microsoft Yahei 14 Plain |
| 5 | lblname | JLable | Backgound : [255,153,0]  Foreground : [255,153,0]  Text: MINH MỘNG MƠ  Font: Segoe UI 18 Plain |
| 6 | Jlable2 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: phone  Font: Microsoft Yahei 14 Plain |
| 7 | Jlable3 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: POINT  Font: Microsoft Yahei 14 Plain |
| 8 | Jlable4 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-takeaway-hot-drink-64.png  Font: Tamoha 11 Plain |
| 9 | Jlable5 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-shutdown-64.png  Font: Tamoha 11 Plain |
| 10 | Jlable6 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-home-64.png  Font: Tamoha 11 Plain |
| 11 | Jlable7 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: CUSTOMER MANNAGE  Font: Microsoft Yahei 14 Plain |
| 12 | Jlable9 | JLable | Backgound : [255,255,255]  Foreground : [255,153,0]  Icon : Asset 7.png  Font: Microsoft Yahei 24 Plain |
| 13 | btnAdd | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: ADD  Font: Tamoha 11 Plain |
| 14 | btnUpdate | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: UPDATE  Font: Tamoha 11 Plain |
| 15 | btnDelete | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: DELETE  Font: Tamoha 11 Plain |
| 16 | btnClear | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: CLEAR  Font: Tamoha 11 Plain |
| 17 | tblCustomer | JTable | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tamoha 11 Plain  No Border |
| 18 | txtPhoneNumber | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
| 19 | txtName | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
| 20 | txtPoint | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
| 21 | JtextField1 | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Text: Search  Font: Tamoha 11 Plain |

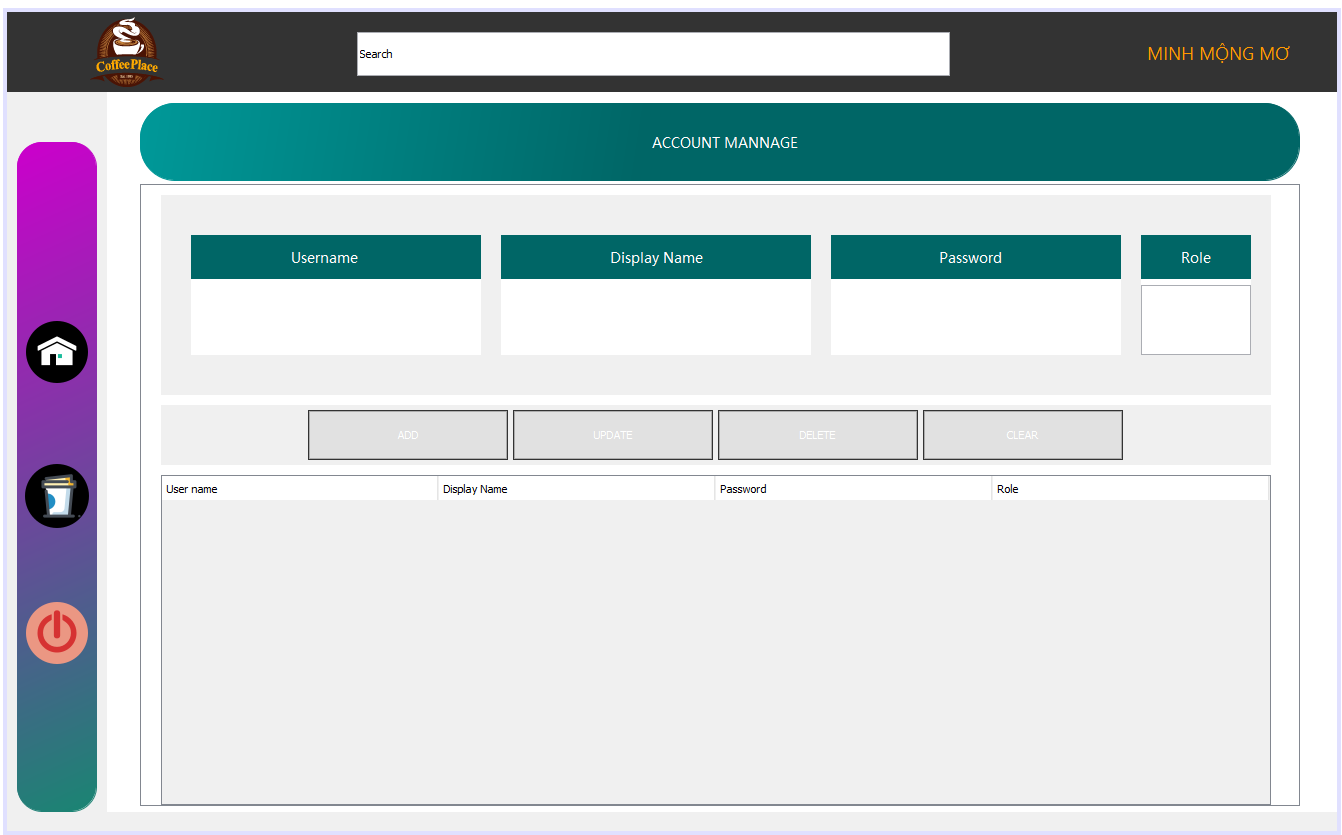
* + - 1. ***Dashboard***



|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | pnlHeader | JPanel | Backgound : [255,255,255]  Foreground : [255,255,255]  Border: No Border |
| 2 | btnMinimize | JButton | Backgound : [255,255,255]  Foreground : [0,0,0]  Icon: Minimize.png |
| 3 | btnMaximize | JButton | Backgound : [255,255,255]  Foreground : [0,0,0]  Icon: Maximize.png |
| 4 | btnExit | JButton | Backgound : [255,255,255]  Foreground : [0,0,0]  Icon: Exit.png |
| 5 | PnlMenu | JPanel | Backgound : [255,255,255]  Foreground : [0,0,0]  Border: No Border |
| 6 | Jlabel17 | JLabel | Backgound : [240,240,240]  Foreground : [52,152,219]  Font: SegoeUI 14 Plain  Text: Logout? |
| 7 | lblAccountName | JLabel | Backgound : [240,240,240]  Foreground : [44,62,80]  Font: Berlin Sans FB Demi 20 Bold  Text: Thinh |
| 8 | Jlabel1 | JLabel | Backgound : [240,240,240]  Foreground : [0,0,0]  Font: Tahoma 11 Plain  Icon: user (1).png |
| 9 | pnlTimeline | JPanel | Backgound : [245,245,245]  Foreground : [0,0,0] |
| 10 | jpsTimeline | JScrollPanel | Backgound : [240,240,240]  Foreground : [0,0,0] |

