**TÀI LIỆU PHÁT THẢO DỰ ÁN**

**I. QUẢN LÝ DỰ ÁN:**

**1. Tên đề tài:**

Phần mềm quản lí hoạt động bán hàng ở quán cafe

**2. Mục tiêu:**

- Quản lí hoạt động bán hàng ở quán cafe một cách thuận tiện và tiết kiệm thời gian.

- Dễ dàng quản lí cửa hàng, quản lí nhân viên

- Tối ưu quy trình bán hàng, giúp hạn chế sai sót trong khâu order và pha chế thức uống.

- Nhanh chóng nắm bắt được quy trình kinh doanh, tổng kết doanh thu.

**3. Phát thảo outline:**

- Quản lí bàn (thêm, chỉnh sửa trạng thái)

- Quản lí hóa đơn (thêm, chỉnh sửa, thanh toán)

- Quản lí menu (thêm, sửa, xóa)

- Quản lí doanh thu (chỉnh sửa)

**4. Viết tài liệu phát thảo dự án:**

**4.1. Quản lý bàn:**

- Quán có nhiều bàn, mỗi bàn có Số bàn, Trạng thái của bàn (trống hoặc đã có khách). Những bàn đã có khách thì không thể đặt trước.

- Khi khách hàng ngồi vào bàn, nhân viên tiến hành ghi order và mở bàn bằng cách click chọn bàn trong danh sách bàn để mở bàn.

- Trong quá trình khách ngồi, nếu khách yêu cầu đổi bàn, nhân viên sẽ tiến hành chuyển bàn cho khách.

- Nếu bàn đó khách hàng đã ra về, chuyển bàn hoặc hủy đặt bàn thì cho phép nhân viên xóa bàn đó khỏi danh sách đang hoạt động.

- Cho biết bàn nào còn trống bàn nào có khách, bàn nào đã được khách hàng đặt trước.

- Khi có khách hàng đặt bàn thì nhân viên mở bàn và điền thông tin của khách hàng đó vào bàn đã được đặt và set trạng thái của bạn đó thành đã có khách.

**4.2. Quản lí hóa đơn:**

- Hóa đơn sẽ được lập khi khách thanh toán ra về. Mỗi bàn thanh toán với 1 hóa đơn duy nhất. Hóa đơn sẽ có ID riêng, thời gian vào, thời gian ra, ngày lập hóa đơn, ID bàn mà khách ngồi và trạng thái của hóa đơn (Đã thanh toán hay chưa thanh toán).

- Trường hợp khách chuyển bàn thì hóa đơn tạm tính ở bàn cũ sẽ giữ nguyên và cộng qua bàn mới.

- Khi xuất hóa đơn cho khách thì sẽ có đầy đủ thông tin ID hóa đơn xuất, ID hóa đơn của đồ uống với số lượng và đơn giá. Sau khi xuất thì hóa đơn sẽ cộng tiền vào bảng thống kê của phần mềm.

- Lưu trữ hóa đơn: Sau khi bán hàng, thông tin hóa đơn sẽ được lưu trữ lại nhằm theo dõi tình hình bán hàng của quán của một ngày

**4.3. Quản lí Menu:**

- Quán sẽ có menu bao gồm các đồ uống của quán.

- Khi quán có thêm món mới, thì nhân viên sẽ thêm món đó vào hệ thống.

- Hoặc là một thức uống nào đó quán không còn chế biến nữa thì sẽ xóa nó ra khỏi hệ thống.

- Ngoài ra, giá của mỗi thức uống có thể thay đổi theo thời gian nên có thể điều chỉnh giá cho phù hợp.

**4.4. Quản lý doanh thu:**

- Vào cuối của mỗi ngày, phần mềm sẽ tổng hợp lại tổng số hóa đơn đã lưu trữ lại của ngày hôm đó. Sau đó sẽ tổng kết doanh thu của một ngày và lưu trữ lại để nhằm theo dõi tình hình doanh thu bán hàng của quán của một tháng.

- Sau một tháng, phần mềm sẽ tính doanh thu của một tháng, theo sự lưu trữ doanh thu của từng ngày.

**5. Tài liệu yêu cầu:**

**5.1. Quản lí bàn**: Quản lí việc thêm bớt số lượng bàn, biết được số bàn đang được sử dụng và số bàn còn trống.

- Bàn: quán có nhiều bàn, mỗi bàn có: Mã số bàn, Vị trí, Số chỗ ngồi, Trạng thái.

- Số lượng bàn: Tổng số bàn hiện tại quán đang có.

- Trạng thái bàn: đã có người sử dụng hoặc chưa có người sử dụng.

- Xóa bàn khỏi danh sách bàn đang được sử dụng: Sau khi khách hàng thanh toán xong và rời khỏi bàn thì chuyển trạng thái cảu bàn từ đang sử dụng thành trống.

⮚ Ràng buộc:

- Mã của mỗi bàn không được trùng lặp.

**5.2. Quản lí hóa đơn**: khi khách order thì số lượng đồ uống kèm giá mà khác đã gọi sẽ được ghi ở hóa đơn thanh toán kèm với số bàn mà khách đang sử dụng và thời gian xuất hóa đơn.

- Thời gian nhập: sau khi khách hàng gọi món xong sẽ lưu lại thời gian khách hàng đã gọi.

- Tên nhân viên lập hóa đơn: Sau khi khách order xong nhân viên phụ trách sẽ lập hóa đơn.

- Số lượng đồ uống khách đã gọi kèm đơn giá: mỗi đồ uống khách gọi sẽ kèm theo số lượng, đơn giá và tổng tiền của món đó.

- Tổng tiền: Tổng tiền của tất cả các đồ uống mà khách đã gọi.

- Thời gian xuất: được in ở cuối hóa đơn khi khách hàng gọi thanh toán.

⮚ Ràng buộc:

- Thời gian vào lớn hơn thời gian ra.

- Tổng tiền là một số không âm.

- Số lượng đồ uống là một số dương.

**5.3. Quản lí menu**: Gồm dánh sách các món của quán có kèm theo giá tiền được sắp xếp theo danh mục.

- Danh sách các đồ uống kèm giá.

- Mỗi đồ uống sẽ có một mã và tên gọi riêng.

- Mỗi đồ uống trong menu sẽ có 1 giá thành riêng đã được định trước.

- Sắp xếp món trong menu: các thức uống cùng một nhóm đồ uống sẽ được sắp xếp vào một mục riêngThêm, xóa, sửa các đồ uống trong menu.

- Thêm đồ uống: khi quán cho ra một đồ uống mới thì đồ uống đó sẽ được thêm vào menu kèm giá và được xếp vào một danh mục thức uống thích hợp.

- Xóa đồ uống: khi một đồ uống không còn được bán tai quán nữa thì sẽ tiến hành xóa đồ uống đó ra khỏi menu.

⮚ Ràng buộc:

- Mã của mỗi đồ uống trong menu sẽ riêng biệt nhau.

- Đơn giá phải được tính bằng VNĐ.

