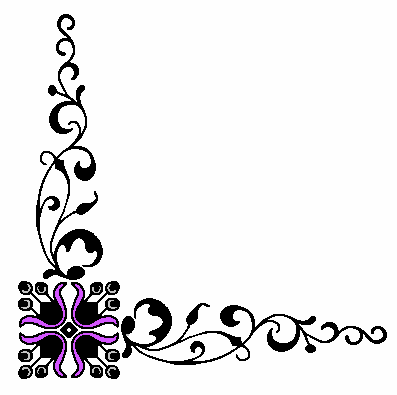
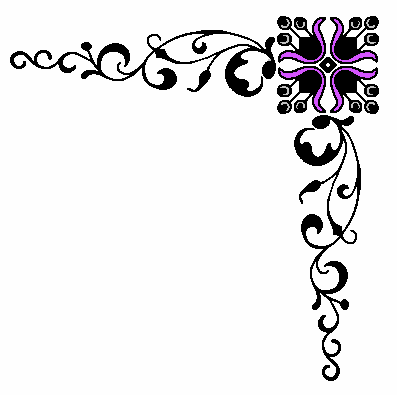
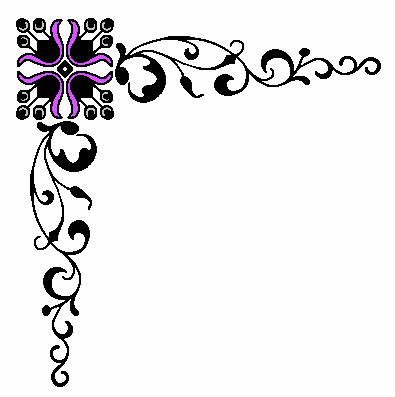
**Link driver các tuần của nhóm 5:**



TRƯỜNG ĐẠI HỌC KHOA HỌC

**KHOA CÔNG NGHỆ THÔNG TIN**

\*\*\*\*\*



**ĐỀ TÀI:**

**KIỂM ĐỊNH HỆ THỐNG QUẢN LÝ CAFE**

**Tên lớp học phần : Kiểm định phần mềm - Nhóm 4**

**Mã lớp học phần : 2023-2024.1.TIN4183.004**

**Giáo viên hướng dẫn : Lê Văn Tường Lân**

**Nhóm thực hiện : Nhóm 5**

**Sinh viên thực hiện :** Nguyễn Trung Hoành (20T1020385)

Lê Phước Thi (20T1020563)

Đoàn Quốc Huy (20T1020661)

Hoàng Công Huy (20T1020049)

Tôn Thất Hoàng Túc (20T1020123)

Trần Lộc Hải (20T1020025)

Nguyễn Thái Hoàng (20T1020037)

**MỤC LỤC**

[**I. QUẢN LÝ NHÓM:** 4](#_Toc144020455)

[**1. Bảng phân công, theo dõi và đánh giá cá nhân của nhóm:** 4](#_Toc144020456)

[1.1. Tuần 1: Chọn nhóm và thống nhất đề tài: 4](#_Toc144020457)

[1.2. Tuần 2: Kiểm thử hộp đen 5](#_Toc144020458)

[1.3. Tuần 3 Kiểm thử hộp trắng 6](#_Toc144020459)

[1.4. Tuần 4: Chỉnh sửa kiểm thử hộp trắng & kiểm thử hộp đen 6](#_Toc144020460)

[1.5. Tuần 5: Kiểm thử module 7](#_Toc144020461)

[**2. Bảng đánh giá cuối cùng theo thang điểm 10:** 8](#_Toc144020462)

[**II. NỘI DUNG:** 9](#_Toc144020463)

[**1. Thông tin cấu hình phần mềm:** 9](#_Toc144020464)

[1.1. Quản lý dự án: 9](#_Toc144020465)

[1.1.1. Thêm đồ uống: 9](#_Toc144020466)

[1.1.2. Xóa đồ uống: 10](#_Toc144020467)

[1.2. Lược đồ quan hệ: 10](#_Toc144020468)

[1.3. Chi tiết bảng: 11](#_Toc144020469)

[1.4. Hồ sơ thiết kế giao diện chức năng: 11](#_Toc144020470)

[1.5. Hồ sơ cài đặt: 13](#_Toc144020471)

[**2. Kiểm tra module:** 17](#_Toc144020472)

[2.1. Thêm đồ uống: 17](#_Toc144020473)

[2.2. Xóa đồ uống: 22](#_Toc144020474)

[**3. Kiểm tra bằng phương pháp hộp đen:** 26](#_Toc144020475)

[3.1. Chức năng thêm đồ uống: 26](#_Toc144020476)

[3.2. Chức năng xóa đồ uống: 33](#_Toc144020477)

[**4. Kiểm tra bằng phương pháp hộp trắng:** 37](#_Toc144020478)

[4.1. Chức năng thêm đồ uống: 37](#_Toc144020479)

[4.2. Chức năng xóa đồ uống: 42](#_Toc144020480)

# I. QUẢN LÝ NHÓM:

## 1. Bảng phân công, theo dõi và đánh giá cá nhân của nhóm:

### 1.1. Tuần 1 (03/07/2023 – 09/07/2023): Chọn nhóm và thống nhất đề tài:

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nhiệm vụ** | **Mức độ đóng góp** |
| 1 | Nguyễn Trung Hoành | Thống nhất hồ sơ dự án và chọn đề tài | **100%** |
| 2 | Lê Phước Thi | Thống nhất hồ sơ dự án và chọn đề tài | **100%** |
| 3 | Đoàn Quốc Huy | Thống nhất hồ sơ dự án và chọn đề tài | **100%** |
| 4 | Tôn Thất Hoàng Túc | Thống nhất hồ sơ dự án và chọn đề tài | **100%** |
| 5 | Hoàng Công Huy | Thống nhất hồ sơ dự án và chọn đề tài | **100%** |
| 6 | Trần Lộc Hải | Thống nhất hồ sơ dự án và chọn đề tài | **100%** |
| 7 | Nguyễn Thái Hoàng | Thống nhất hồ sơ dự án và chọn đề tài | **100%** |