* + - 1. ***MenuManager***

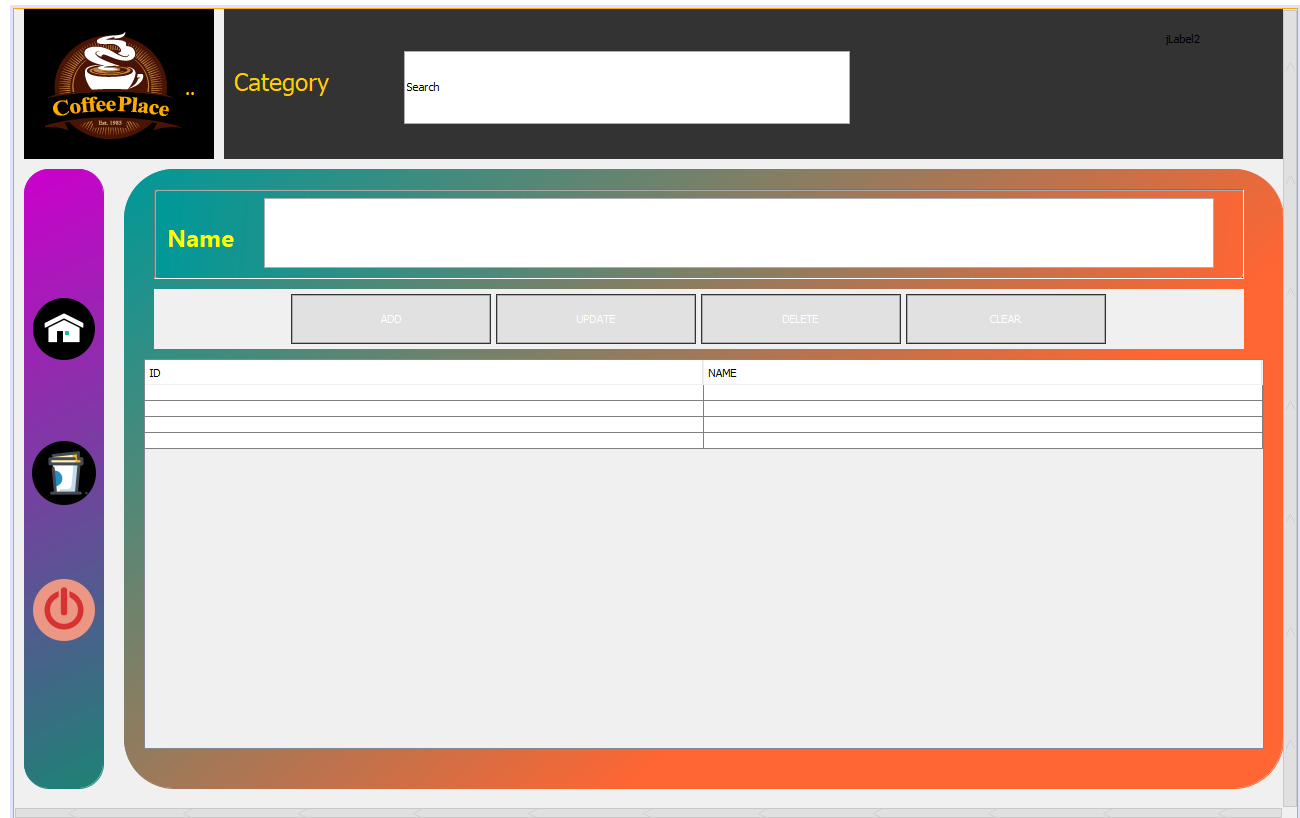
|  |  |  |
| --- | --- | --- |
| **Thành phần** | **Kiểu** | **Thuộc tính** |
| kGradientPane3 | kGradientPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| kGradientPanel4 | kGradientPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| Jpanel1 -> Jpanel6 | JPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| Jlable1 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: CATEGORY  Font: Microsoft Yahei 14 Plain |
| lblname | JLable | Backgound : [255,153,0]  Foreground : [255,153,0]  Text: MINH MỘNG MƠ  Font: Segoe UI 18 Plain |
| Jlable2 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: NAME  Font: Microsoft Yahei 14 Plain |
| Jlable3 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: PRICE  Font: Microsoft Yahei 14 Plain |
| Jlable4 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-takeaway-hot-drink-64.png  Font: Tamoha 11 Plain |
| Jlable5 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-shutdown-64.png  Font: Tamoha 11 Plain |
| Jlable6 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-home-64.png  Font: Tamoha 11 Plain |
| Jlable7 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: MENU MANAGER  Font: Microsoft Yahei 14 Plain |
| Jlable9 | JLable | Backgound : [255,255,255]  Foreground : [255,153,0]  Icon : Asset 7.png  Font: Microsoft Yahei 24 Plain |
| btnAdd | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: ADD  Font: Tamoha 11 Plain |
| btnUpdate | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: UPDATE  Font: Tamoha 11 Plain |
| btnDelete | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: DELETE  Font: Tamoha 11 Plain |
| btnClear | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: CLEAR  Font: Tamoha 11 Plain |
| tblManage | JTable | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tamoha 11 Plain  No Border |
| txtPhoneNumber | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
| cbCategory | JComboBox | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Microsoft Yahei 14 Plain |
| txtPoint | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
| JtextField1 | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Text: Search  Font: Tamoha 11 Plain |

* + - 1. ***AccountManege***

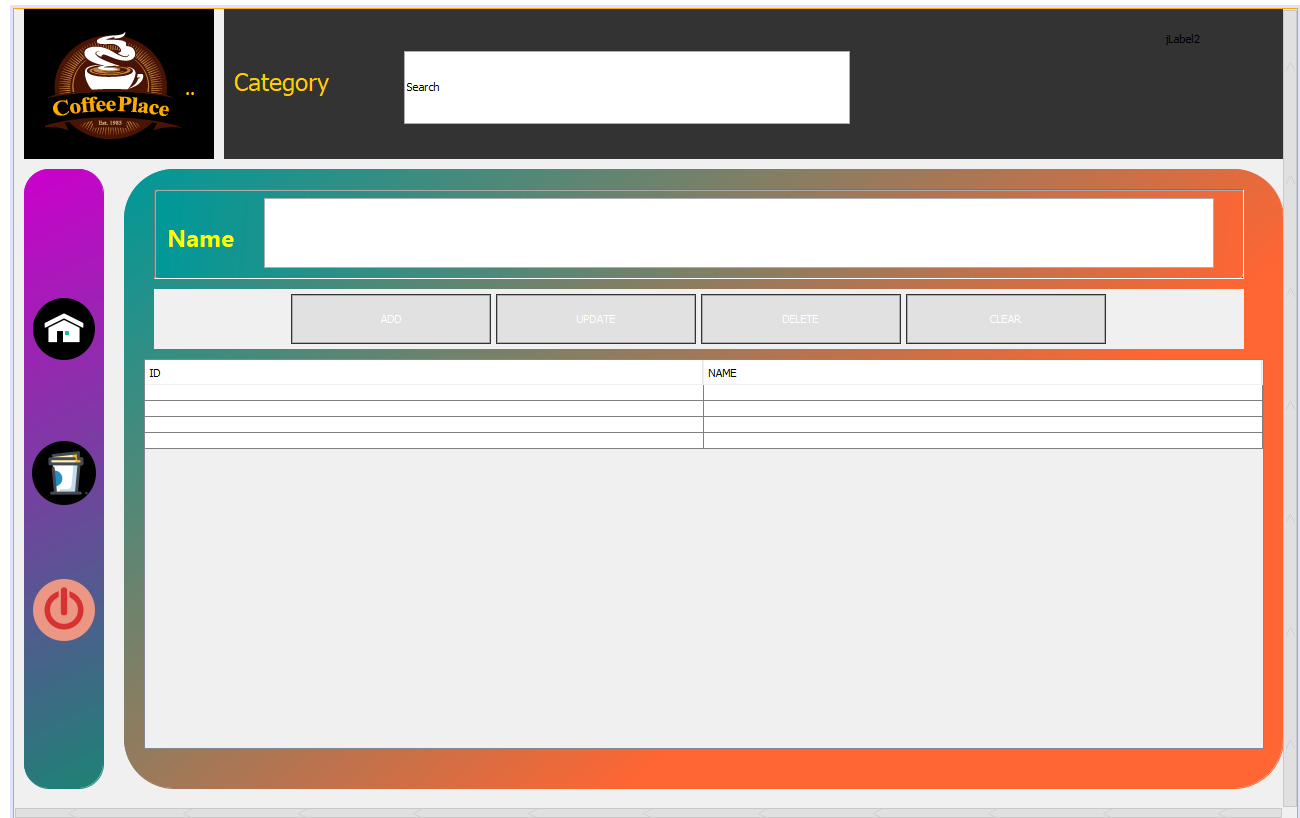
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | kGradientPane3 | kGradientPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 2 | kGradientPanel4 | kGradientPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 3 | Jpanel1 -> Jpanel6 | JPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 4 | Jlable1 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: Display Name  Font: Microsoft Yahei 14 Plain |
| 5 | lblname | JLable | Backgound : [255,153,0]  Foreground : [255,153,0]  Text: MINH MỘNG MƠ  Font: Segoe UI 18 Plain |
| 6 | Jlable2 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: Username  Font: Microsoft Yahei 14 Plain |
| 7 | Jlable3 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: Password  Font: Microsoft Yahei 14 Plain |
| 8 | Jlable4 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-takeaway-hot-drink-64.png  Font: Tamoha 11 Plain |
| 9 | Jlable5 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-shutdown-64.png  Font: Tamoha 11 Plain |
| 10 | Jlable6 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-home-64.png  Font: Tamoha 11 Plain |
| 11 | Jlable7 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: CUSTOMER MANNAGE  Font: Microsoft Yahei 14 Plain |
| 12 | Jlable9 | JLable | Backgound : [240,240,240]  Foreground : [255,255,255]  Icon : Asset 7.png  Font: Microsoft Yahei 24 Plain |
|  | Jlable10 | JLable | Backgound : [255,255,255]  Foreground : [255,255,255]  Text: Role  Font: Microsoft Yahei 24 Plain |
|  | btnAdd | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: ADD  Font: Tamoha 11 Plain |
|  | btnUpdate | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: UPDATE  Font: Tamoha 11 Plain |
|  | btnDelete | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: DELETE  Font: Tamoha 11 Plain |
|  | btnClear | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: CLEAR  Font: Tamoha 11 Plain |
|  | tblAccount | JTable | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tamoha 11 Plain  No Border |
|  | txtUsername | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
|  | txtDisplayName | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
|  | txtPassword | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
|  | txtRole | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Segoe UI 14 Plain |
|  | JtextField1 | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Text: Search  Font: Tamoha 11 Plain |