**5.4. Quản lí doanh thu:**

- Vào cuối của mỗi ngày, phần mềm sẽ tổng hợp lại tổng số hóa đơn đã lưu trữ lại của ngày hôm đó. Sau đó sẽ tổng kết doanh thu của một ngày và lưu trữ lại để nhằm theo dõi tình hình doanh thu bán hàng của quán của một tháng.

- Sau một tháng, phần mềm sẽ tính doanh thu của một tháng, theo sự lưu trữ doanh thu của từng ngày.

**6. Lập lịch dự án**

❖ Lập mã công việc

• Thành lập nhóm, chọn đề tài (A)

• Lập kế hoạch khảo sát sơ bộ (B)

• Khảo sát địa điểm, phỏng vấn nhân viên - quản lí (C)

• Phát thảo outline dự án (D)

• Lập lịch ban đầu cho dự án đã phát thảo (E)

• Hoàn thành tài liệu yêu cầu của dự án (F)

• Thảo luận thống nhất công nghệ, cơ sở dữ diệu lập trình (G)

• Viết code cho phần giao diện Menu (H)

• Viết code cho phần giao diện Thanh toán (I)

• Viết code cho phần giao diện Điều chỉnh thông tin đồ uống (thêm đồ uống, chỉnh sửa giá, xóa đồ uống quán ko còn pha chế nữa) (K)

• Viết code cho phần giao diện Doanh thu (L)

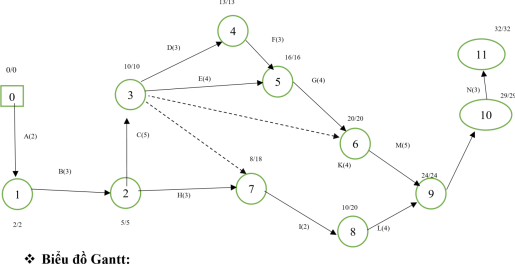
• Thực nghiệm và chỉnh sửa phần mềm (M)

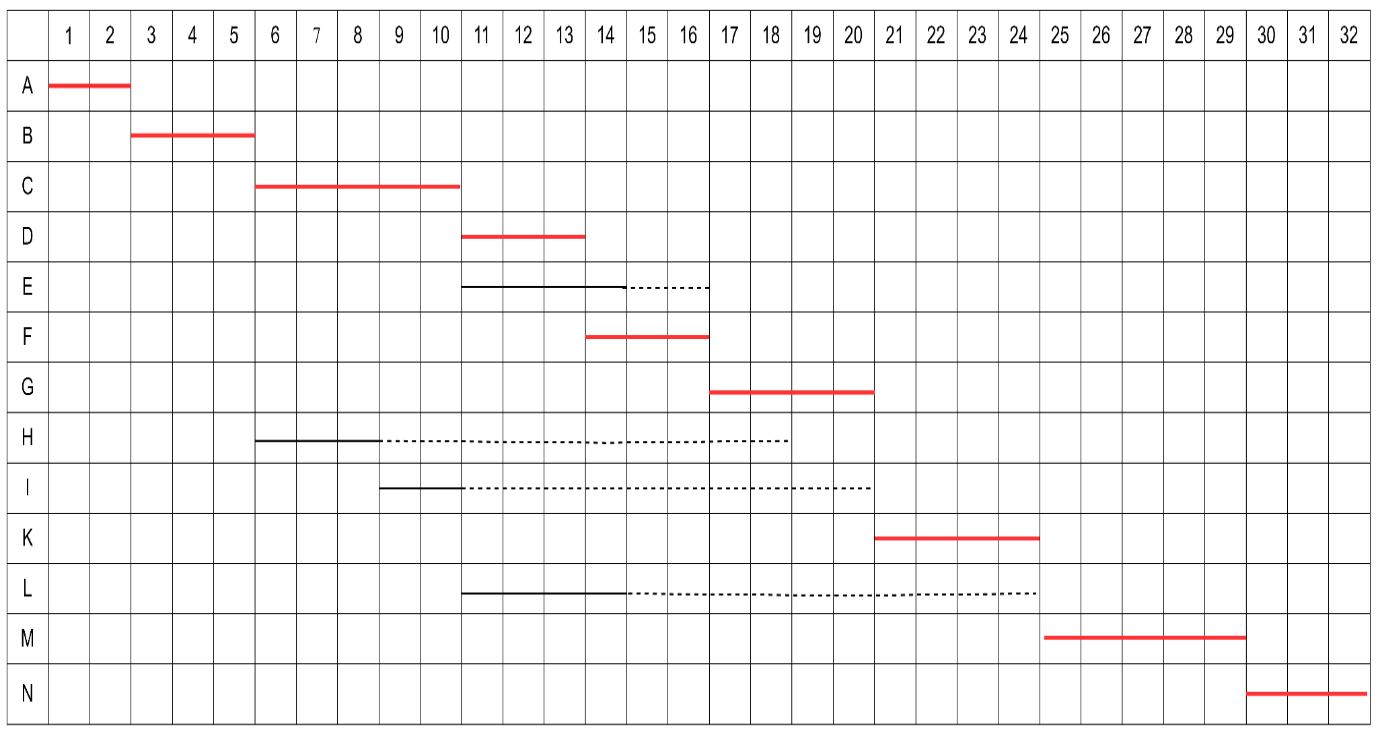
• Bàn giao dự án (N)

❖ Xác định công việc đi trước và thời gian thực hiện mỗi công việc

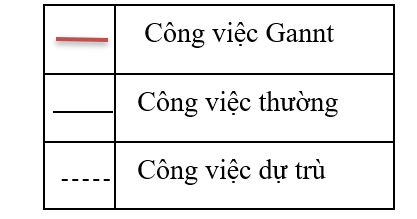
|  |  |  |
| --- | --- | --- |
| Công việc | Thời gian | Công việc đi trước |
| A | 2 | - |
| B | 3 | A |
| C | 5 | B |
| D | 3 | C |
| E | 4 | C |
| F | 3 | D |
| G | 4 | E,F |
| H | 3 | B |
| I | 2 | C,H |
| K | 4 | C,G |
| L | 4 | I |
| M | 5 | K,L |
| N | 3 | M |

❖ Lập mạng công việc:





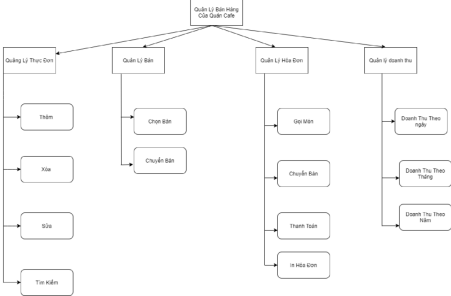
❖ Chú thích:



**II. HỒ SƠ THIẾT KẾ DỰ ÁN:**

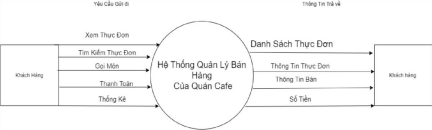
**1.Hồ sơ thiết kế chức năng của hệ thống**