### 1.2. Tuần 2(10/07/2023 – 16/07/2023): Kiểm thử hộp đen

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Họ và tên** | **Nhiệm vụ** | **Mức độ đóng góp** |
| 1 | Nguyễn Trung Hoành | Chỉnh sửa, hoàn chỉnh đề tài dự án | **100%** |
| 2 | Lê Phước Thi | Chỉnh sửa, hoàn chỉnh đề tài dự án | **100%** |
| 3 | Đoàn Quốc Huy | Chỉnh sửa, hoàn chỉnh đề tài dự án | **100%** |
| 4 | Tôn Thất Hoàng Túc | Chỉnh sửa, hoàn chỉnh đề tài dự án | **100%** |
| 5 | Hoàng Công Huy | Chỉnh sửa, hoàn chỉnh đề tài dự án | **100%** |
| 6 | Trần Lộc Hải | Chỉnh sửa, hoàn chỉnh đề tài dự án | **100%** |
| 7 | Nguyễn Thái Hoàng | Chỉnh sửa, hoàn chỉnh đề tài dự án | **100%** |

### 1.3. Tuần 3(17/07/2023 – 23/07/2023): Kiểm thử hộp trắng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nhiệm vụ** | **Mức độ đóng góp** |
| 1 | Nguyễn Trung Hoành | Kiểm thử hộp đen chức năng thêm, xóa đồ uống | **100%** |
| 2 | Lê Phước Thi | Kiểm thử hộp đen chức năng thêm, xóa đồ uống | **100%** |
| 3 | Đoàn Quốc Huy | Kiểm thử hộp đen chức năng thêm, xóa đồ uống | **100%** |
| 4 | Tôn Thất Hoàng Túc | Kiểm thử hộp đen chức năng thêm, xóa đồ uống | **100%** |
| 5 | Hoàng Công Huy | Kiểm thử hộp đen chức năng thêm, xóa đồ uống | **100%** |
| 6 | Trần Lộc Hải | Kiểm thử hộp đen chức năng thêm, xóa đồ uống | **100%** |
| 7 | Nguyễn Thái Hoàng | Kiểm thử hộp đen chức năng thêm, xóa đồ uống | **90%** |

### 1.4. Tuần 4(24/07/2023 – 30/07/2023): Chỉnh sửa kiểm thử hộp trắng & kiểm thử hộp đen

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nhiệm vụ** | **Mức độ đóng góp** |
| 1 | Nguyễn Trung Hoành | Kiểm thử hộp trắng chức năng thêm, xóa đồ uống | **100%** |
| 2 | Lê Phước Thi | Kiểm thử hộp trắng chức năng thêm, xóa đồ uống | **100%** |
| 3 | Đoàn Quốc Huy | Kiểm thử hộp trắng chức năng thêm, xóa đồ uống | **100%** |
| 4 | Tôn Thất Hoàng Túc | Kiểm thử hộp trắng chức năng thêm, xóa đồ uống | **100%** |
| 5 | Hoàng Công Huy | Kiểm thử hộp trắng chức năng thêm, xóa đồ uống | **100%** |
| 6 | Trần Lộc Hải | Kiểm thử hộp trắng chức năng thêm, xóa đồ uống | **100%** |
| 7 | Nguyễn Thái Hoàng | Kiểm thử hộp trắng chức năng thêm, xóa đồ uống | **100%** |

### 1.5. Tuần 5(31/07/2023 – 06/08/2023): Kiểm thử module

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **Nhiệm vụ** | **Mức độ đóng góp** |
| 1 | Nguyễn Trung Hoành | Kiểm thử modulechức năng thêm, xóa đồ uống | **100%** |
| 2 | Lê Phước Thi | Kiểm thử modulechức năng thêm, xóa đồ uống | **100%** |
| 3 | Đoàn Quốc Huy | Kiểm thử modulechức năng thêm, xóa đồ uống | **100%** |
| 4 | Tôn Thất Hoàng Túc | Kiểm thử modulechức năng thêm, xóa đồ uống | **90%** |
| 5 | Hoàng Công Huy | Kiểm thử modulechức năng thêm, xóa đồ uống | **100%** |
| 6 | Trần Lộc Hải | Kiểm thử modulechức năng thêm, xóa đồ uống | **100%** |
| 7 | Nguyễn Thái Hoàng | Kiểm thử modulechức năng thêm, xóa đồ uống | **100%** |

## 2. Bảng đánh giá cuối cùng theo thang điểm 10:

|  |  |
| --- | --- |
| **Tên Thành Viên** | **Điểm đánh giá** |
| Nguyễn Trung Hoành | 9.5 |
| Lê Phước Thi | 10 |
| Đoàn Quốc Huy | 9.5 |
| Tôn Thất Hoàng Túc | 8 |
| Trần Lộc Hải | 9.5 |
| Nguyễn Thái Hoàng | 8 |
| Hoàng Công Huy | 9.5 |

# II. NỘI DUNG:

## 1. Thông tin cấu hình phần mềm:

### 1.1. Quản lý dự án:

Quản lý menu: Gồm danh 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 riêng: Thêm, 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 tại quán nữa thì sẽ tiến hành xóa đồ uống đó ra khỏi menu.

⮚ Ràng buộc:

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

- Đơn giá phải được tính bằng VNĐ.

### 1.1.1. 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”.

### 1.1.2. 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

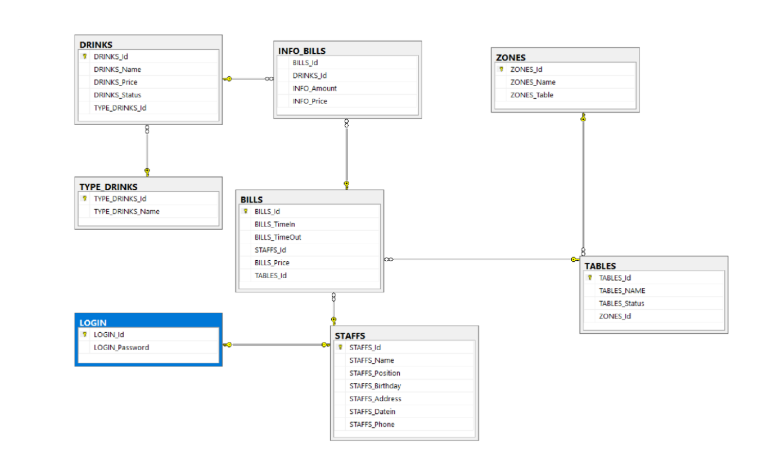
**\*Ràng buộc:** + Id phải tồn tại trong hệ thống.