* + - 1. ***Thống kê***

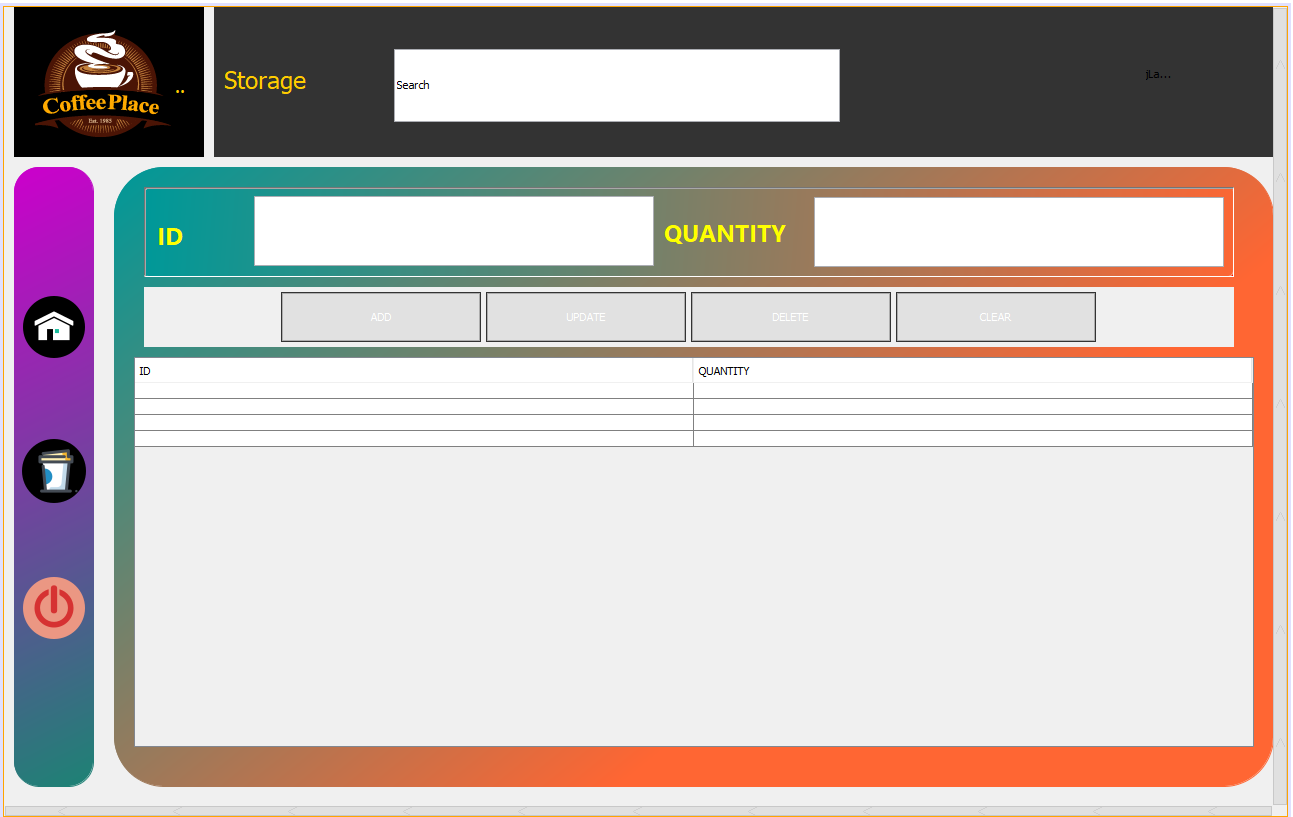
|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | Jpanel1 | JPanel | Backgound : [51,51,51]  Foreground : [0,0,0]  Border: No Border |
| 2 | Jlabel1 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Font: Tahoma 11 Plain  Text: Thống Kê |
| 3 | Jlabel1 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: Asset 6.png  Font: Tahoma 11 Plain |
| 4 | Jbutton1 | JButton | Backgound : [0,153,153]  Foreground : [255,255,0]  Text: Chọn  Font: Segoe UI 24 Plain |
| 5 | Jbutton2 | JButton | Backgound : [0,153,153]  Foreground : [255,255,0]  Text: Thoát  Font: Segoe UI 24 Plain |
| 6 | cb | JComboBox | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tahoma 18 Plain |

* + - 1. ***BillHistory***

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | kGradientPane3 | kGradientPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: No Border |
| 2 | kGradientPanel1 | kGradientPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: No Border |
| 3 | Jpanel1 | JPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: No Border |
| 4 | Jpanel3 | JPanel | Backgound : [51,51,51]  Foreground : [0,0,0]  Border: No Border |
| 5 | Date\_min | JDateChooser | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border  Date: Dec 22, 2019 11:33:40 PM |
| 6 | Date\_max | JDateChooser | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border  Date: Dec 22, 2019 11:33:40 PM |
| 7 | Jlable2 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Text: jLabel2  Icon:Asset 6.png  Font: Tahoma 11 Plain |
| 8 | Jlable4 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-takeaway-hot-drink-64.png  Font: Tamoha 11 Plain |
| 9 | Jlable5 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-shutdown-64.png  Font: Tamoha 11 Plain |
| 10 | Jlable6 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-home-64.png  Font: Tamoha 11 Plain |
| 11 | Jlable7 | JLable | Backgound : [255,51,102]  Foreground : [255,153,0]  Text: Bill History  Font: Segoe UI 21 Plain |
| 12 | tblBill | JTable | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tahoma 11 Plain |
| 13 | JtextField3 | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tahoma 11 Plain  Text:Search |

* + - 1. ***CategoryFRM***

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | kGradientPane3 | kGradientPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: No Border |
| 2 | kGradientPanel2 | kGradientPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: Soft Bevel Border |
| 3 | kGradientPanel1 | kGradientPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 4 | Jpanel1 | JPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: No Border |
| 5 | Jpanel3 | JPanel | Backgound : [51,51,51]  Foreground : [0,0,0]  Border: No Border |
| 6 | Jpanel6 | JPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 7 | Date\_min | JDateChooser | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border  Date: Dec 22, 2019 11:33:40 PM |
| 8 | Date\_max | JDateChooser | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border  Date: Dec 22, 2019 11:33:40 PM |
| 9 | Jlable1 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Text: jLabel1  Icon:Asset 6.png  Font: Tahoma 11 Plain |
| 10 | Jlable3 | JLable | Backgound : [255,204,0]  Foreground : [255,255,0]  Text:NAME  Font: Segoe UI 24 Bold |
| 11 | txtName | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tamoha 11 Plain |
| 12 | Jlable4 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-takeaway-hot-drink-64.png  Font: Tamoha 11 Plain |
| 13 | Jlable5 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-shutdown-64.png  Font: Tamoha 11 Plain |
| 14 | Jlable6 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-home-64.png  Font: Tamoha 11 Plain |
| 15 | Jlable7 | JLable | Backgound : [255,51,102]  Foreground : [255,153,0]  Text: Bill History  Font: Segoe UI 21 Plain |
|  | tblCategory | JTable | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tahoma 11 Plain |
|  | JtextField3 | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tahoma 11 Plain  Text:Search |

* + - 1. ***StorageFRM***

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Thành phần** | **Kiểu** | **Thuộc tính** |
| 1 | kGradientPane3 | kGradientPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: No Border |
| 2 | kGradientPanel2 | kGradientPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: Soft Bevel Border |
| 3 | kGradientPanel1 | kGradientPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 4 | Jpanel1 | JPanel | Backgound : [102,102,102]  Foreground : [0,0,0]  Border: No Border |
| 5 | Jpanel3 | JPanel | Backgound : [51,51,51]  Foreground : [0,0,0]  Border: No Border |
| 6 | Jpanel6 | JPanel | Backgound : [240,240,240]  Foreground : [0,0,0]  Border: No Border |
| 7 | btnAdd | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: ADD  Font: Tamoha 11 Plain |
| 8 | btnUpdate | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: UPDATE  Font: Tamoha 11 Plain |
| 9 | btnDelete | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: DELETE  Font: Tamoha 11 Plain |
| 10 | btnClear | JButton | Backgound : [51,51,51]  Foreground : [255,255,255]  Text: CLEAR  Font: Tamoha 11 Plain |
| 11 | Jlable1 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Text: jLabel1  Icon:Asset 6.png  Font: Tahoma 11 Plain |
| 12 | Jlable3 | JLable | Backgound : [255,204,0]  Foreground : [255,255,0]  Text:ID  Font: Segoe UI 24 Bold |
| 13 | Jlable2 | JLable | Backgound : [255,204,0]  Foreground : [255,255,0]  Text:QUANTITY  Font: Segoe UI 24 Bold |
| 14 | txtName | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tamoha 11 Plain |
| 15 | Jlable4 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-takeaway-hot-drink-64.png  Font: Tamoha 11 Plain |
| 16 | Jlable5 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-shutdown-64.png  Font: Tamoha 11 Plain |
| 17 | Jlable6 | JLable | Backgound : [240,240,240]  Foreground : [0,0,0]  Icon: icons8-home-64.png  Font: Tamoha 11 Plain |
| 18 | Jlable7 | JLable | Backgound : [255,51,102]  Foreground : [255,153,0]  Text: Bill History  Font: Segoe UI 21 Plain |
| 19 | tblStorage | JTable | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tahoma 11 Plain |
| 20 | JtextField3 | JTextField | Backgound : [255,255,255]  Foreground : [0,0,0]  Font: Tahoma 11 Plain  Text:Search |

* + - 1. ***Hướng dẫn sử dụng***

Thiết kế giao diện:

Thiết kế các trang web hướng dẫn sử dụng phần mềm. Trong phần viết mã sẽ hướng dẫn cách mở các trang web này khi người dùng nhấp vào mục hướng dẫn trong menu trợ giúp.