**a. Biểu đồ phân rã chức năng (BFD)**

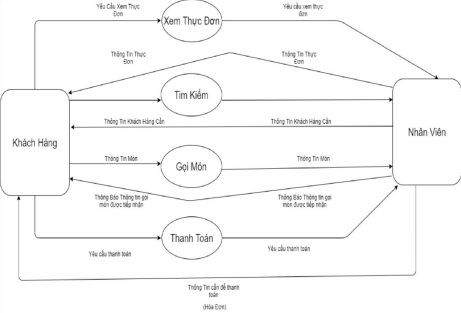
****

**b. Biểu Đồ Phân Luồng Dữ Liệu (DFD)**

❖ **Mức 0 (Mức ngữ cảnh)**

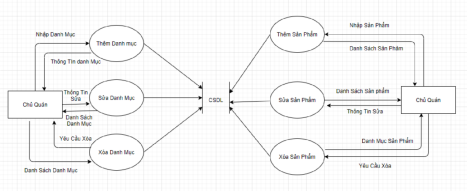


❖ **Mức 1 (Mức đỉnh)**

****

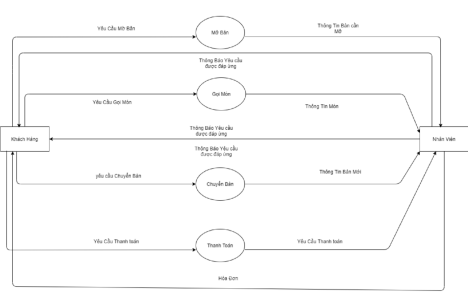
❖ **Mức 2( Mức chi tiết)**

⮚ ***Chức Năng quản lý thực đơn***

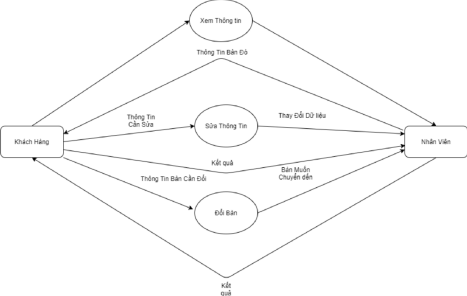
******

⮚ ***Chức Năng thống kê doanh thu***

****** ⮚ ***Chức năng quản lý hóa đơn***



⮚ ***Chức năng quản lí bàn***



**2. Hồ sơ thiết kế CSDL:**

**a. Sơ đồ thực thể mối quan hệ ERD:** Các thực thể:

- STAFFS (**STAFFS\_Id**, STAFFS\_Name, STAFFS\_Position, STAFFS\_Birthday,

STAFFS\_Address, STAFFS\_Datein, STAFFS\_Phone)

- TABLES (**TABLES\_Id**, TABLES\_Name, TABLES\_Status)

- ZONES (**ZONES\_Id**, ZONES\_Name, ZONES\_Table)

- TYPE\_DRINKS (**TYPE\_DRINKS\_Id**, TYPE\_DRINKS\_Name)

- DRINKS (**DRINKS\_Id**, DRINKS\_Name, DRINKS\_Price,

DRINKS\_Status, TYPE\_DRINKS\_Id)

- INFO\_BILLS (**BILLS\_Id**, **DRINKS\_Id,** INFO\_Amount, INFO\_Price)

-BILLS (**BILLS\_Id**, BILLS\_TimeIn, BILLS\_TimeOut, STAFFS\_id, BILLS\_Price, TABLE\_Id)

⮚ Từ yêu cầu của bài toán ta xác định được các mối quan hệ sau:

- Mỗi bàn có 1 hóa đơn

- Mỗi hóa đơn có nhiều đồ uống

=>DRINKS và INFO\_BILLS có quan hệ **n-1**

- Một bàn chỉ thuộc về 1 khu vực,1 khu vực có nhiều bàn

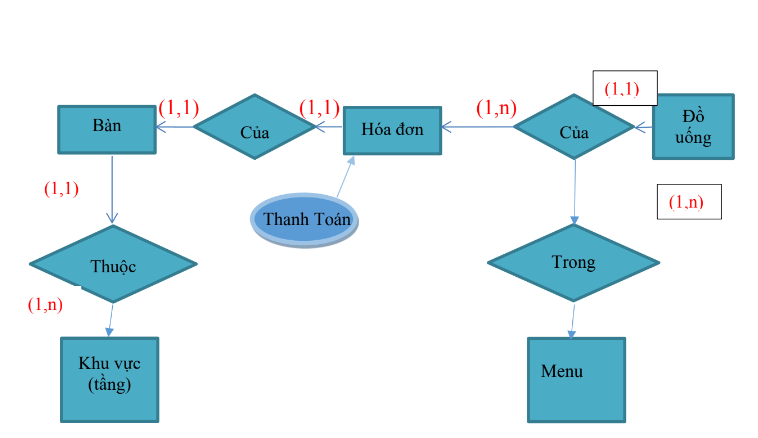
=>ZONES và TABLES có quan hệ **1-n**

- Mỗi nhân viên thì làm việc ở 1 khu vực, 1 khu vực có nhiều nhân viên

=>STAFFS và ZONES có quan hệ **n-1**

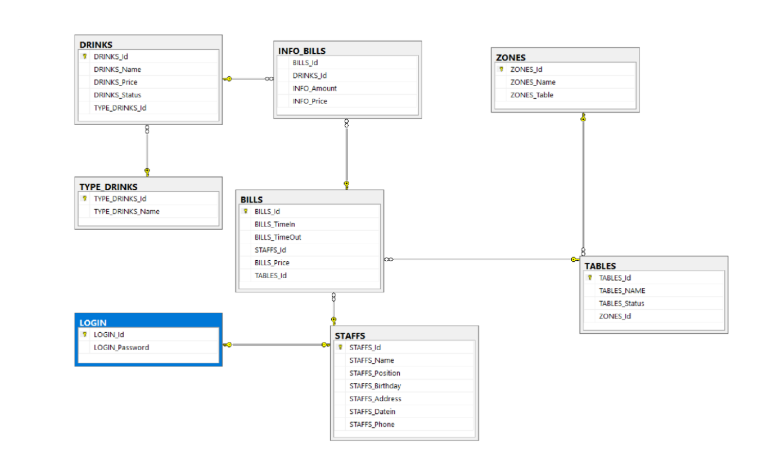
- Mỗi đồ uống thì thuộc 1 loại,1 loại có nhiều đồ uống

=>DRINKS và TYPE\_DRINKS có quan hệ **n-1**



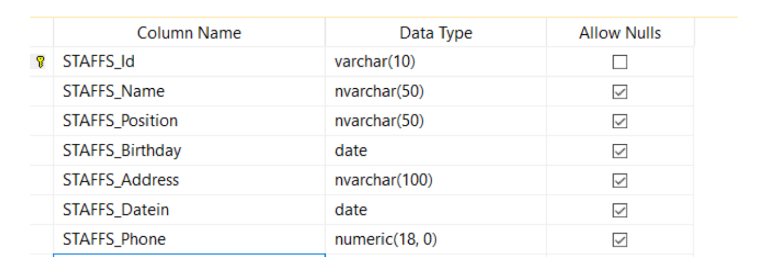
**Sơ đồ thực thể mối quan hệ (ERD)**

**b. Lược đồ quan hệ:**



**c. Chi tiết các bảng**

❖ **Bảng STAFFS**



Bảng STAFFS sẽ lưu thông tin của tất cả nhân viên trong quán, bao gồm:

+ STAFFS\_Id: Lưu mã nhân viên, mỗi mã nhân viên chỉ dành cho một nhân viên duy nhất.