- 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

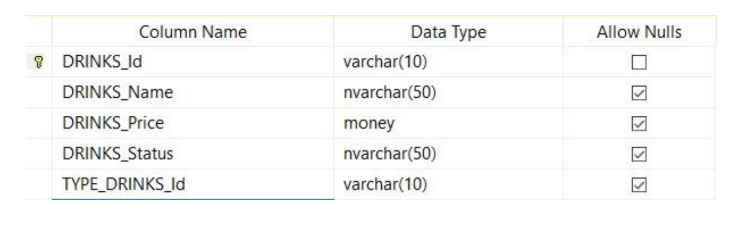
- Đầu ra: + Thông báo từ hệ thống, nếu hợp lệ thì thông báo “Xóa thức uống thành công”. Ngược lại thông báo lỗi tương ứng.

### 1.2. Lược đồ quan hệ:



### 1.3. Chi tiết bảng:

❖ **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

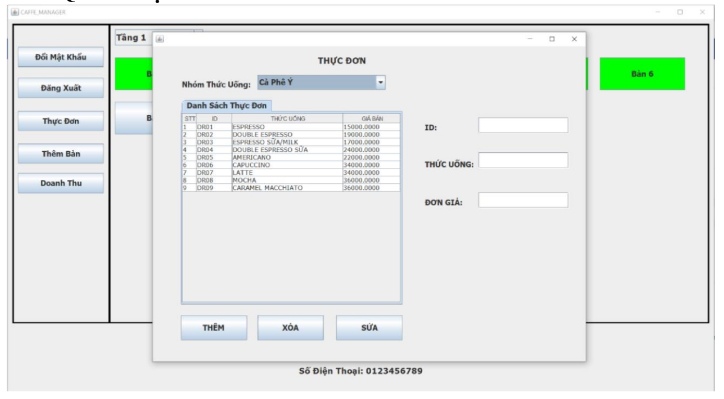
+ DRINKS\_Price: Chứa giá của các loại đồ uống

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

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

### 1.4. Hồ sơ thiết kế giao diện chức năng:

❖ **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 nhấn button “sửa” | App sẽ update lại thông tin đồ uống vừa mới chỉnh sửa |  |

### 1.5. Hồ sơ cài đặt:

**1.5.1. Tổng quan**

Ngôn ngữ lập trình: Java

Phần mềm lập trình: Eclipse, Microsoft SQL server

**1.5.2. Cài đặt chương trình**

❖ **Chức năng thêm đồ uống:**

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();

}

}

}

❖ **Chức năng xóa đồ uống:**

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();

}

}

});

## 2. Kiểm tra module:

### 2.1. Thêm đồ uống:

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

bt\_add.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

{A: id, name, price ∈ String , temp ∈ ArrayList<Drinks>}

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);

{E1}

if(id.equals("")) {

JOptionPane.*showMessageDialog*(null, "ID chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

{E2}

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

JOptionPane.*showMessageDialog*(null, "ID chưa đúng định dạng DR ", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

{E3}

        if(name.equals("")) {

      JOptionPane.*showMessageDialog*(null, "Tên thức uống chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;}

{E4}

if(price.equals("")) {

JOptionPane.*showMessageDialog*(null, "Giá bán thức uống chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}

{E5}

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*);

{E6}

if(i == 0) {

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

{C}

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

//Q

{C, j < temp.size()}

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

    {E7}

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

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

return;}

{C} } {C, j > temp.size()}

//Q1

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

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

refresh\_table();

}

}

});

{B: id, name, price ∈ String, temp ∈ ArrayList<Drinks>, add\_drink(id, name, price) }

**2.1.1. Mô tả mệnh đề Vào/Ra của chức năng thêm đồ uống:**

- Đầu vào:

{A: id, name, price ∈ String , temp ∈ ArrayList<Drinks>}

- Đầu ra:

{B: id, name, price ∈ String, temp ∈ ArrayList<Drinks>, add\_drink(id, name, price) }

**2.1.2. Mô tả mệnh đề bất biến:**

{C: id, name, price ∈ String, temp ∈ ArrayList<Drinks>, j, temp.size() ∈ N,

j < temp.size()}

**2.1.3. Chứng minh:**

- Gọi mệnh đề thể hiện tính chất dữ liệu vào là {A}

{A: id, name, price ∈ String , temp ∈ ArrayList<Drinks>}

- Gọi mệnh đề thể hiện tính chất dữ liệu ra là {B}

{B: id, name, price ∈ String, temp ∈ ArrayList<Drinks>, add\_drink(id, name, price) }

- Theo tính chất của phép gán ta có:

* {A1}={A[id| tf\_id.getText().toString().trim()]}
* {A2}={A1[name| tf\_drinks.getText().toString().trim()]}
* {A3}={A2[price|tf\_price.getText().toString().trim()]}

\*Kiểm tra: {A3} if(id.equals("")) {

JOptionPane.*showMessageDialog*(null, "ID chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}, ta có:

i. {A3,E1}JOptionPane.*showMessageDialog*(null, "ID chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*); return;{B}

ii. {A3,!E1}=>(Logic) {A3}

\*Kiểm Tra:{A3}if(!id.matches("DR\\d{3}")) {

JOptionPane.*showMessageDialog*(null, "ID chưa đúng định dạng DR ", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}, ta có:

i. {A3,E2}JOptionPane.*showMessageDialog*(null, "ID chưa đúng định dạng DR ", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);return;

{B}

ii. {A3,!E2}=>(Logic) {A3}

\*Kiểm tra: {A3} if(name.equals("")) {

      JOptionPane.*showMessageDialog*(null, "Tên thức uống chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}, ta có:

i. {A3,E3} JOptionPane.*showMessageDialog*(null, "Tên thức uống chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);{B}

ii. {A3,!E3}=>(Logic) {A3}

\*Kiểm tra: {A3}if(price.equals("")) {

JOptionPane.*showMessageDialog*(null, "Giá bán thức uống chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}, ta có:

i. {A3,E4}JOptionPane.*showMessageDialog*(null, "Giá bán thức uống chưa được nhập", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

{B}

ii. {A3,!E4}=>(Logic) {A3}

\*Kiểm tra: {A3}if(price.matches("\\d")){

JOptionPane.*showMessageDialog*(null, "Giá bán thức uống phải là số", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);

return;

}, ta có:

i. {A3,E5}JOptionPane.*showMessageDialog*(null, "Giá bán thức uống phải là số", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);{B}

ii. {A3, !E5} =>(Logic) {A3}

- Theo phép gán ta có:

{A4}={A3 [i|JOptionPane.*showConfirmDialog*(null, "Bạn muốn thêm thức uống "+ name +" với giá bán "+price, "Thêm Thức Uống", JOptionPane.*YES\_NO\_OPTION*)]}