# THỰC HIỆN VIẾT MÃ

* 1. **VIẾT MÃ TẠO CSDL**

## Tạo CSDL

1. create database Cafe\_DuAn1
2. go
3. use Cafe\_DuAn1
4. go
5. --Table
6. create table OrderNumber
7. (
8. id int identity primary key,
9. [status] bit not null default 0 -- trống || có người
10. )
11. go
12. --Account
13. create table Account
14. (
15. Username nvarchar(100) primary key,
16. DisplayName nvarchar(100)NOT NULL default N'Nhân Viên',
17. [PassWord] nvarchar(1000)NOT NULL default 0,
18. [Type] bit NOT NULL default 0--1:chủ && 0:NV
19. )
20. go
21. --Category
22. create table Category
23. (
24. id int identity primary key,
25. name nvarchar(100)NOT NULL default N'chưa đặt tên',
26. )
27. go
28. --Drinks&somethingelse
29. create table Drinks
30. (
31. id int identity primary key,
32. name nvarchar(100)NOT NULL default N'Chưa đặt tên',
33. idcategory int NOT NULL foreign key references Category(id),
34. price float NOT NULL default 0,
35. images nvarchar(100) not null default N'Phong đẹp trai.jpg'
36. )
37. go
38. --Customer
39. create table Customer
40. (
41. phone nvarchar(12) primary key,
42. name nvarchar(50) not null,
43. point int default 0
44. )
45. go
46. --Bill
47. create table Bill
48. (
49. id int identity primary key,
50. DateCheckIn date NOT NULL default getdate(),
51. OrderNumber int foreign key references OrderNumber(id),
52. Staffname nvarchar(50) not null,
53. CustomerPhoneNumber nvarchar(12) foreign key references Customer(phone),
54. TotalPrice float default 0
55. )
56. go
57. --BillInfo
58. create table BillInfo
59. (
60. idBill int NOT NULL foreign key references Bill(id),
61. idDrinks int NOT NULL foreign key references Drinks(id),
62. Quantity int NOT NULL default 0,
63. Price float not null,
64. constraint pk\_billinfo primary key (idBill,idDrinks)
65. )
66. go
67. --Storage
68. create table Storage
69. (
70. iddrink int foreign key references Drinks(id),
71. quantity int
72. )
73. go
74. --Supplier
75. create table supplier
76. (
77. id nvarchar(100) primary key,
78. name nvarchar(100) not null
79. )
80. go
81. --Import
82. create table Import
83. (
84. importDate date default getdate(),
85. iddrinks int foreign key references Drinks(id),
86. idsupplier nvarchar(100) foreign key references supplier(id),
87. paymoney float,
88. quantity int
89. )
90. go

## SQL truy vấn và thao tác bảng

* + - 1. ***Account***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO Account (Username,DisplayName,PassWord,Type) VALUES (?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE Account SET PassWord=?, DisplayName=?, VaiTro=? WHERE Username=? |
| Xóa theo mã | DELETE FROM NhanVien WHERE Username=? |
| Truy vấn tất cả | SELECT \* FROM ACCOUNT |
| Truy vấn theo mã | where Username like |

* + - 1. ***Bill***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | insert into Bill(OrderNumber,Staffname,CustomerPhoneNumber,TotalPrice) values (?,?,?,?) |
| Cập nhật theo mã | UPDATE Bill SET Staffname=?,CustomerPhoneNumber=?, TotalPrice=? WHERE id=? |
| Xóa theo mã | DELETE FROM Bill WHERE id=? |
| Truy vấn tất cả | select max(id) as id from Bill |
| Truy vấn theo mã | select \* from Bill where id like ? |

* + - 1. ***BillInfo***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO BillInfo (idBill,idDrinks,Quantity,Price) VALUES (?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE BillInfo SET idBill=?, idDrinks=?, CountItem=? WHERE id=? |

|  |  |
| --- | --- |
| Xóa theo mã | DELETE FROM BillInfo WHERE id=? |
| Truy vấn tất cả | select \* from BillInfo |
| Truy vấn theo mã | select \* from BillInfo where id like=? |

* + - 1. ***KhoaHoc***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | Insert dbo.Category values (?) |
| Cập nhật theo mã | Update dbo.Category set name = ? where id = ? |
| Xóa theo mã | Delete from dbo.Category where id = ? |
| Truy vấn tất cả | select \* from Category |
| Truy vấn theo mã | select \* from Category where name like = ? |

* + - 1. ***Customer***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | Insert into dbo.Customer(phone,name,point) values (?,?,?) |
| Cập nhật theo mã | Delete from dbo.Customer where phone = ? |
| Xóa theo mã | Update dbo.Customer set name = ? , point = ? where phone = ? |
| Truy vấn tất cả | select \* from Customer |
| Truy vấn theo mã | select \* from Customer where Phone like =? |

* + - 1. ***Drinks***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | insert into drinks(name,idcategory,price) values (?,?,?) |
| Cập nhật theo mã | UPDATE Drinks SET name=?, idcategory=?, price=? WHERE id=? |
| Xóa theo mã | DELETE FROM Drinks WHERE id=? |
| Truy vấn tất cả | select \* from drinks where idcategory =? |
| Truy vấn theo mã | select \* from drinks where name like =? |

* + - 1. ***Import***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO Import (importDate,iddrinks,idsupplier,paymoney,quantity) VALUES (?, ?, ?, ?, ?) |
| Cập nhật theo mã | UPDATE Bill SET DateCheckIn=?, DateCheckOut=?, idTable=?, status=?, totalprice=? WHERE id=? |
| Xóa theo mã | DELETE FROM Import WHERE id=? |
| Truy vấn tất cả | select \* from Import |
| Truy vấn theo mã | select \* from Import where id like =? |

* + - 1. ***OrderNumber***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | Insert into OrderNumber values (?,?) |
| Cập nhật theo mã | Update OrderNumber set status = ? where id = ?" |
| Xóa theo mã | Delete from OrderNumber where id = ? |
| Truy vấn tất cả | select \* from OrderNumber |
| Truy vấn theo mã | select \* from OrderNumber where id =? |

* + - 1. ***Storage***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | Insert into dbo.tablecoffe values (?,?) |
| Cập nhật theo mã | Update dbo.tablecoffe set name = ? , status = ? where id = ? |
| Xóa theo mã | Delete from dbo.tablecoffe where id = ? |
| Truy vấn tất cả | select \* from Import |
| Truy vấn theo mã | "select \* from Import where id like =? |

* + - 1. ***Supplier***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | INSERT INTO supplier (id,name) VALUES (?, ?) |
| Cập nhật theo mã | UPDATE supplier SET name=? WHERE id=? |
| Xóa theo mã | DELETE FROM supplier WHERE id=? |
| Truy vấn tất cả | select \* from supplier |
| Truy vấn theo mã | select \* from supplier where id like =?, select \* from supplier where name like =? |

* + - 1. ***TableCoffe***

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Thêm mới | Insert into dbo.tablecoffe values (?,? |
| Cập nhật theo mã | Update dbo.tablecoffe set name = ? , status = ? where id = ? |
| Xóa theo mã | Delete from dbo.tablecoffe where id = ? |
| Truy vấn tất cả | select \* from Tablecoffe |
| Truy vấn theo mã | select \* from Tablecoffe where id like =?, select \* from Tablecoffe where name like =? |

## Thêm dữ liệu

* + - 1. ***Thêm OrderNumber***

declare @i int = 1

while @i <= 100

begin

insert dbo.OrderNumber(status) values(0)

set @i=@i+1

end

go

* 1. **LẬP TRÌNH JDBC**

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



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.

* + - 1. ***Lớp tiện ích DateHelper***

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.Date;

/\*\*

\*

\* @author Linh

\*/

public class DateHelper {

static final SimpleDateFormat DATE\_FORMATER = new SimpleDateFormat("dd-MM-yyyy");

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

try {

if (pattern.length > 0) {

DATE\_FORMATER.applyPattern(pattern[0]);

}

if (date == null) {

return DateHelper.now();

}

return DATE\_FORMATER.parse(date);

} catch (ParseException ex) {

throw new RuntimeException(ex);

}

}

/\*\*

\* Chuyển đổi từ Date sang String

\*

\* @param date là Date cần chuyển đổi

\* @param pattern là định dạng thời gian

\* @return String kết quả

\*/

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

if (pattern.length > 0) {

DATE\_FORMATER.applyPattern(pattern[0]);

}

if (date == null) {

date = DateHelper.now();

}

return DATE\_FORMATER.format(date);

}

/\*\*

\* Lấy thời gian hiện tại

\*

\* @return Date kết quả

\*/

public static Date now() {

return new Date();

}

/\*\*

\* Bổ sung số ngày vào thời gian

\*

\* @param date thời gian hiện có

\* @param days số ngày cần bổ sung váo date

\* @return Date kết quả

\*/

public static Date addDays(Date date, int days) {

date.setTime(date.getTime() + days \* 24 \* 60 \* 60 \* 1000);

return date;

}

/\*\*

\* Bổ sung số ngày vào thời gian hiện hành

\*

\* @param days số ngày cần bổ sung vào thời gian hiện tại

\* @return Date kết quả

\*/

public static Date add(int days) {

Date now = DateHelper.now();

now.setTime(now.getTime() + days \* 24 \* 60 \* 60 \* 1000);

return now;

}

}

* + - 1. ***Lớp tiện ích JdbcHelper***

##### Chứa các phương thức thực hiện các câu lệnh thao tác và truy vấn được gọn nhẹ hơn.

package Helper;