+ STAFFS \_Name: Lưu tên của nhân viên.

+ STAFFS \_Position: Lưu chức vụ của nhân viên (nhân viên, quản lí).

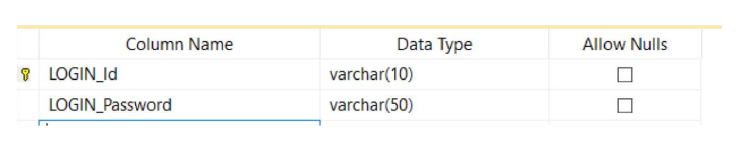
+ STAFFS \_Birthday: Lưu ngày tháng năm sinh của nhân viên.

+ STAFFS \_Address: Lưu địa chỉ của nhân viên.

+ STAFFS \_Datein: Lưu ngày nhân viên vào làm tại quán.

+ Staffs\_Phone: Lưu số điện thoại của nhân viên.

❖ **Bảng LOGIN**

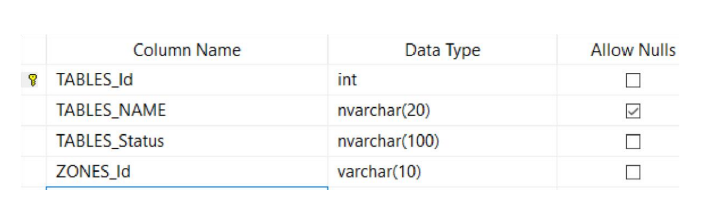


Bảng LOGIN chứa thông tin đăng nhập của mỗi nhân viên khi đăng nhập vào app:

+ LOGIN\_ID: Chứa mã nhân viên khi nhân viên đăng nhập, LOGIN\_Id là duy nhất đối với mỗi nhân viên và trùng với STAFFS\_Id tương ứng của nhân viên đó

+ LOGIN\_Password: Chứa mật khẩu của mỗi nhân viên khi đăng nhập

❖ **Bảng TABLES**



Bảng TABLES chứa thông tin của các bàn trong quán:

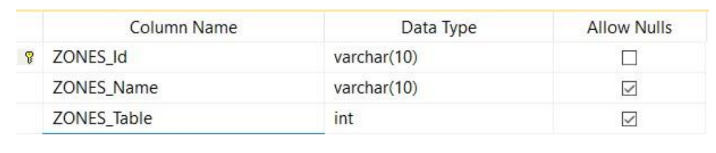
+ TABLES\_Id: Chứa mã của bàn và mỗi mã chỉ dành cho một bàn duy nhất

+ TABLES\_Name: Chứa tên của bàn

+ TABLES\_Status: Chứa trạng thái của bàn (còn trống hay đang có khách)

+ ZONES\_Id: Chứa khu vực mà bàn đang được đặt

❖ **Bảng ZONES**



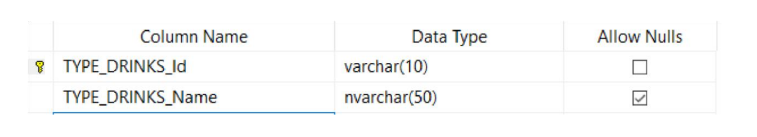
Bảng ZONES chứa thông tin của các khu vực trong quán:

+ ZONES\_ID: Chứa mã của mỗi khu vực và tương ứng với mỗi khu vực thì chỉ có một mã duy nhất

+ ZONES\_Name: Chứa tên của khu vực

+ ZONES\_Table: chứa số lượng bàn trong khu vực đó

❖ **Bảng TYPE\_DRINKS**

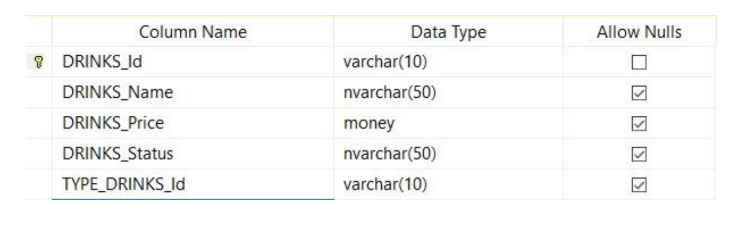


Bảng TYPE\_DRINKS chứa thông tin của các nhóm đồ uống:

+ TYPE\_DRINKS\_Id: chứa mã các nhóm đồ uống và mỗi nhóm đồ uống chỉ có một mã duy nhất

+ TYPE\_DRINKS\_Name: chứa tên của các nhóm đồ uống

❖ **Bảng DRINKS**



Bảng DRINKS chứa thông tin của các loại đồ uống:

+ DRINKS\_Id: Chứa mã của các loại đồ uống, mỗi đồ uống chỉ có một mã duy nhất

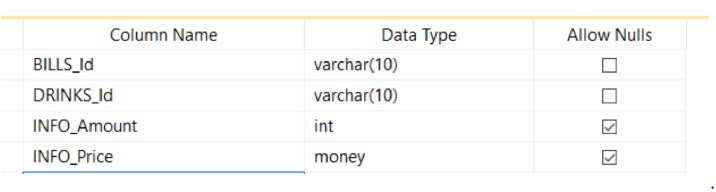
+ DRINKS\_Name: Chứa tên của các loại đồ uống

+ DRINKS\_Price: Chứa giá của các loại đồ uống

+ DRINKS\_Status: Chứa tình trạng của các loại đồ uống

+ TYPE\_DRINKS\_Id: Chứa mã của nhóm đồ uống chứa loại đồ uống đó

❖ **Bảng INFO\_BILL**



Bảng INFO\_BILL chứa thông tin chi tiết của hóa đơn:

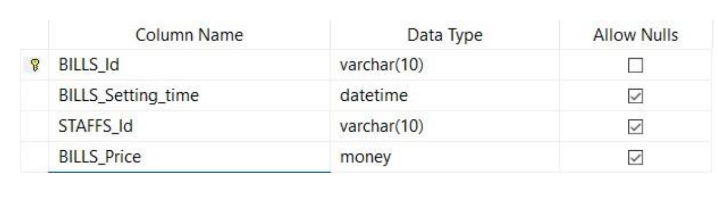
+ BILLS\_Id: Chứa mã của hóa đơn chi tiết mỗi mã chỉ dành cho duy nhất một hóa đơn và mã hóa đơn này trùng với mã hóa đơn tổng quát của nó khi thanh toán

+ DRINKS\_Id: chứa mã của các loại đồ uống trong hóa đơn

+ INFO\_BILL\_Amount: Chứa số lượng của mỗi loại đồ uống trong hóa đơn

+ INFO\_BILL\_Price: Chứa tổng giá của mỗi loại đồ uống trong hóa đơn

❖ **Bảng BILLS**



Bảng BILLS chứa thông tin của hóa đơn khi xuất ra cho khách hàng hay lưu doanh thu vào hệ thống:

+ BILLS\_Id: Chứa mã hóa đơn, mỗi mã chỉ dành cho một hóa đơn duy nhất.