\*Kiểm Tra: {A4,!E6}=>(Logic){B}

\*Kiểm Tra:{A4,E6}ArrayList<Drinks> temp = ds\_drinks.get(cb\_type\_drinks.getSelectedItem());{A5}

- Dự trù {C} là mệnh đề bất biến đoạn đoạn trình Q với điều kiện (j < temp.size())

{C: id, name, price ∈ String, temp ∈ ArrayList<Drinks>, j, temp.size() ∈ N,

j < temp.size()}

- Để chứng minh {C} là bất biến, ta cần chứng minh {C,(j < temp.size()} Q {C}

ta có:

{A5}j=0{C}

{C,j<temp.size()}={K}

\*Kiểm tra: {K} if(temp.get(j).getId().equals(id)) {

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

return;}, ta có:

i. {K,E7} JOptionPane.*showMessageDialog*(null, "Loại Thức Uống Này Đã Có!", "Thêm Thức Uống",JOptionPane.*OK\_CANCEL\_OPTION*);{B}

ii. {K, !E7} =>(Logic) {C}

- Do đó mệnh đề {C}là bất biến của đoạn trình Q, Kết thúc ta có mệnh đề: {C,j>=temp.size()}

{C,j>=temp.size()}Q1{B}

Từ đó ta suy ra {A}P{B} đoạn trình trên là đúng

### 2.2. Xóa đồ uống:

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

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

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

{A: id ∈ String , temp ∈ ArrayList<Drinks>}

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

//{E1}

**if**(id.equals("")) {

JOptionPane.*showMessageDialog*(**null**, "ID chưa nhập!", "Xóa Thức Uống",JOptionPane.***OK\_CANCEL\_OPTION***);

**return**;

}

//{E2}

**if**(!id.matches("DR\\d{2}")) {

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”, "Xóa Thức Uống", JOptionPane.***YES\_NO\_OPTION)***

//{E3}

**if**(i==0) {

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

**int** k = 0;

{C}

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

//Q

{C, k < temp.size()}

{E4}

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

**break**;

}

{C}

} {C, k > temp.size()}

{E5}

**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));

//P

action\_menu.delete\_drink(id);

refresh\_table();

}

}

}

});

{B: id ∈ String, temp ∈ ArrayList<Drinks>, delete\_drink(id) }

**2.2.1. Mô tả mệnh đề Vào/Ra của chức năng thêm đồ uống:**

- Đầu vào:

{A: id ∈ String , temp ∈ ArrayList<Drinks>}

- Đầu ra:

{B: id ∈ String, temp ∈ ArrayList<Drinks>, delete\_drink(id) }

**2.2.2 Mô tả mệnh đề bất biến:**

{C: id ∈ String, temp ∈ ArrayList<Drinks>, k, temp.size() ∈ N, k < temp.size()}

**2.2.3. Chứng minh:**

- Gọi mệnh đề thể hiện tính chất dữ liệu vào là {A}

{A: id ∈ String , temp ∈ ArrayList<Drinks>}

- Gọi mệnh đề thể hiện tính chất dữ liệu ra là {B}

{B: id ∈ String, temp ∈ ArrayList<Drinks>, delete\_drink(id) }

- Theo tính chất phép gán:

{A1}={A[id| tf\_id.getText().toString().trim()]}

\*Kiểm tra: {A1} **if**(id.equals("")) {

JOptionPane.*showMessageDialog*(**null**, "ID chưa nhập!", "Xóa Thức Uống",JOptionPane.***OK\_CANCEL\_OPTION***);

**return**;

}, ta có:

i. {A1,E1}JOptionPane.*showMessageDialog*(**null**, "ID chưa nhập!", "Xóa Thức Uống",JOptionPane.***OK\_CANCEL\_OPTION***);{B}

ii. {A1, !E2} =>(Logic) {A1}

\*Kiểm tra: {A1} **if**(!id.matches("DR\\d{2}")) {

JOptionPane.*showMessageDialog*(**null**, "ID định dạng sai!", "Xóa Thức Uống",JOptionPane.***OK\_CANCEL\_OPTION***);

**return**;

}, ta có:

i. {A1,E2}JOptionPane.*showMessageDialog*(**null**, "ID định dạng sai!", "Xóa Thức Uống",JOptionPane.***OK\_CANCEL\_OPTION***);{B}

ii. {A1, !E2} =>(Logic) {A1}

{A2}={A1 [i|JOptionPane.*showConfirmDialog*(**null**, "Bạn muốn xóa thức uống”, "Xóa Thức Uống", JOptionPane.***YES\_NO\_OPTION)***]}

\*Kiểm Tra:{A2,!E3}=>(Logic){B}

\*Kiểm Tra: {A2,E3}temp=ds\_drinks.get(cb\_type\_drinks.getSelectedItem()){A3}

- Dự trù mệnh đề {C} là mệnh đề bất biến của đoạn trình Q với điều kiện

(k < temp.size())

{C: id ∈ String, temp ∈ ArrayList<Drinks>, k, temp.size() ∈ N, k < temp.size()}

- Để chứng minh {C} là bất biến, ta cần chứng minh {C,( k < temp.size()} Q {C}, ta có:

{A3}k=0{C}

{C,k<temp.size()}={K}

\*Kiểm tra: {K} **if**(temp.get(k).getId().equals(id)) {

**break**;

} {C}, ta có:

i. {K,E4}break;{C}

ii. {K, !E4} =>(Logic) {C}

- Do đó {C} là mệnh đề bất biến của đoạn trình Q, kết thúc ta có {C,k>=temp.size()}

\*Kiểm tra:{C,k>=temp.size(),E5}

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

**return**; {B}

\*Kiểm tra:{C,k>=temp.size(),!E5}P{B}

Từ đó suy ra: đoạn trình trên là đúng

## 3. Kiểm tra bằng phương pháp hộp đen:

### 3.1. Chức năng thêm đồ uống:

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

- 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.

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

- 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”.