/\*\*

\*

\* @author Linh

\*/

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class JdbcHelper {

private static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

private static String dburl = "jdbc:sqlserver://localhost;database=Cafe\_DuAn1";

private static String username = "sa";

private static String password = "123";

/\*

\* Nạp driver

\*/ static {

try {

Class.forName(driver);

} catch (ClassNotFoundException ex) {

throw new RuntimeException(ex);

}

}

/\*\*

\* Xây dựng PreparedStatement

\*

\* @param sql là câu lệnh SQL chứa có thể chứa tham số. Nó có thể là một lời

\* gọi thủ tục lưu

\* @param args là danh sách các giá trị được cung cấp cho các tham số trong

\* câu lệnh sql

\* @return PreparedStatement tạo được

\* @throws java.sql.SQLException lỗi sai cú pháp

\*/

public static PreparedStatement prepareStatement(String sql, Object... args) throws SQLException {

Connection connection = DriverManager.getConnection(dburl, username, password);

PreparedStatement pstmt = null;

if (sql.trim().startsWith("{")) {

pstmt = connection.prepareCall(sql);

} else {

pstmt = connection.prepareStatement(sql);

}

for (int i = 0; i < args.length; i++) {

pstmt.setObject(i + 1, args[i]);

}

return pstmt;

}

/\*\*

\* Thực hiện câu lệnh SQL thao tác (INSERT, UPDATE, DELETE) hoặc thủ tục lưu

\* thao tác dữ liệu

\*

\* @param sql là câu lệnh SQL chứa có thể chứa tham số. Nó có thể là một lời

\* gọi thủ tục lưu

\* @param 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 stmt = prepareStatement(sql, args);

try {

stmt.executeUpdate();

} finally {

stmt.getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

/\*\*

\* Thực hiện câu lệnh SQL truy vấn (SELECT) hoặc thủ tục lưu truy vấn dữ

\* liệu

\*

\* @param sql là câu lệnh SQL chứa có thể chứa tham số. Nó có thể là một lời

\* gọi thủ tục lưu

\* @param 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 stmt = prepareStatement(sql, args);

return stmt.executeQuery();

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

}

## Model class - Các lớp mô tả dữ liệu

##### Để tránh các sai sót về kiểu dữ liệu khi làm việc với CSDL, chúng ta cần phải xây dựng các lớp mô tả dữ liệu với kiểu phù hợp. Điều này giúp làm việc với CSDL chặt chẽ hơn, dễ hơn.

* + - 1. ***Account***

***package Model;***

***/\*\****

***\****

***\* @author Minh***

***\*/***

***public class Account {***

***private String Username;***

***private String DisplayName;***

***private String PassWord;***

***private boolean Role;***

***public Account() {***

***}***

***public Account(String Username, String DisplayName, String PassWord, boolean Role) {***

***this.Username = Username;***

***this.DisplayName = DisplayName;***

***this.PassWord = PassWord;***

***this.Role = Role;***

***}***

***public String getUsername() {***

***return Username;***

***}***

***public String getDisplayName() {***

***return DisplayName;***

***}***

***public String getPassWord() {***

***return PassWord;***

***}***

***public boolean isRole() {***

***return Role;***

***}***

***public void setUsername(String Username) {***

***this.Username = Username;***

***}***

***public void setDisplayName(String DisplayName) {***

***this.DisplayName = DisplayName;***

***}***

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

***this.PassWord = PassWord;***

***}***

***public void setRole(boolean Role) {***

***this.Role = Role;***

***}***

***}***

* + - 1. ***Bill***

/\*

\* 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 Model;

import java.util.Date;

/\*\*

\*

\* @author Minh

\*/

public class Bill {

private int id;

private Date DateCheckIn;

private int Ordernumber;

private String Staffname;

private String CustomerPhoneNumber;

private Float totalprice;

public Bill(int id, Date DateCheckIn, int Ordernumber, String Staffname, String CustomerPhoneNumber, Float totalprice) {

this.id = id;

this.DateCheckIn = DateCheckIn;

this.Ordernumber = Ordernumber;

this.Staffname = Staffname;

this.CustomerPhoneNumber = CustomerPhoneNumber;

this.totalprice = totalprice;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public Date getDateCheckIn() {

return DateCheckIn;

}

public void setDateCheckIn(Date DateCheckIn) {

this.DateCheckIn = DateCheckIn;

}

public int getOrdernumber() {

return Ordernumber;

}

public void setOrdernumber(int Ordernumber) {

this.Ordernumber = Ordernumber;

}

public String getStaffname() {

return Staffname;

}

public void setStaffname(String Staffname) {

this.Staffname = Staffname;

}

public String getCustomerPhoneNumber() {

return CustomerPhoneNumber;

}

public void setCustomerPhoneNumber(String CustomerPhoneNumber) {

this.CustomerPhoneNumber = CustomerPhoneNumber;

}

public Float getTotalprice() {

return totalprice;

}

public void setTotalprice(Float totalprice) {

this.totalprice = totalprice;

}

}

* + - 1. ***BillInfo***

***package Model;***

***/\*\****

***\****

***\* @author Minh***

***\*/***

***public class BillInfo {***

***private int idBill;***

***private int idDrinks;***

***private int quantity;***

***private float Price;***

***public BillInfo(int idBill, int idDrinks, int quantity, float Price) {***

***this.idBill = idBill;***

***this.idDrinks = idDrinks;***

***this.quantity = quantity;***

***this.Price = Price;***

***}***

***public int getIdBill() {***

***return idBill;***

***}***

***public void setIdBill(int idBill) {***

***this.idBill = idBill;***

***}***

***public int getIdDrinks() {***

***return idDrinks;***

***}***

***public void setIdDrinks(int idDrinks) {***

***this.idDrinks = idDrinks;***

***}***

***public int getQuantity() {***

***return quantity;***

***}***

***public void setQuantity(int quantity) {***

***this.quantity = quantity;***

***}***

***public float getPrice() {***

***return Price;***

***}***

***public void setPrice(float Price) {***

***this.Price = Price;***

***}***

***}***

* + - 1. ***Category***

package Model;

/\*\*

\*

\* @author Minh

\*/

public class Category {

private int id;

private String name;

public Category() {

}

public Category(int id, String name) {

this.id = id;

this.name = name;

}

public int getId() {

return id;

}

public String getName() {

return name;

}

public void setId(int id) {

this.id = id;

}

public void setName(String name) {

this.name = name;

}

}

* + - 1. ***Customer***

package Model;

/\*\*

\*

\* @author Minh

\*/

public class Customer {

private String phone;

private String name;

private int point;

public Customer() {

}

public Customer(String phone, String name, int point) {

this.phone = phone;

this.name = name;

this.point = point;

}

public String getPhone() {

return phone;

}

public String getName() {

return name;

}

public int getPoint() {

return point;

}

public void setPhone(String phone) {

this.phone = phone;

}

public void setName(String name) {

this.name = name;

}

public void setPoint(int point) {

this.point = point;

}

}

* + - 1. ***Drinks***

***/\****

***\* 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 Model;***

***/\*\****

***\****

***\* @author Minh***

***\*/***

***public class Drinks {***

***private int id;***

***private String name;***

***private int idcategory;***

***private float price;***

***private String images;***

***public Drinks() {***

***}***

***public Drinks(int id, String name, int idcategory, float price) {***

***this.id = id;***

***this.name = name;***

***this.idcategory = idcategory;***

***this.price = price;***

***}***

***public int getId() {***

***return id;***

***}***

***public String getName() {***

***return name;***

***}***

***public int getIdcategory() {***

***return idcategory;***

***}***

***public float getPrice() {***

***return price;***

***}***

***public void setId(int id) {***

***this.id = id;***

***}***

***public void setName(String name) {***

***this.name = name;***

***}***

***public void setIdcategory(int idcategory) {***

***this.idcategory = idcategory;***

***}***

***public void setPrice(float price) {***

***this.price = price;***

***}***

***}***

* + - 1. ***Import***

/\*

\* 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 Model;

import java.util.Date;

/\*\*

\*

\* @author Minh

\*/

public class Import {

private Date importDate;

private int iddrinks;

private String idsupplier;

private float paymoney;

private int quantity;

public Import() {

}

public Import(Date importDate, int iddrinks, String idsupplier, float paymoney, int quantity) {

this.importDate = importDate;

this.iddrinks = iddrinks;

this.idsupplier = idsupplier;

this.paymoney = paymoney;

this.quantity = quantity;

}

public Date getImportDate() {

return importDate;

}

public int getIddrinks() {

return iddrinks;

}

public String getIdsupplier() {

return idsupplier;

}

public float getPaymoney() {

return paymoney;

}

public int getQuantity() {

return quantity;

}

public void setImportDate(Date importDate) {

this.importDate = importDate;

}

public void setIddrinks(int iddrinks) {

this.iddrinks = iddrinks;

}

public void setIdsupplier(String idsupplier) {

this.idsupplier = idsupplier;

}

public void setPaymoney(float paymoney) {

this.paymoney = paymoney;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

}

* + - 1. ***OrderNumber***

package Model;