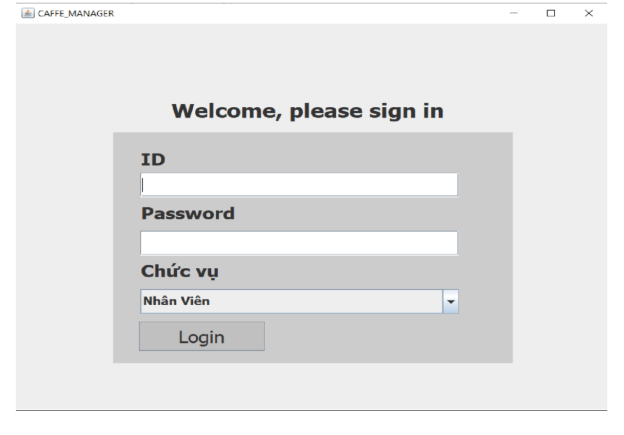
+ BILLS\_Setting\_time: Chứa thời gian lập hóa đơn

+ STAFF\_Id: Chứa mã nhân viên lập hóa đơn

+ BILLS\_Price: Chứa tổng tiền của hóa đơn

**3. Hồ sơ thiết kế giao diện chức năng**

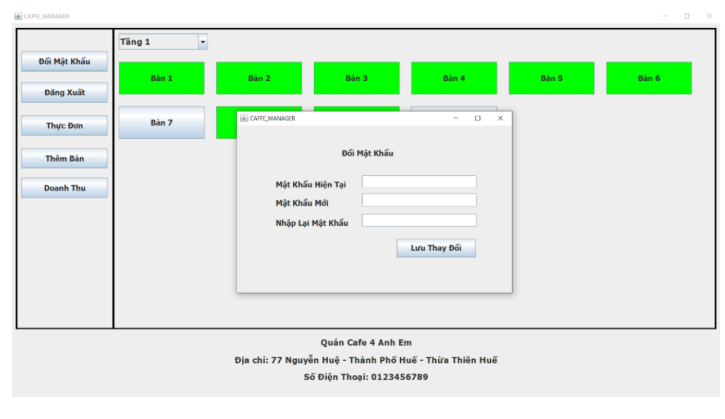
⮚ **Chức năng đăng nhập**





|  |  |  |  |
| --- | --- | --- | --- |
| Tên  hành  động | Mô tả | Thành  công | Thất bại |
| Nhập  thông  tin | Nhập đầy đủ các  thông tin vào các  trường đã được đã được chỉ định sẵn |  |  |
| Đăng  nhập | Nhấn vào button “Login” để đăng nhập vào app | Vào giao diện chính của app | Yêu cầu nhập lại thông tin sai hoặc bổ xung thông tin thiếu. |

⮚ **Chức năng đổi mật khẩu:**



|  |  |  |  |
| --- | --- | --- | --- |
| Tên  hành  động | Mô tả | Thành  công | Thất bại |
| Nhập  thông tin | Nhập đầy đủ  các thông tin  vào các trường đã được đã được chỉ định sẵn |  |  |
| Đăng  nhập | Nhấn vào  button “Lưu  thay đổi” để đổi mật khẩu | Đổi mật khẩu mới | Yêu cầu nhập lại thông tin sai hoặc bổ xung thông tin thiếu. |

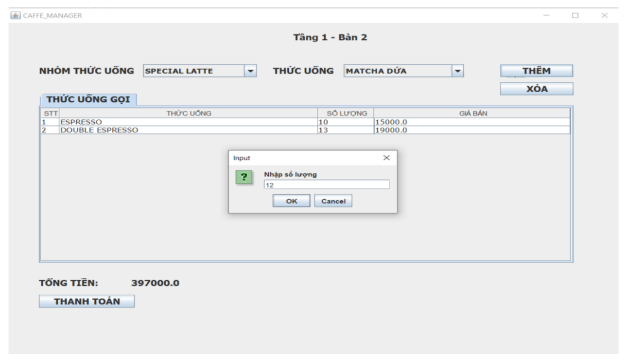
❖ **Chức năng quản lí bàn:**



|  |  |  |  |
| --- | --- | --- | --- |
| Tên hành động | Mô tả | Thành công | Thất bại |
| Thêm  bàn | Nhấn vào button “Thêm bàn” | Xuất hiện thông báo thêm bàn |  |
| Nhập số lượng  bàn | Nhập số lượng bàn cần thêm vào cửa sổ thêm bàn | Màn hình chính của app sẽ xuất hiện thêm số bàn đã nhập |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Nhấn chọn bàn trống cần mở | Xuất hiện thông báo mở bàn, sau đó nhấn “Yes” |  |

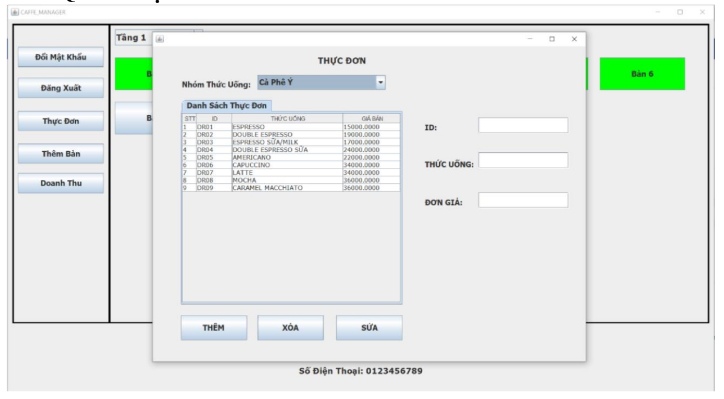
❖ **Chức năng quản lý hóa đơn**



|  |  |  |  |
| --- | --- | --- | --- |
| Tên  hành  động | Mô tả | Thành công | Thất bại |
| Thêm  Đồ  Uống | Chọn vào bàn  cần thêm đồ  uống | Xuất hiện cửa sổ trên |  |
| Nhập  số  lượng | Chọn đồ uống  cần thêm sau  đó nhập số  lượng rồi bấm  “ OK” | Đồ uống sẽ xuất hiện ở hóa đơn. |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Thanh  toán | Bấm Vào  button “thanh  toán” | Sau Đó nhấn “Yes” để in hóa đơn |  |