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

**Phương pháp phân tích lớp tương đương:**

|  |  |  |
| --- | --- | --- |
| **Các giá trị đầu vào** | **Các lớp tương đương hợp lệ** | **Các lớp tương đương không hợp lệ** |
| Id đồ uống | Id đồ uống là chuỗi ký tự bắt đầu bằng DR và 3 chữ số. (1)    Id đồ uống là chuỗi ký tự khác rỗng. (4)    Id đồ uống không được trùng nhau. (7) | Id đồ uống không bắt đầu bằng DR. (2)    Id đồ uống là bắt đầu bằng DR và không phải 3 chữ số. (3)    Id đồ uống là chuỗi các giá trị > 5 ký tự (5) hoặc rỗng. (6)    Id đồ uống trùng nhau giữa 2 sản phẩm khác nhau. (8) |
| Tên đồ uống | Tên đồ uống là 1 chuỗi không trùng với tên đồ uống đã có. (9)    Tên đồ uống có độ dài thuộc [1,50]. (11) | Tên đồ uống là chuỗi trùng . (10)    Tên đồ uống có độ dài  > 50 (12) hoặc rỗng. (13) |
| Giá | Giá là một chuỗi số nguyên dương. (14) | Giá nhập vào không phải là số nguyên dương. (15) |

**Các ca kiểm thử:**

+ Các ca kiểm thử hợp lệ (1),(4),(7),(9),(11),(14)

Id:DR099

Tên đồ uống: Nước ép mía

Giá : 7000000

+ Các ca kiểm thử không hợp lệ

(2) CC051

(3) DRBR2

(5) DR-NM03

(6) " "

(8)

DR016 CAFE1 19000.0000

DR016 CAFE2 29000.0000

(10)

DR016 CAFE1 19000.0000

DR099 CAFE1 29000.0000

(12) Nước ép mía siêu ngon siêu cấp siêu pro siêu pro max

(13) ""

(15) mười chín ngàn đồng

**Phân tích giá trị biên:**

|  |  |
| --- | --- |
| **Giá trị đầu vào kiểm thử** | **Kết quả mong muốn** |
| ID đồ uống là chuỗi có đúng 5 ký tự theo định dạng (Vd: DR007), Tên đồ uống có đúng 50 ký tự, giá là một chuỗi số nguyên dương | Lưu các thông tin vào cơ sở dữ liệu.  Hiển thị thông báo : Thêm đồ uống thành công |
| ID đồ uống là chuỗi lớn hơn 5 ký tự, Tên đồ uống có đúng 50 ký tự, giá là một chuỗi số nguyên dương | Hiển thị thông báo : Nhập sai định dạng |
| ID đồ uống là chuỗi có đúng 5 ký tự theo định dạng (Vd: DR007), Tên đồ uống lớn hơn 50 ký tự, giá là một chuỗi số nguyên dương | Hiển thị thông báo : Sai định dạng |
| ID đồ uống là chuỗi có đúng 5 ký tự theo định dạng (Vd: DR007), Tên đồ uống có đúng 1 ký tự, giá là một chuỗi số nguyên dương | Lưu các thông tin vào cơ sở dữ liệu.  Hiển thị thông báo : Thêm đồ uống thành công |
| ID đồ uống là chuỗi có đúng 5 ký tự theo định dạng (Vd: DR007), Tên đồ uống lớn hơn 50 ký tự, giá không phải là một chuỗi số nguyên dương | Hiển thị thông báo : Sai định dạng |
| Nhập vào chuỗi ID hoặc tên đồ uống hoặc giá là 1 chuỗi rỗng. | Hiển thị thông báo : cần nhập đầy đủ thông tin |

**Đồ thị nguyên nhân – kết quả:**

- Nguyên nhân:

+ Id đồ uống là chuỗi ký tự bắt đầu bằng DR và 3 chữ số. Tên đồ uống có độ dài nhỏ hơn 50 ký tự và Giá là một chuỗi số nguyên dương. (1)

+ Id đồ uống, giá và tên đồ uống là chuỗi khác rỗng.(2)

+ Id đồ uống không được trùng nhau và tên đồ uống là 1 chuỗi không trùng với tên đồ uống đã có. (3)

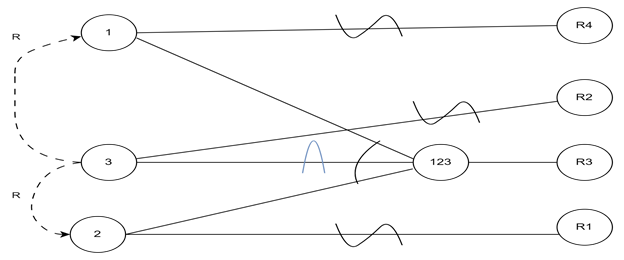
- Kết quả:

+ R1: Thông báo bạn cần nhập đầy đủ thông tin bắt buộc.

+ R2: Thông tin này đã tồn tại, thêm không thành công.

+ R3: Thông tin nhập vào sẽ được lưu vào cơ sở dữ liệu và hiển thị thông báo: Thêm thành công.

+ R4: Thông tin nhập vào không hợp lệ, và hiển thị thông báo: Thêm không thành công.



**Bảng quyết định:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **1** | **2** | **3** | **4** |
| **1** | 0 |  |  | 1 |
| **2** |  | 0 |  | 1 |
| **3** |  |  | 0 | 1 |
| **R1** | 0 | 1 | 0 | 0 |
| **R2** | 0 | 0 | 1 | 0 |
| **R3** | 0 | 0 | 0 | 1 |
| **R4** | 1 | 0 | 0 | 0 |

**Các trường hợp kiểm thử theo phương pháp đồ thị nguyên nhân- kết quả:**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Các điều kiện | Ca kiểm thử | Hành động |
| 1 | ID là chuỗi ký tự khác với yêu cầu “DR + 3 chữ số”.  Hoặc Giá không phải là một số nguyên dương.  Hoặc tên đồ uống có độ dài lớn hơn 50 ký tự | ID = “DR7”  Hoặc Giá = “-11500,05”  Hoặc Tên = “ Nước ép mía siêu ngon siêu cấp siêu pro siêu pro max” | R4 |
| 2 | ID là chuỗi ký tự có độ dài bằng rỗng  Hoặc Tên là chuỗi ký tự có độ dài bằng rỗng  Hoặc Giá là giá trị rỗng. | ID = “”  Hoặc Tên = ””  Hoặc Giá = “” | R1 |
| 3 | ID là chuỗi ký tự bao gồm ký tự DR và ba chữ số, chưa tồn tại trong hệ thống. Và Tên là chuỗi tự khác rỗng, chưa tồn tại trong hệ thống và Giá là một số nguyên dương. | ID = “DR909”  Và Tên = “CÀ PHÊ MUỐI”  Và Giá = “4000” | R3 |
| 4 | ID hoặc tên là chuỗi ký tự đã tồn tại trong hệ thống. | ID = DR01 đã tồn tại.  Hoặc Tên = LATTE đã tồn tại. | R2 |

### 3.2. Chức năng 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

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

- Xử lý:

- Kiểm tra thông tin đồ uống muốn sửa 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 chỉnh sửa có hợp lệ thông. Thông tin hợp lệ: Name phải khác rỗng

- Kiểm tra giá đồ uống chỉnh sửa có hợp lệ không. Thông tin hợp lệ: Price khác rỗng và phải là số

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

- Ràng buộc : Mã của mỗi đồ uống trong menu sẻ riêng biệt nhau

**Phương pháp phân tích lớp tương đương**

|  |  |  |
| --- | --- | --- |
| **Các giá trị đầu vào** | **Lớp tương đương hợp lệ** | **Lớp tương đương không hợp lệ** |
| Ký tự nhập vào | -ID là chuỗi ký tự bắt đầu bằng DR và 3 chữ số (1)                -ID đã có trong hệ thống (2) | -Chuỗi ID có ít hơn 5 ký tự (3)  -Chuỗi ID có nhiều hơn 5 ký tự(4)  -Chuỗi ID bắt đầu bằng ký tự khác DR (5)  -Chuỗi ID có 3 ký tự sau là khác chữ số (6)    -Chuỗi ID chưa tồn tại trong hệ thống (7) |

\*Các ca kiểm thử:

Ta có dữ liệu trên hệ thống: tableDRINKS = {(DR001, ESPRESSO, 15000), (DR002, DOUBLE ESPRESSO, 19000)}

Các ca kiểm thử hợp lệ là các ca kiểm thử bao phủ các lớp tương đương hợp lệ: (1), (2) => Drink\_type (DR011)

Các ca kiểm thử không hợp lệ:

(3) Drink\_type (CF11)

(4)Drink\_type (DR1127)

(5)Drink\_type(AB123)

(6) Drink\_type (DR99A)

(7) Drink\_type (DR999)

**Phân tích giá trị biên:**

|  |  |
| --- | --- |
| Giá trị đầu vào kiểm thử | Kết quả mong muốn |
| ID đồ uống là chuỗi có đúng 5 ký tự theo định dạng | Lưu các thông tin vào cơ sở dữ liệu.  Hiển thị thông báo : Xóa đồ uống thành công |
| ID đồ uống là chuỗi có lớn 5 ký tự theo định dạng | Hiển thị thông báo : Id đồ uống sai định dạng |
| ID đồ uống rỗng | Hiển thị thông báo :Chưa nhập id đồ uống |

**Đồ thị nguyên nhân –kết quả.**

- **Nguyên nhân:**

1. Thông tin “ID” nhập vào là chuỗi ký tự khác rỗng.

2. Thông tin “ID” nhập vào là chuỗi bắt đầu bằng DR và 3 chữ số.

3. Thông tin “ID” nhập vào đã tồn tại trong hệ thống.

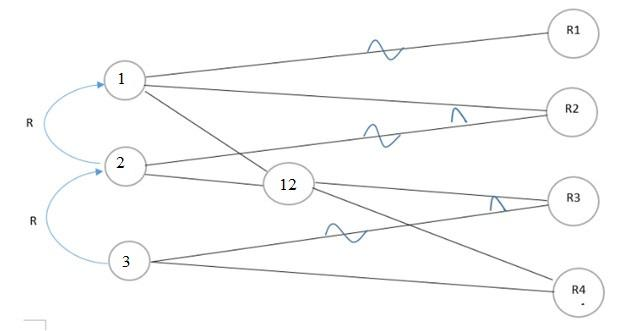
- **Kết quả:**

R1. Thông báo “ID không được để trống”.

R2. Thông báo “ID phải bắt đầu DR và 3 chữ số”.

R3. Thông báo “ID này không tồn tại”.

R4. Thông báo “Xóa thức uống thành công”.



**Bảng quyết định:**

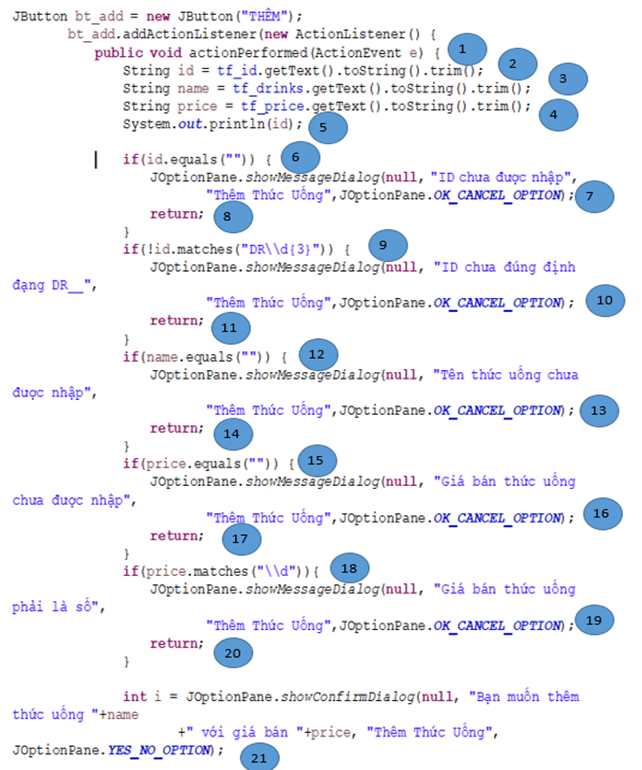
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 |
| 1 | 0 | 0 | 0 | 1 |
| 2 |  | 0 |  | 1 |
| 3 |  |  | 0 | 1 |
| R1 | 1 | 0 | 0 | 0 |
| R2 | 0 | 1 | 0 | 0 |
| R3 | 0 | 0 | 0 | 1 |
| R4 | 0 | 0 | 1 | 0 |

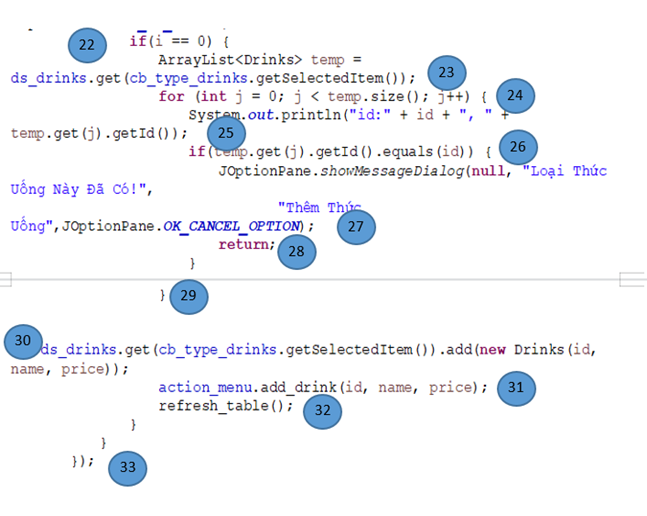
**Các trường hợp kiểm thử theo phương pháp đồ thị nguyên nhân- kết quả:**