/\*\*

\*

\* @author Linh

\*/

public class OrderNumber {

private int id;

private boolean status;

public OrderNumber(int id, boolean status) {

this.id = id;

this.status = status;

}

public int getId() {

return id;

}

public void setId(int id) {

this.id = id;

}

public boolean isStatus() {

return status;

}

public void setStatus(boolean status) {

this.status = status;

}

}

* + - 1. ***Storage***

package Model;

/\*\*

\*

\* @author Minh

\*/

public class Storage {

private int id;

private int quantity;

public Storage() {

}

public Storage(int id, int quantity) {

this.id = id;

this.quantity = quantity;

}

public int getId() {

return id;

}

public int getQuantity() {

return quantity;

}

public void setId(int id) {

this.id = id;

}

public void setQuantity(int quantity) {

this.quantity = quantity;

}

}

* + - 1. ***Suppiler***

package Model;

/\*\*

\*

\* @author Minh

\*/

public class Supplier {

private String id;

private String name;

public Supplier() {

}

public Supplier(String id, String name) {

this.id = id;

this.name = name;

}

public String getId() {

return id;

}

public String getName() {

return name;

}

public void setId(String id) {

this.id = id;

}

public void setName(String name) {

this.name = name;

}

}

## 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.

Trong phạm vi ứng dụng này chúng ta cần xây dựng mỗi DAO chứa tối thiểu các hành động sau

/\*\*

* Cập nhật thực thể vào CSDL
* @param entity là thực thể chứa thông tin bản ghi cần cập nhật

\*/

public void update(Entity entity){}

/\*\*

* Xóa bản ghi khỏi CSDL
* @param id là mã của bản ghi cần xóa

\*/

public void delete(String id){}

/\*\*

* Truy vấn tất cả các các thực thể
* @return danh sách các thực thể

\*/

public List<Entity> select(){}

/\*\*

* Truy vấn thực thể theo mã
* @param id là mã của bản ghi được truy vấn
* @return thực thể chứa thông tin của bản ghi

\*/

public Entity findById(String id){}

/\*\*

* Thêm mới thực thể vào CSDL
* @param entity là thực thể chứa thông tin bản ghi mới

\*/

public void insert(Entity entity){}

Sau đây là các lớp DAO cụ thể:

* + - 1. ***AccountDAO***

package DAO;

import Helper.JdbcHelper;

import Model.Account;

import javax.swing.JOptionPane;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author Minh

\*/

public class AccountDAO {

public void insert(String Username, String DisplayName, String PassWord, boolean Type) {

try {

String sql = "INSERT INTO Account (Username,DisplayName,PassWord,Type) VALUES (?, ?, ?, ?)";

JdbcHelper.executeUpdate(sql,

Username,

PassWord,

DisplayName,

Type);

} catch (Exception e) {

System.out.println("Loi insert");

}

}

public void update(Account model) {

try {

String sql = "UPDATE Account SET PassWord=?, DisplayName=?, VaiTro=? WHERE Username=?";

JdbcHelper.executeUpdate(sql,

model.getPassWord(), model.getDisplayName(), model.isRole(), model.getUsername());

} catch (Exception e) {

System.out.println("Loi update");

}

}

public void delete(String Username) {

try {

String sql = "DELETE FROM NhanVien WHERE Username=?";

JdbcHelper.executeUpdate(sql, Username);

} catch (Exception e) {

System.out.println("Loi xoas");

}

}

public boolean IsNumber(String num) {

if (num == null) {

return false;

}

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException ex) {

return false;

}

return true;

}

public List<Account> ShowAccounts(String username) {

try {

String sql = "select \* from Account";

if(username.length()>0){

sql+= " where Username like ?";

}

ResultSet rs = JdbcHelper.executeQuery(sql, username);

List<Account> list = new ArrayList<Account>();

while(rs.next()){

String name = rs.getString(1);

String display = rs.getString(2);

String password = rs.getString(3);

boolean type = rs.getBoolean(4);

Account ac = new Account(name, display, password, type);

list.add(ac);

}

return list;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

}

* + - 1. ***BillDAO***

package DAO;

import Helper.JdbcHelper;

import Model.Bill;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

/\*\*

\*

\* @author Minh

\*/

public class BillDAO {

public void insert(int ordernumber, String staffname, String customerphone, float totalprice) {

try {

String sql = "insert into Bill(OrderNumber,Staffname,CustomerPhoneNumber,TotalPrice) values (?,?,?,?)";

JdbcHelper.executeUpdate(sql,

ordernumber,

staffname,

customerphone,

totalprice);

} catch (Exception e) {

e.printStackTrace();

}

}

public void update(Bill model) {

try {

String sql = "UPDATE Bill SET Staffname=?,CustomerPhoneNumber=?, TotalPrice=? WHERE id=?";

JdbcHelper.executeUpdate(sql,

model.getStaffname(),

model.getCustomerPhoneNumber(),

model.getTotalprice(),

model.getId());

} catch (Exception e) {

System.out.println("Loi update");

}

}

public void delete(String id) {

try {

String sql = "DELETE FROM Bill WHERE id=?";

JdbcHelper.executeUpdate(sql, id);

} catch (Exception e) {

System.out.println("Loi xoas");

}

}

public boolean IsNumber(String num) {

if (num == null) {

return false;

}

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException ex) {

return false;

}

return true;

}

public int maxbill(int order) {

try {

String sql = "select max(id) as id from Bill";

if(order>0)

sql+=" where OrderNumber="+order;

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Integer> list = new ArrayList<>();

while(rs.next()){

int a = rs.getInt("id");

list.add(a);

}

int a= list.get(0).intValue();

return a;

} catch (Exception e) {

return 0;

}

}

public List<Bill> findbyid(String bi) {

try {

String sql = "select \* from Bill";

if (bi.length() > 0) {;

if (IsNumber(bi) == true) {

sql += " where id like %'" + bi + "'%";

}

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Bill> list = new ArrayList<Bill>();

while (rs.next()) {

Integer id = Integer.parseInt(rs.getString("id"));

Date dateCheckIn = rs.getDate("DateCheckIn");

int Ordernumber = rs.getInt("OrderNumber");

String Staffname = rs.getString("Staffname");

String CustomerPhoneNumber = rs.getString("CustomerPhoneNumber");

Float TotalPrice = rs.getFloat("TotalPrice");

Bill tc = new Bill(id, dateCheckIn, Ordernumber, Staffname, CustomerPhoneNumber, TotalPrice);

list.add(tc);

}

return list;

} catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

}

* + - 1. ***BillInfoDAO***

package DAO;

import Helper.JdbcHelper;

import Model.BillInfo;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

/\*\*

\*

\* @author Minh

\*/

public class BillInfoDAO {

public void insert(int idBill, int idDrinks, int Quantity,float Price) {

try {

String sql = "INSERT INTO BillInfo (idBill,idDrinks,Quantity,Price) VALUES (?, ?, ?, ?)";

JdbcHelper.executeUpdate(sql,

idBill,

idDrinks,

Quantity,

Price);

} catch (Exception e) {

e.printStackTrace();

}

}

public void update(BillInfo model) {

try {

String sql = "UPDATE BillInfo SET idBill=?, idDrinks=?, CountItem=? WHERE id=?";

JdbcHelper.executeUpdate(sql,

model.getIdBill(),

model.getIdDrinks(),

model.getQuantity(),

model.getPrice());

} catch (Exception e) {

System.out.println("Loi update");

}

}

public void delete(String id) {

try {

String sql = "DELETE FROM BillInfo WHERE id=?";

JdbcHelper.executeUpdate(sql, id);

} catch (Exception e) {

System.out.println("Loi xoa");

}

}

public boolean IsNumber(String num) {

if (num == null) {

return false;

}

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException ex) {

return false;

}

return true;

}

public List<BillInfo> findbyid(String bi) {

try {

String sql = "select \* from BillInfo";

if (bi.length() > 0) {

if (IsNumber(bi) == true) {

sql += "where id like %'" + bi + "'%";

}

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<BillInfo> list = new ArrayList<BillInfo>();

while (rs.next()) {

int idbill = rs.getInt("idBill");

int iddrink = rs.getInt("idDrinks");

int Quantity = rs.getInt("Quantity");

float price = rs.getFloat("Price");

BillInfo tc = new BillInfo(idbill, iddrink, Quantity,price);

list.add(tc);

}

return list;

} catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

}

* + - 1. ***CategoryDAO***

package DAO;

import Helper.JdbcHelper;

import Model.Category;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author Linh

\*/

public class CategoryDAO {

public boolean IsNumber(String num) {

if (num == null) {

return false;

}

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException ex) {

return false;

}

return true;

}

public List<Category> ShowCate(String namecategory){

try {

String sql = "select \* from Category";

if(namecategory.length()>0){

if(IsNumber(namecategory)==true) sql+=" where id like '%"+namecategory+"%'";

else

sql+= " where name like '%"+namecategory+"%'";

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Category> list = new ArrayList<Category>();

while(rs.next()){

int id = rs.getInt("id");

String name = rs.getString("name");

Category ctg = new Category(id, name);

list.add(ctg);

}

return list;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public void Insert(Category model){

try {

String sql="Insert dbo.Category values (?)";