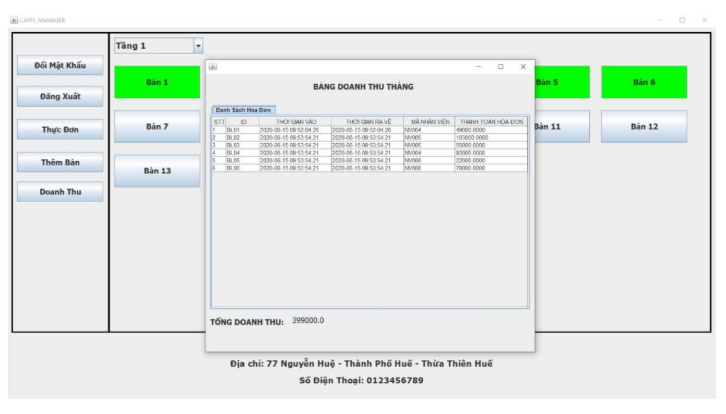
❖ **Quản lý thực đơn**



|  |  |  |  |
| --- | --- | --- | --- |
| Tên hành động | Mô tả | Thành công | Thất bại |

|  |  |  |  |
| --- | --- | --- | --- |
| Sửa thực đơn | Nhấn vào button “Thực đơn” | Xuất hiện danh sách các thức uống và nhóm thức uống |  |
| Thêm  thức uống | Nhập đầy đủ các thông tin của thức uống sau đó nhấn vào button “thêm” | App sẽ thêm một đồ uống mới vào danh sách đồ uống |  |
| Xóa thức uống | Chọn đồ uống  muốn xóa và nhân vào button “xóa” | App sẽ loại đồ uống đó ra khỏi danh sách đồ uống |  |
| Sửa đồ  uống | Chọn vào đồ  uống muốn sửa, sau đó sửa các  thông tin đồ uống rồi nhân button “sửa” | App sẽ update lại thông tin đồ uống vừa mới chỉnh sửa |  |

❖ **Chức năng quản lí doanh thu**



|  |  |  |  |
| --- | --- | --- | --- |
| Tên hành động | Mô tả | Thành công | Thất bại |
| Xem  doanh  thu | Nhấn vào button “doanh thu” | Xuất hiện danh sách các hóa đơn đã lập trong tháng và tổn thu nhập của quán trong tháng đó |  |
|  |  |  |  |

❖ **Chức năng đăng xuất**



|  |  |  |  |
| --- | --- | --- | --- |
| Tên hành động | Mô tả | Thành công | Thất bại |
| Đăng  xuất | Nhấn vào button “đăng xuất” | Xuất hiện thông báo nhắc bạn có muốn đăng xuất hay không |  |
| Đăng  xuất | Nhấn vào  button”Yes” | Thoát ra khỏi giao diện làm việc của app | Giữ nguyên giao diện cũ nhưng  thôn báo biến mất |

**III. HỒ SƠ CÀI ĐẶT:**

**1. Tổng quan**

Ngôn ngữ lập trình: Java

Phần mềm lập trình: Eclipse, Microsoft SQL server

**2. Cài đặt chương trình**

❖ **Chức Năng Đăng Nhập**

**public** Boolean Login(String id, String password, String position) { **try** {

dao\_connective.*connective*();

**if**(Login\_account(id, password) && Login\_postion(id, position)) **return true**;

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return false**;

}

**Chức Năng Đổi Mật Khẩu:**

**public boolean** changepassword(String id, String password, String newpassword1, String newpassword2) {

**try** {

dao\_connective.*connective*();

**if**(Kiemtra\_newpassword(newpassword1, newpassword2))

**if**(Kiemtra\_password(id, password)) {

String sql = "UPDATE LOGIN SET

LOGIN\_Password = ? WHERE LOGIN\_Id = ?"; PreparedStatement cmd = dao\_connective.*cn*.prepareStatement(sql); cmd.setString(1, newpassword1);

cmd.setString(2, id);

cmd.executeUpdate();

JOptionPane.showMessageDialog(**null**, "Đổi

mật khẩu thành công!");

**return true**;

}

27 | P a g e

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return false**;

}

❖ **Chức Năng Quản lý Bàn :**

**public void** get\_zones\_tables() {

**try** {

String sql = "SELECT \* FROM TABLES WHERE

ZONES\_Id = ?";

PreparedStatement cmd =

dao\_connective.*cn*.prepareStatement(sql);

cmd.setString(1, zones\_id);

ResultSet rs = cmd.executeQuery();

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

table++;

button\_tables = **new** JButton();

String table\_id =

String.valueOf(rs.getInt("TABLES\_Id"));

String table\_name =

rs.getString("TABLES\_NAME");

button\_tables.setName(table\_id);

button\_tables.setText(table\_name); button\_tables.setBounds(x, y, width, height);

button\_tables.setFont(**new** Font("Tahoma",

Font.BOLD, 16));

button\_tables.setBackground(Color.LIGHT\_GRAY);

**if**(!table\_check(Integer.valueOf(button\_tables.getName())))

button\_tables.setBackground(Color.GREEN);;

set\_event(button\_tables);

list\_button\_tables

set\_size();

}

} **catch** (Exception e) {

.add(button\_tables);

// **TODO**: handle exception

e.printStackTrace();

}

}

❖ **Chức Năng Quản Lý Hóa Đơn:**

**public boolean** check\_info\_bill(String bills\_id, String drinks\_id) { **try** {

String sql = "SELECT \*\r\n" +

"FROM INFO\_BILLS\r\n" +

"WHERE BILLS\_Id = ? AND DRINKS\_Id =

?";

PreparedStatement cmd =

connectiveSQL.*cn*.prepareStatement(sql);

cmd.setString(1, bills\_id);

cmd.setString(2, drinks\_id);

ResultSet rs = cmd.executeQuery();

**if**(rs.next()) **return true**; } **catch**

(Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return false**;

}

❖ **Chức Năng Quản Lý Thực Đơn:**

**public void** add\_drink(String id, String name, String price) { **try** {

**float** drink\_price =Float.valueOf(price);

String sql = "INSERT INTO DRINKS (DRINKS\_Id,

DRINKS\_Name, DRINKS\_Price, DRINKS\_Status, TYPE\_DRINKS\_Id) \r\n" + "VALUES (?, ?, ?, 1, ?)";

PreparedStatement cmd =

connectiveSQL.*cn*.prepareStatement(sql);

cmd.setString(1, id); cmd.setString(2,

name); cmd.setFloat(3, drink\_price);

cmd.setString(4, type\_drinks\_id);

cmd.executeUpdate();

JOptionPane.showMessageDialog(**null**, "Thêm thức uống

thành công", "Thêm Thức

Uống",JOptionPane.OK\_CANCEL\_OPTION);

**return**;

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

JOptionPane.showMessageDialog(**null**, "Có lỗi xảy ra, thêm thức uống không thành công", "Thêm Thức

Uống",JOptionPane.OK\_CANCEL\_OPTION);

}

**public void** fix\_drink(String id, String name, String price) {

**try** {

**float** pr = Float.valueOf(price.trim());

String sql = "UPDATE DRINKS\r\n" +

"SET DRINKS\_Name = ?, DRINKS\_Price =

?\r\n" +

"WHERE DRINKS\_Id=?";

PreparedStatement cmd =

connectiveSQL

.*cn*.prepareStatement(sql);

cmd.setString(1, name); cmd.setFloat(2,

pr); cmd.setString(3, id);

cmd.executeUpdate();

JOptionPane.showMessageDialog(**null**, "Cập nhật dữ liệu thức uống thành công", "Cập nhập dữ liệu Thức Uống",JOptionPane.OK\_CANCEL\_OPTION);

**return**;

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

JOptionPane.showMessageDialog(**null**, "Có lỗi xảy ra, cập nhât uống không thành công", "Cập nhật dữ liệu thức

uống",JOptionPane.OK\_CANCEL\_OPTION);

}

**public void** delete\_drink(String id) {

**try** {

String sql = "DELETE FROM DRINKS WHERE

DRINKS\_Id = ?";