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Các điều kiện | Ca kiểm thử | Hành động |
| 1 | ID là chuỗi ký tự có độ dài bằng rỗng. | ID = “” | R1 |
| 2 | ID là chuỗi ký tự không tồn tại trong hệ thống. | ID = DR111 không tồn tại. | R2 |
| 3 | ID là chuỗi ký tự bao gồm ký tự DR và ba chữ số, đã tồn tại. | ID = “DR003” | R3 |
| 4 | ID là chuỗi ký tự khác với yêu cầu “DR + 3 chữ số”. | ID = “DR7” | R4 |

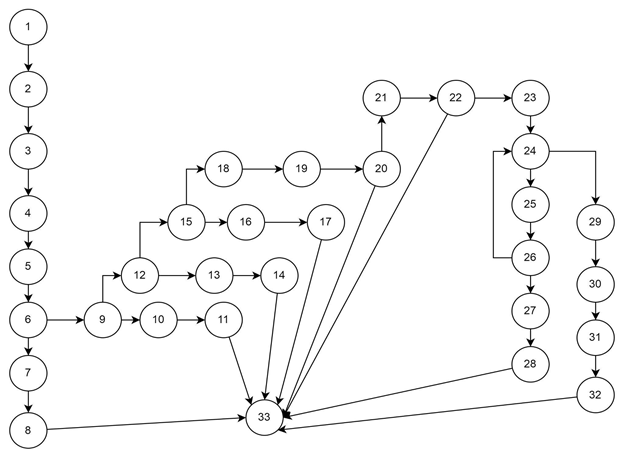
## 4. Kiểm tra bằng phương pháp hộp trắng:

### 4.1. Chức năng thêm đồ uống:

****

****

**4.1.1. Đồ thị thể hiện đường diễn tiến của chương trình:**

****

**4.1.2. Các đường kiểm thử:**

Xác định số đường kiểm thử: V(G) = 40 - 33 + 2 = 9

1. 1.2.3.4.5.6.7.8.33

2. 1.2.3.4.5.6.9.10.11.33

3. 1.2.3.4.5.6.9.12.13.14.33

4. 1.2.3.4.5.6.9.12.15.16.17.33

5. 1.2.3.4.5.6.9.12.15.18.19.20.33

6. 1.2.3.4.5.6.9.12.15.18.19.20.21.22.33

7. 1.2.3.4.5.6.9.12.15.18.19.20.21.22.23.24.29.30.31.32.33

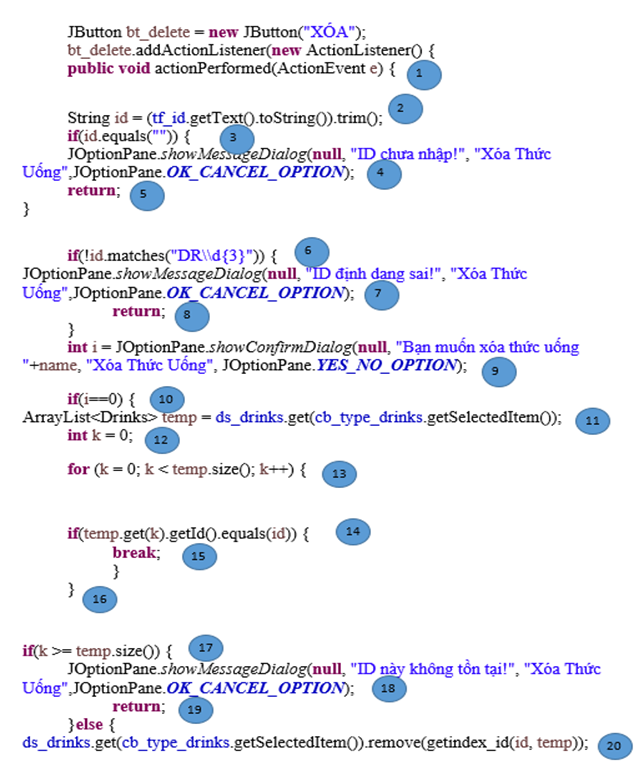
8. 1.2.3.4.5.6.9.12.15.18.19.20.21.22.23.24.25.26.24….27.28.33

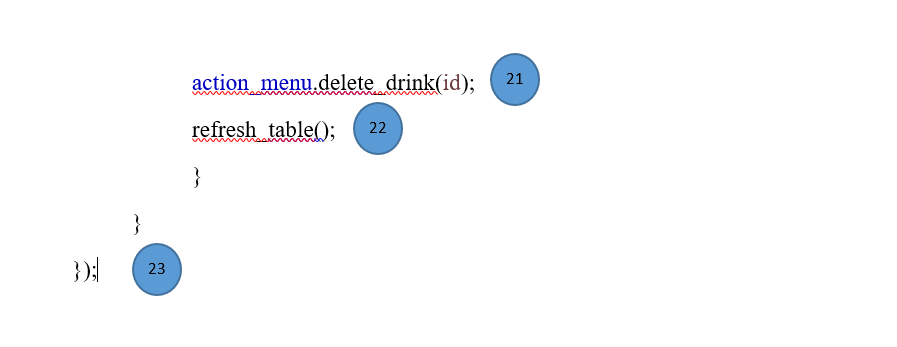
9. 1.2.3.4.5.6.9.12.15.18.19.20.21.22.23.24.25.26.24…29.30.31.32.33