JdbcHelper.executeQuery(sql, model.getName());

} catch (Exception e) {

System.out.println(e);

}

}

public void Delete(Category model){

try {

String sql="Delete from dbo.Category where id = ?";

JdbcHelper.executeQuery(sql, model.getId());

} catch (Exception e) {

System.out.println(e);

}

}

public void Update(Category model){

try {

String sql="Update dbo.Category set name = ? where id = ?";

JdbcHelper.executeQuery(sql, model.getName(),model.getId());

} catch (Exception e) {

System.out.println(e);

}

}

}

* + - 1. ***CustomerDAO***

package DAO;

import Helper.JdbcHelper;

import Model.Customer;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import java.util.logging.Level;

import java.util.logging.Logger;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Linh

\*/

public class CustomerDAO {

public void Insert(String phone, String name, int point) {

try {

String sql = "Insert into dbo.Customer(phone,name,point) values (?,?,?)";

JdbcHelper.executeUpdate(sql, phone,name,point);

} catch (Exception e) {

e.printStackTrace();

}

}

public void Delete(String phone) {

try {

String sql = "Delete from dbo.Customer where phone = ?";

JdbcHelper.executeUpdate(sql, phone);

} catch (Exception e) {

e.printStackTrace();

}

}

public void Update(Customer model) {

try {

String sql = "Update dbo.Customer set name = ? , point = ? where phone = ?";

JdbcHelper.executeUpdate(sql, model.getName(), model.getPoint(), model.getPhone());

} catch (Exception e) {

e.printStackTrace();

}

}

public boolean IsNumber(String num) {

if (num == null) {

return false;

}

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException ex) {

return false;

}

return true;

}

public List<Customer> findbyid(String tk) {

try {

String sql = "select \* from Customer";

if (tk.length() > 0) {

if (IsNumber(tk) == true) {

sql += " where Phone like '%" + tk + "%'";

}

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Customer> list = new ArrayList<Customer>();

while (rs.next()) {

String name = rs.getString("name");

String phone = rs.getString("phone");

Integer point = Integer.parseInt(rs.getString("point"));

Customer kh = new Customer(phone, name, point);

list.add(kh);

}

return list;

} catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

public boolean check(String phone)

{

String sql="select \* from customer where phone like ?";

ResultSet rs=JdbcHelper.executeQuery(sql, phone);

try {

if(rs.next())

{

return true;

}

} catch (SQLException ex) {

Logger.getLogger(CustomerDAO.class.getName()).log(Level.SEVERE, null, ex);

}

return false;

}

}

* + - 1. ***DrinksDAO***

package DAO;

import Helper.JdbcHelper;

import Model.Drinks;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author Minh

\*/

public class DrinksDAO {

public boolean isNumber(String num){

if(num == null) return false;

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException e) {

return false;

}

return true;

}

public boolean check(String ten) {

String sql = "select \* from Drinks where name like ?";

ResultSet rs = JdbcHelper.executeQuery(sql, ten);

try {

if (rs.next()) {

return true;

}

} catch (Exception e) {

System.out.println("loi");

}

return false;

}

public List<Drinks> ShowDrinks(String namedrink){

try {

String sql = "select \* from drinks";

if(namedrink.length()>0){

if(isNumber(namedrink)==true){

sql+= " where id = "+Integer.parseInt(namedrink)+"";

}

else{

sql+= " where name like '%"+namedrink+"%'";

}

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Drinks> list = new ArrayList<Drinks>();

while(rs.next()){

int id = rs.getInt("id");

String name = rs.getString("name");

int category = rs.getInt("idcategory");

float price = rs.getFloat("price");

Drinks dr = new Drinks(id, name, category, price);

list.add(dr);

}

return list;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public List<Drinks> ShowbyCate(int idcate){

try {

String sql = "select \* from drinks where idcategory ="+idcate;

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Drinks> list = new ArrayList<Drinks>();

while(rs.next()){

int id = rs.getInt("id");

String name = rs.getString("name");

int category = rs.getInt("idcategory");

float price = rs.getFloat("price");

Drinks dr = new Drinks(id, name, category, price);

list.add(dr);

}

return list;

} catch (Exception e) {

e.printStackTrace();

}

return null;

}

public void insert(String name,int cate,float price) {

try {

String sql = "insert into drinks(name,idcategory,price) values (?,?,?)";

JdbcHelper.executeUpdate(sql,

name,

cate,

price

);

} catch (Exception e) {

e.printStackTrace();

}

}

public void update(Drinks model) {

try {

String sql = "UPDATE Drinks SET name=?, idcategory=?, price=? WHERE id=?";

JdbcHelper.executeUpdate(sql,

model.getName(),

model.getIdcategory(),

model.getPrice(),

model.getId());

} catch (Exception e) {

// System.out.println("Loi update");

e.printStackTrace();

}

}

public void delete(int id) {

try {

String sql = "DELETE FROM Drinks WHERE id=?";

JdbcHelper.executeUpdate(sql, id);

} catch (Exception e) {

System.out.println("Loi xóa");

}

}

}

* + - 1. ***ImportDAO***

package DAO;

import Helper.JdbcHelper;

import Model.Import;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

/\*\*

\*

\* @author Minh

\*/

public class ImportDAO {

public void insert(Date importDate, int iddrinks, String idsupplier, int paymoney, int quantity) {

try {

String sql = "INSERT INTO Import (importDate,iddrinks,idsupplier,paymoney,quantity) VALUES (?, ?, ?, ?, ?)";

JdbcHelper.executeUpdate(sql,

importDate,

iddrinks,

idsupplier,

paymoney,

quantity);

} catch (Exception e) {

System.out.println("Loi insert");

}

}

public void update(Import model) {

try {

String sql = "UPDATE Bill SET DateCheckIn=?, DateCheckOut=?, idTable=?, status=?, totalprice=? WHERE id=?";

JdbcHelper.executeUpdate(sql,

model.getImportDate(),

model.getIddrinks(),

model.getIdsupplier(),

model.getPaymoney(),

model.getQuantity());

} catch (Exception e) {

System.out.println("Loi update");

}

}

public void delete(String id) {

try {

String sql = "DELETE FROM Import WHERE id=?";

JdbcHelper.executeUpdate(sql, id);

} catch (Exception e) {

System.out.println("Loi xoas");

}

}

public boolean IsNumber(String num) {

if (num == null) {

return false;

}

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException ex) {

return false;

}

return true;

}

public List<Import> findbyid(String bi) {

try {

String sql = "select \* from Import";

if (bi.length() > 0) {

if (IsNumber(bi) == true) {

sql += "where id like %'" + bi + "'%";

}

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Import> list = new ArrayList<Import>();

while (rs.next()) {

Date importDate = rs.getDate("importDate");

Integer iddrinks = Integer.parseInt(rs.getString("iddrinks"));

String idsupplier = rs.getString("idsupplier");

Float paymoney = Float.parseFloat("paymoney");

Integer quantity = Integer.parseInt(rs.getString("quantity"));

Import tc = new Import(importDate, iddrinks, idsupplier, paymoney, quantity);

list.add(tc);

}

return list;

} catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

}

* + - 1. ***StorageDAO***

package DAO;

import Helper.JdbcHelper;

import Model.Storage;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author Minh

\*/

public class StorageDAO {

public void insert(int iddrnk,int quantity) {

try {

String sql = "INSERT INTO Storage (iddrink,quantity) VALUES (?, ?)";

JdbcHelper.executeUpdate(sql,

iddrnk,

quantity);

} catch (Exception e) {

System.out.println("Loi insert");

}

}

public void update(Storage model) {

try {

String sql = "UPDATE Storage SET quantity=? WHERE iddrink=?";

JdbcHelper.executeUpdate(sql,

model.getQuantity(),model.getId());

} catch (Exception e) {

System.out.println("Loi update");

}

}

public void delete(String iddrink) {

try {

String sql = "DELETE FROM Storage WHERE iddrink=?";

JdbcHelper.executeUpdate(sql, iddrink);

} catch (Exception e) {

System.out.println("Loi xoas");

}

}

public boolean IsNumber(String num) {

if (num == null) {

return false;

}

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException ex) {

return false;

}

return true;

}

public boolean check(int id)

{

String sql = "select \* from Storage where iddrink like ?";

ResultSet rs = JdbcHelper.executeQuery(sql, id);

try {

if (rs.next()) {

return true;

}

} catch (Exception e) {

System.out.println("e");

}

return false;

}

public List<Storage> findbyid(String id) {

try {

String sql = "select \* from Storage";

if (id.length() > 0) {

sql += " where iddrink = " + id;

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Storage> list = new ArrayList<Storage>();

while (rs.next()) {

Integer iddrink = Integer.parseInt(rs.getString("iddrink"));

Integer quantity = Integer.parseInt(rs.getString("quantity"));

Storage ac = new Storage(iddrink, quantity);

list.add(ac);

}

return list;

} catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

public static boolean isNumberic(String strnumb) {

if (strnumb == null) {

return false;

}

try {

float d = Float.parseFloat(strnumb);

} catch (NumberFormatException e) {

return false;

}

return true;