PreparedStatement cmd =

connectiveSQL.*cn*.prepareStatement(sql);

cmd.setString(1, id);

cmd.executeUpdate();

JOptionPane.showMessageDialog(**null**, "Xóa thức uống thành công", "Xóa Thức

Uống",JOptionPane.OK\_CANCEL\_OPTION);

**return**;

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

JOptionPane.showMessageDialog(**null**, "Có lỗi xảy ra, xóa thức uống không thành công", "Xóa Thức

Uống",JOptionPane.OK\_CANCEL\_OPTION);

}

❖ **Chức Năng Quản Lý Doanh Thu**

**public float** month\_revenue() {

**float** month = 0;

**try** {

connectiveSQL.*connective*();

String sql = "SELECT SUM(BILLS\_Price)as sum\r\n"

+

"FROM BILLS\r\n" +

"WHERE MONTH(BILLS\_TimeOut) =

MONTH(GETDATE()) AND YEAR(BILLS\_TimeOut) =

YEAR(GETDATE())";

PreparedStatement cmd =

connectiveSQL.*cn*.prepareStatement(sql);

ResultSet rs = cmd.executeQuery();

**if**(rs.next())

month = rs.getFloat("sum");

} **catch** (Exception e) {

// **TODO**: handle exception

e.printStackTrace();

}

**return** month;

}

❖ **Chức Năng Đăng Xuất:**

bt\_logout.addActionListener(**new** ActionListener() {

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

**int** n = JOptionPane.showConfirmDialog(**null**, "Bạn

có muốn đăng xuất khỏi tài khoản", "Đăng xuất khỏi tài khoản", JOptionPane.YES\_NO\_OPTION); **if**(n==0)

dispose();

}

});

**CHỨC NĂNG KIỂM THỬ**

### 1. Thêm đồ uống:

- Thêm đồ uống: Khi muốn thêm đồ uống mới vào menu (id, tên đồ uống, giá, loại đồ uống).

**\*Ràng buộc:**

- Mỗi id, tên đồ uống là duy nhất .

- Phải nhập đầy đủ thông tin đồ uống, gồm:

+ Id đồ uống (là 1 chuỗi ký tự khác rỗng, bắt đầu bằng DR + ba chữ số nguyên)

VD: DR001, DR002, ...

+ Tên đồ uống(là 1 chuỗi ký tự khác rỗng có độ dài tối đa <= 50 ký tự)

+ Price (Đơn vị VNĐ (Việt Nam đồng), là chuỗi ký tự số và khác rỗng)

- Đầu vào:Thông tin đồ uống muốn thêm(id, tên đồ uống, giá, loại đồ uống)

**-** Xử lý:

+ Kiểm tra thông tin đồ uống muốn thêm có hợp lệ không. Thông tin hợp lệ là: Id khác rỗng, bắt đầu bằng DR và 3 chữ số và duy nhất.

+ Kiểm tra tên đồ uống muốn thêm vào có hợp lệ không. Thông tin hợp lệ: Name phải khác rỗng, có độ dài tối đa <= 50 ký tự và duy nhất

+ Kiểm tra giá đồ uống được thêm có hợp lệ không. Thông tin hợp lệ: Price khác rỗng và phải là số nguyên.

- Đầu ra: Thông báo từ hệ thống, hệ thống hỏi xác nhận “Bạn muốn thêm thức uống với giá đã nhập hay không?”, nếu hợp lệ thì thông báo “Thêm thức uống thành công”. Ngược lại thông báo “Loại thức uống này đã có”.

**\*Chương trình :**

JButton bt\_add = new JButton("THÊM");

bt\_add.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String id = tf\_id.getText().toString().trim();

String name = tf\_drinks.getText().toString().trim();

String price = tf\_price.getText().toString().trim();

System.*out*.println(id);

if(id.equals("")) {

JOptionPane.*showMessageDialog*(null, "ID chưa được nhập", "Thêm Thức Uống",

JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

if(!id.matches("DR\\d{3}")) {

JOptionPane.*showMessageDialog*(null, "ID chưa đúng định dạng DR\_\_",

"Thêm Thức Uống", JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

if(name.equals("")) {

JOptionPane.*showMessageDialog*(null, "Tên thức uống chưa được nhập",

"Thêm Thức Uống", JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

if(price.equals("")) {

JOptionPane.*showMessageDialog*(null, "Giá bán thức uống chưa được nhập",

"Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

if(price.matches("\\d")){

JOptionPane.*showMessageDialog*(null, "Giá bán thức uống phải là số",

"Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

int i = JOptionPane.*showConfirmDialog*(null, "Bạn muốn thêm thức uống "+ name

+" với giá bán "+price, "Thêm Thức Uống", JOptionPane.*YES\_NO\_OPTION*);

if(i == 0) {

ArrayList<Drinks> temp = ds\_drinks.get(cb\_type\_drinks.getSelectedItem());

for (int j = 0; j < temp.size(); j++) {

System.*out*.println("id:" + id + ", " + temp.get(j).getId());

if(temp.get(j).getId().equals(id)) {

JOptionPane.*showMessageDialog*(null, "Loại Thức Uống Này Đã Có!", "Thêm Thức Uống", JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

}

ds\_drinks.get(cb\_type\_drinks.getSelectedItem()).add(new Drinks(id, name, price));

action\_menu.add\_drink(id, name, price);

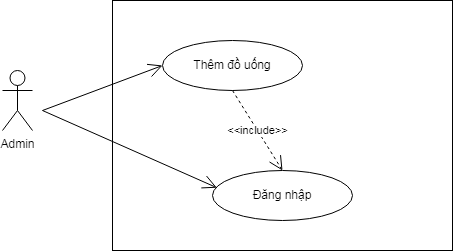
refresh\_table();