**4.1.3. Xác định các testcase:**

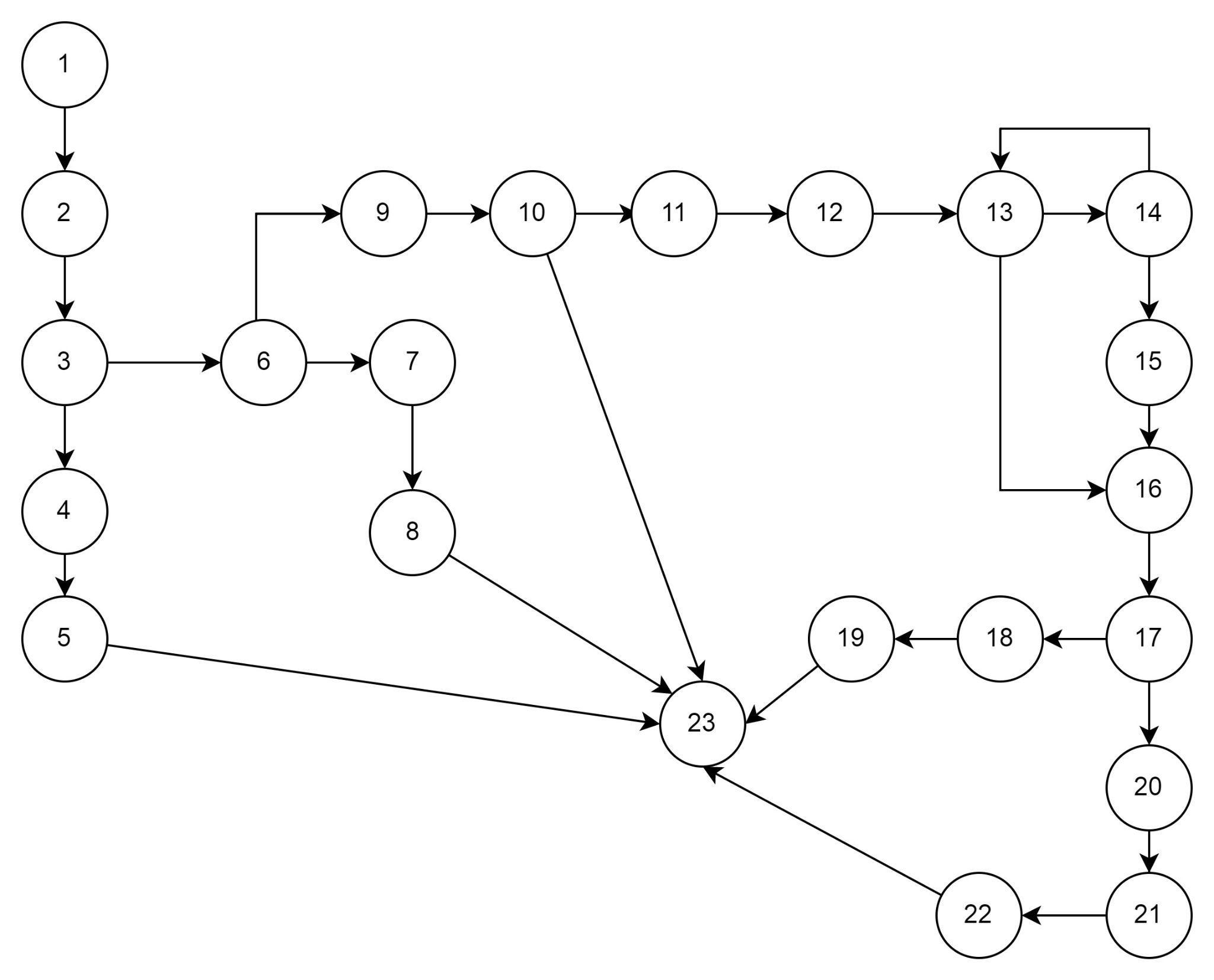
|  |  |  |
| --- | --- | --- |
| Đường kiểm thử | Giá trị đầu vào | Kết quả mong đợi |
| 1 | - Id là một chuỗi rỗng. | Hiển thị thông báo: Id chưa nhập! |
| 2 | - Id sai định dạng | Hiển thị thông báo:Id chưa nhập |
| 3 | - Tên thức uống là 1 chuỗi rỗng | Hiển thị thông báo: Tên đồ uống chưa được nhập. |
| 4 | - Giá thức uống là 1 chuỗi rỗng | Hiển thị thông báo : Giá bán đồ uống chưa được nhập. |
| 5 | - Giá đồ uống không phải là chuỗi ký tự số | Hiển thị thông báo: Giá bán thức uống phải là số. |
| 6 | - Từ chối thêm thức uống | Hủy thêm thức uống thành công. |
| 7 | -Thông tin nhập vào đầy đủ  -Danh sách đồ uống rỗng | Hiển thị thông báo:Thêm đồ uống thành công.  Lưu đồ uống vào cơ sở dữ liệu. |
| 8 | -Thông tin nhập vào đầy đủ.  - ID đồ uống đã tồn tại trong cơ sở dữ liệu. | Hiển thị thông báo:Loại đồ uống này đã có. |
| 9 | -Thông tin nhập vào đầy đủ.  - ID đồ uống chưa có trong cơ sở dữ liệu | Hiển thị thông báo:Thêm đồ uống thành công.  Lưu đồ uống vào cơ sở dữ liệu. |

### 4.2. Chức năng xóa đồ uống:





**4.2.1. Đồ thị thể hiện đường diễn tiến của chương trình:**

****

**4.2.2 Các đường kiểm thử:**

Xác định số đường kiểm thử: V(G) = 28-23+2 = 7

1. 1.2.3.4.5.23

2. 1.2.3.6.7.8.23

3. 1.2.3.6.9.10.23

4. 1.2.3.6.9.10.11.12.13.16.17.18.19.23

5. 1.2.3.6.9.10.11.12.13.14.13...16.17.18.19.23

6. 1.2.3.6.9.10.11.12.13.14.13…15.16.17.20.21.22.23

7. 1.2.3.6.9.10.11.12.13.14.13.16.17.18.19.23

**4.2.3 Xác định các testcase:**

|  |  |  |
| --- | --- | --- |
| **Đường kiểm thử** | **Giá trị đầu vào** | **Kết quả mong đợi** |
| **1** | -Id là một chuỗi rỗng. | Hiển thị thông báo: Id chưa nhập! |
| **2** | -Id định dạng sai | Hiển thị thông báo:Id định dạng sai |
| **3** | -Từ chối xóa thức uống | -Hủy xóa thành công |
| **4** | -Id nhập vào đúng định dạng  - Danh sách đồ uống rỗng | Hiển thị thông báo:Id này không tồn tại |
| **5** | -Id nhập vào đúng định dạng  - Id nhập vào không tồn tại trong danh sách đồ uống | Hiển thị thông báo:Id này không tồn tại |
| **6** | -Id nhập vào đúng định dạng  -Id tồn tại trong danh sách | Hiển thị thông báo : Xóa thức uống thành công |
| **7** | -Id nhập vào đúng định dạng  -Id không tồn tại trong danh sách  -Danh sách đồ uống chỉ có 1 đồ uống | Hiển thị thông báo:Id này không tồn tại |