}

public static boolean isNullOrEmpty(String str) {

if (str.equals("")) {

return false;

}

return true;

}

}

* + - 1. ***OrderNumberDAO***

package DAO;

import Helper.JdbcHelper;

import Model.OrderNumber;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Linh

\*/

public class OrderNumberDAO {

public void Insert(int id,boolean status) {

try {

String sql = "Insert into OrderNumber values (?,?)";

JdbcHelper.executeUpdate(sql, id,status);

} catch (Exception e) {

System.out.println(e);

}

}

public void Delete(int id) {

try {

String sql = "Delete from OrderNumber where id = ?";

JdbcHelper.executeUpdate(sql, id);

} catch (Exception e) {

JOptionPane.showMessageDialog(null, e);

}

}

public void Update(OrderNumber model) {

try {

String sql = "Update OrderNumber set status = ? where id = ?";

if(model.isStatus())

JdbcHelper.executeUpdate(sql,1, model.getId());

else

JdbcHelper.executeUpdate(sql,0, model.getId());

} catch (Exception e) {

e.printStackTrace();

}

}

public List<OrderNumber> findbyid(String tb) {

try {

String sql = "select \* from OrderNumber";

if (tb.length() > 0) {

sql += " where id like %'" + tb + "'%";

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<OrderNumber> list = new ArrayList<OrderNumber>();

while (rs.next()) {

Integer id = Integer.parseInt(rs.getString("id"));

boolean status = rs.getBoolean("status");

OrderNumber tc = new OrderNumber(id, status);

list.add(tc);

}

return list;

} catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

public List<OrderNumber> findbystatus(boolean status) {

try {

String sql = "select \* from OrderNumber";

if (status) {

sql += " where status = 1";

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<OrderNumber> list = new ArrayList<OrderNumber>();

while (rs.next()) {

Integer id = Integer.parseInt(rs.getString("id"));

boolean Status = rs.getBoolean("status");

OrderNumber tc = new OrderNumber(id, Status);

list.add(tc);

}

return list;

} catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

}

* + - 1. ***SuppierDAO***

package DAO;

import Helper.JdbcHelper;

import Model.Supplier;

import java.sql.ResultSet;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author Minh

\*/

public class SupplierDAO {

public void insert(String id, String name) {

try {

String sql = "INSERT INTO supplier (id,name) VALUES (?, ?)";

JdbcHelper.executeUpdate(sql,

id,

name);

} catch (Exception e) {

System.out.println("Loi insert");

}

}

public void update(Supplier model) {

try {

String sql = "UPDATE supplier SET name=? WHERE id=?";

JdbcHelper.executeUpdate(sql,

model.getId(), model.getName());

} catch (Exception e) {

System.out.println("Loi update");

}

}

public void delete(String id) {

try {

String sql = "DELETE FROM supplier WHERE id=?";

JdbcHelper.executeUpdate(sql, id);

} catch (Exception e) {

System.out.println("Loi xoas");

}

}

public boolean IsNumber(String num) {

if (num == null) {

return false;

}

try {

int n = Integer.parseInt(num);

} catch (NumberFormatException ex) {

return false;

}

return true;

}

public List<Supplier> findbyid(String id) {

try {

String sql = "select \* from supplier";

if (id.length() > 0) {

if(IsNumber(id)==true) sql += " where id=" + id;

else

sql+=" where name like '%"+id+"%'";

}

ResultSet rs = JdbcHelper.executeQuery(sql);

List<Supplier> list = new ArrayList<Supplier>();

while (rs.next()) {

String ids = rs.getString("id");

String name = rs.getString("name");

Supplier ac = new Supplier(ids, name);

list.add(ac);

}

return list;

} catch (Exception ex) {

ex.printStackTrace();

}

return null;

}

}

### 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.

* + - 1. **AccountManage**
      2. **BillHistory**
      3. **CategoryFRM**
      4. **CustomerManager1**
      5. **Dashboard**
      6. **Login**
      7. **Menu**
      8. **MenuManage**
      9. **StorageFRM**
      10. **ThongKeFRM**
      11. ***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

package Helper;

//import Model.NhanVien;

import Model.Account;

import java.awt.Image;

import java.io.File;

import java.nio.file.Files;

import java.nio.file.Path;

import java.nio.file.Paths;

import java.nio.file.StandardCopyOption;

import javax.swing.ImageIcon;

/\*\*

\*

\* @author Linh

\*/

public class ShareHelper {

public static final Image APP\_ICON;

static {

// Tải biểu tượng ứng dụng

String file = "/logo.jpg";

APP\_ICON = new ImageIcon(ShareHelper.class.getResource(file)).getImage();

}

public static boolean saveLogo(File file) {

File dir = new File("logos");

// Tạo thư mục nếu chưa tồn tại

if (!dir.exists()) {

dir.mkdirs();

}

File newFile = new File(dir, file.getName());

try {

// Copy vào thư mục logos (đè nếu đã tồn tại)

Path source = Paths.get(file.getAbsolutePath());

Path destination = Paths.get(newFile.getAbsolutePath());

Files.copy(source, destination, StandardCopyOption.REPLACE\_EXISTING);

return true;

} catch (Exception ex) {

return false;

}

}

public static ImageIcon readLogo(String fileName) {

File path = new File("logos", fileName);

return new ImageIcon(path.getAbsolutePath());

}

public static Account USER = null;

public static void logoff() {

ShareHelper.USER = null;

}

public static boolean authenticated() {

return ShareHelper.USER != null;

}

* + - 1. ***DialogHelper***

Lớp DialogHelper chứa các phương thức đơn giản hóa JOptionPane

package Helper;

import java.awt.Component;

import javax.swing.JOptionPane;

/\*\*

\*

\* @author Linh

\*/

public class DialogHelper {

public static void alert(Component parent, String message) {

JOptionPane.showMessageDialog(parent, message, "Hệ thống quản lý Cafe", JOptionPane.INFORMATION\_MESSAGE);

}

public static boolean confirm(Component parent, String message) {

int result = JOptionPane.showConfirmDialog(parent, message, "Hệ thống quản lý Cafe", JOptionPane.YES\_NO\_OPTION, JOptionPane.QUESTION\_MESSAGE);

return result == JOptionPane.YES\_OPTION;

}

public static String prompt(Component parent, String message) {

return JOptionPane.showInputDialog(parent, message, "Hệ thống quản lý Cafe", JOptionPane.INFORMATION\_MESSAGE);

}

## 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.

# KIỂM THỬ

4.1 KIỂM THỬ FORM QUẢN LÝ ACCOUNT MANAGE

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Thao tác | | |
| Add | Update | Xóa |
| Thông báo thêm thành công | Thông báo update thành công | Thông báo xóa thành công |
| Username | Không để trống | Không để trống |
| Displayname | Không để trống | Không để trống |
| Password | Không để trống | Không để trống |
| Type | Không để trống | Không để trống |

4.2 KIỂM THỬ FORM CUSTOMER MANAGE

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Thao tác | | |
| Add | Update | Xóa |
| Thông báo thêm thành công | Thông báo update thành công | Thông báo xóa thành công |
| Phone | Không để trống | Không để trống |
| Name | Không để trống | Không để trống |
| Points | Không để trống | Không để trống |

4.3.KIỂM THỬ FORM MENU MANAGE

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Thao tác | | |
| Add | Update | Xóa |
| Thông báo thêm thành công | Thông báo update thành công | Thông báo xóa thành công |
| ID | Không để trống | Không để trống |
| Name | Không để trống | Không để trống |
| Price | Không để trống | Không để trống |
| Category | Không để trống | Không để trống |

4.4.KIỂM THỬ FORM STORAGE

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Thao tác | | |
| Add | Update | Xóa |
| Thông báo thêm thành công | Thông báo update thành công | Thông báo xóa thành công |
| ID | Không để trống | Không để trống |
| Quantity | Không để trống | Không để trống |

4.5.KIỂM THỬ FORM CATEGORY

|  |  |  |  |
| --- | --- | --- | --- |
| Thuộc tính | Thao tác | | |
| Add | Update | Xóa |
| Thông báo thêm thành công | Thông báo update thành công | Thông báo xóa thành công |
| ID | Không để trống | Không để trống |
| Name | Không để trống | Không để trống |

# ĐÓNG GÓI VÀ TRIỂN KHAI

### HƯỚNG DẪN CHUYỂN ĐỔI JAR THÀNH EXE

* + 1. Sử dụng phần mềm EXE4J để chuyển đổi jar sang exe
    2. Sử dụng phần mềm InnoSetup để đóng gói

### HƯỚNG DẪN CÀI ĐẶT TRIỂN KHAI

* + 1. Cài SQL Server 2008 trở lên
    2. Tạo CSDL Polypro bằng cách chạy file Polypro.sql
    3. Cài đặt JDK 1. trở lên
    4. Chạy file setup

### HƯỚNG DẪN SỬ DỤNG PHẦN MỀM

* + 1. Đăng nhập với tài khoản username: admin và password:admin
    2. Cấp các tài khoản có vai trò là trưởng phòng
    3. Đăng nhập tài khoản mới và xóa tài khoản admin
    4. Xem hướng dẫn sử dụng trong menu trợ giúp của phần mềm