}

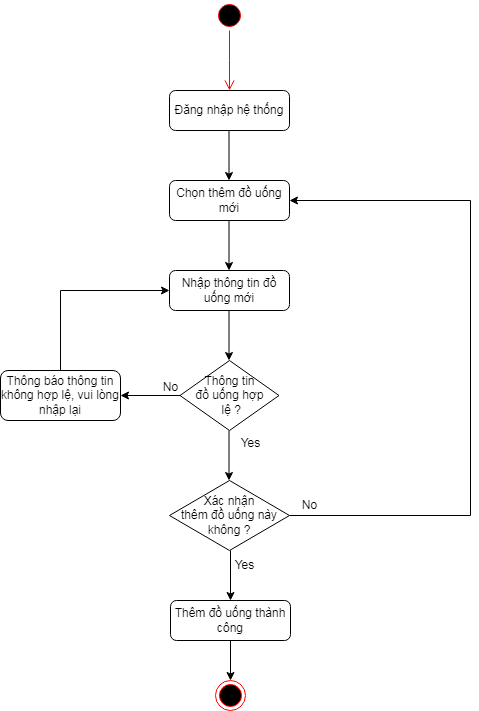
}

}

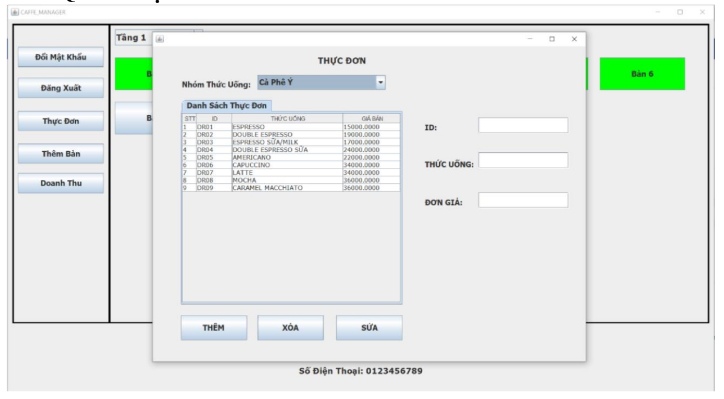
**\*Biểu đồ Usecase:**



**\*Biểu đồ Hoạt động:**

****

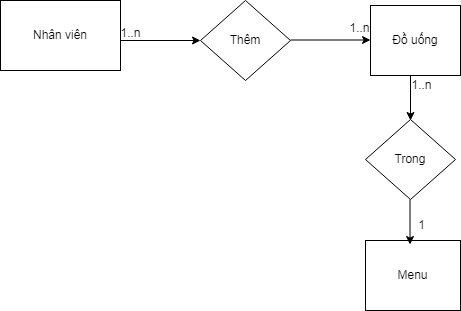
**\*Kịch bản và phác thảo giao diện:**



|  |  |  |  |
| --- | --- | --- | --- |
| Tên hành động | Mô tả | Thành công | Thất bại |

|  |  |  |  |
| --- | --- | --- | --- |
| Thêm  thức uống | Nhập đầy đủ các thông tin của thức uống sau đó nhấn vào button “thêm” | App sẽ thêm một đồ uống mới vào danh sách đồ uống | Gửi thông báo thông tin nhập không chính xác. |

**\*Mô hình thực thể mối quan hệ:**



### 2. Xóa đồ uống:

- Xóa đồ uống: Khi một đồ uống không còn được bán tại quán nữa thì sẽ tiến hành xóa đồ uống đó ra khỏi menu

**\*Ràng buộc:** + Id phải tồn tại trong hệ thống.

- Đầu vào: + Thông tin đồ uống muốn xóa (Id)

- Xử lý: + Kiểm tra thông tin đồ uống muốn xóa có hợp lệ không. Thông tin hợp lệ là: Id khác rỗng, bắt đầu bằng DR và 3 chữ số, Id tồn tại trong CSDL

- Đầu ra: + Thông báo từ hệ thống, nếu hợp lệ thì thông báo “Xóa thức uống thành công”. Ngược lại thông báo lỗi tương ứng.

**\*Chương trình**:

JButton bt\_delete = new JButton("XÓA");

bt\_delete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

String id = (tf\_id.getText().toString()).trim();

String name = (tf\_drinks.getText().toString()).trim();

if(id.equals("")) {

JOptionPane.*showMessageDialog*(null, "ID chưa nhập!",

"Xóa Thức Uống", JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

if(!id.matches("DR\\d{3}")) {

JOptionPane.*showMessageDialog*(null, "ID định dạng sai!",

"Xóa Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

int i = JOptionPane.*showConfirmDialog*(null, "Bạn muốn xóa thức uống

"+name, "Xóa Thức Uống", JOptionPane.*YES\_NO\_OPTION*);

if(i==0){ ArrayList<Drinks>temp=ds\_drinks.get(cb\_type\_drinks.getSelectedItem());

int k = 0;

for (k = 0; k < temp.size(); k++) {

System.*out*.println(id + ","+temp.get(k).getId()+","+k+", s:"+temp.size());

if(temp.get(k).getId().equals(id)) {

break;

}

}

if(k >= temp.size()) {

JOptionPane.*showMessageDialog*(null, "ID này không tồn tại!", "Xóa Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}else{ ds\_drinks.get(cb\_type\_drinks.getSelectedItem()).remove(getindex\_id(id, temp));

action\_menu.delete\_drink(id);

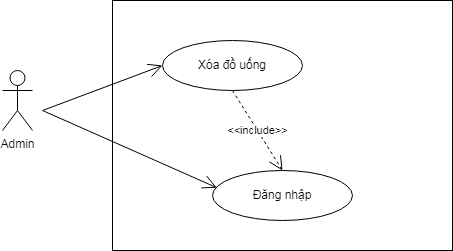
refresh\_table();

}

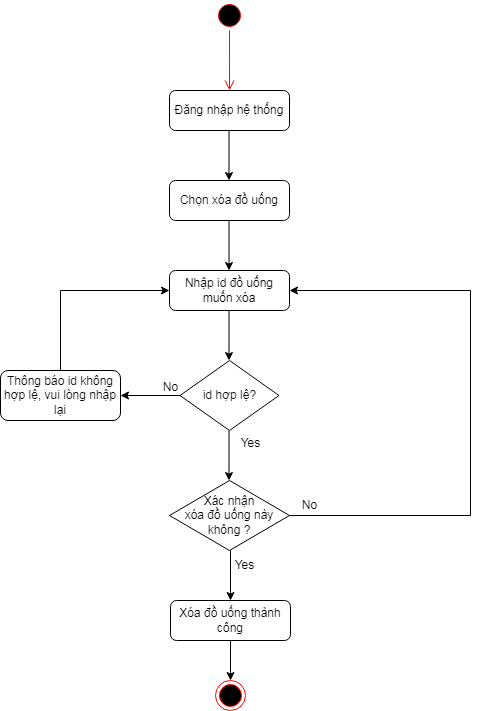
}

});

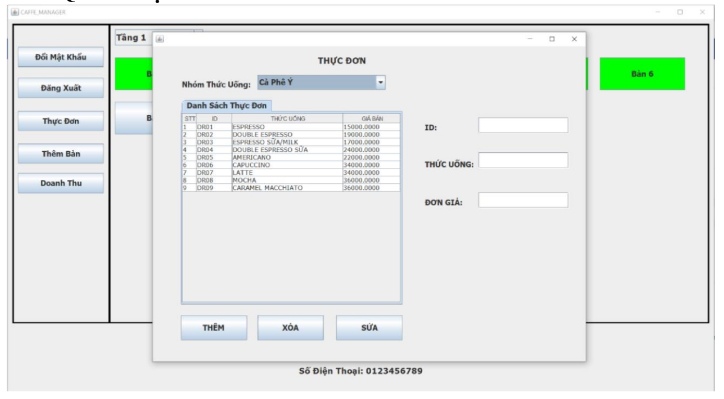
**\*Biểu đồ Usecase:**



**\*Biểu đồ Hoạt động:**



**\*Kịch bản và phác thảo giao diện:**



|  |  |  |  |
| --- | --- | --- | --- |
| Tên hành động | Mô tả | Thành công | Thất bại |

|  |  |  |  |
| --- | --- | --- | --- |
| Xóa thức uống | Chọn đồ uống  muốn xóa và nhân vào button “xóa” | App sẽ loại đồ uống đó ra khỏi danh sách đồ uống | Thông báo thông tin nhập bị sai |

**\*Mô hình thực thể mối quan hệ:**

