**BIÊN BẢN NGHIỆM THU PHẦN MỀM**

**1. Đối tượng nghiệm thu:**

\* Phần mềm: Phần mềm quản lý hoạt động bán hàng tại quán café Mô (138 Nguyễn Huệ).

**2. Thành phần nghiệm thu:**

|  |  |
| --- | --- |
| 21T1020581 | Nguyễn Ngọc Cảnh Phong |
| 21T1020210 | Đỗ Thị Diệu Ái |
| 21T1020035 | Trần Văn Huy |
| 21T1020146 | Hoàng Thị Kiều Oanh |
| 21T1020729 | Nguyễn Thị Minh Thư |
| 21T1020761 | Lê Thị Bảo Trâm |
| 21T1020818 | Hoàng Thanh Tùng |

**3. Thời gian tiến hành nghiệm thu:**

Bắt đầu: Thứ 7, ngày 11 tháng 05 năm 2024.

Kết thúc: Thứ , ngày tháng 05 năm 2024.

Tại:

**4. Đánh giá hạng mục:**

4.1 Tài liệu nghiệm thu: Hồ sơ thiết kế Phần mềm quản lý hoạt động bán hàng tại quán café Mô (138 Nguyễn Huệ).

4.2 Nội dung nghiệm thu: Tiến hành kiểm thử module chức năng “Thêm món”

Môi trường kiểm tra: C#, .Net.

Công cụ hỗ trợ: Visual Studio.

Dữ liệu dùng để kiểm tra: Dữ liệu được tích hợp tồn tại trong hồ sơ thiết kế dự án.

Các tính năng của phần mềm: “Thêm món”.

4.3 Tiến hành nghiệm thu:

4.3.1 Chương trình:

a. Chương trình giao diện:

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

{

string name = txbFoodName.Text;

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

float price = (float)nmFoodPrice.Value;

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

{

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

LoadListFood();

if (insertFood != null)

insertFood(this, new EventArgs());

}

else

{

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

}

}

b. Chương trình chính:

public bool InsertFood(string name, int id, float price)

{

string query = string.Format("INSERT dbo.Food ( name, idCategory, price )VALUES ( N'{0}', {1}, {2})", name, id, price);

int result = DataProvider.Instance.ExecuteNonQuery(query);

return result > 0;

}

c. Chương trình phụ:

public int ExecuteNonQuery(string query, object[] parameter = null)

{

int data = 0;

using (SqlConnection connection = new SqlConnection(connectionSTR))

{

connection.Open();

SqlCommand command = new SqlCommand(query, connection);

if (parameter != null)

{

string[] listPara = query.Split(' ');

int i = 0;

foreach (string item in listPara)

{

if (item.Contains('@'))

{

command.Parameters.AddWithValue(item, parameter[i]);

i++;

}

}

}

data = command.ExecuteNonQuery();

connection.Close();

}

return data;

}

d. Chương trình lấy danh sách thức ăn:

public List<Food> GetListFood()

{

List<Food> list = new List<Food>();

string query = "select \* from Food";

DataTable data = DataProvider.Instance.ExecuteQuery(query);

foreach (DataRow item in data.Rows)

{

Food food = new Food(item);

list.Add(food);

}

return list;

}

e. Chương trình cập nhập danh sách hiển thị món:

void LoadListFood()

{

foodList.DataSource = FoodDAO.Instance.GetListFood();

}

4.3.2 Cấu trúc chương trình.

a. Chương trình giao diện:

{**A**}

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

{

//P1

string name = txbFoodName.Text;

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

float price = (float)nmFoodPrice.Value; //P1

if (FoodDAO.Instance.InsertFood(name, categoryID, price)) //E1

{

MessageBox.Show("Thêm món thành công"); // P2

LoadListFood(); // P3

if (insertFood != null) // E4

insertFood(this, new EventArgs()); //P5

}

else

{

MessageBox.Show("Có lỗi khi thêm thức ăn"); //P6

}

}

{**B**}

b. Chương trình chính:

public bool InsertFood(string name, int id, float price)

{

string query = string.Format("INSERT dbo.Food ( name, idCategory, price )VALUES ( N'{0}', {1}, {2})", name, id, price); //P7

int result = DataProvider.Instance.ExecuteNonQuery(query); //P8

return result > 0; //E2

}

c. Chương trình phụ:

public int ExecuteNonQuery(string query, object[] parameter = null)

{

int data = 0; //P9

using (SqlConnection connection = new SqlConnection(connectionSTR)) //P10

{

connection.Open(); //P11

SqlCommand command = new SqlCommand(query, connection); //P12

if (parameter != null) //E3

{

string[] listPara = query.Split(' '); //P13

int i = 0; //P14

// first = &ListPara.First();

// last = &ListPara.Last();

// item = \*first

//C

foreach (string item in listPara) //{C, first NOT EQUAL last}

{

if (item.Contains('@')) //{C, first NOT EQUAL last} P15, P16 {C}

{

command.Parameters.AddWithValue(item, parameter[i]);

i++;

}

}

//C

}

data = command.ExecuteNonQuery(); //P17

connection.Close(); //P18

}

return data; //P19

}

d. Chương trình lấy danh sách thức ăn:

public List<Food> GetListFood()

{

List<Food> list = new List<Food>(); //P20

string query = "select \* from Food"; //P21

DataTable data = DataProvider.Instance.ExecuteQuery(query); //P22

// first = &ListPara.First();

// last = &ListPara.Last();

// item = \*first

//{D}

foreach (DataRow item in data.Rows) // {D, first NOT EQUAL last}

{

Food food = new Food(item); //P23

list.Add(food); //P24

}

//{D, first EQUAL last}

return list; //P25

}

e. Chương trình cập nhập danh sách hiển thị món:

void LoadListFood()

{

foodList.DataSource = FoodDAO.Instance.GetListFood(); //P26

}

4.3.3 Xác định mệnh đề vào, ra, bất biến:

Mệnh đề vào: {A: sender, e; sender € object, e € EventArgs}

Mệnh đề ra: {B: MessageBox; MessageBox € enums}

Mệnh đề bất biến: {C: item. Contain (‘@’)?, Parameter.Add(item, parameter[i])}; item € listPara, parameter € object [], listPara = SQLInsertCommandFormat(name, categoryID, price).split(), listPara € string[]}

4.3.4 Kế hoạch kiểm thử:

Bước 1: Dự trù mệnh đề {C} bất biến.

Bước 2: Kiểm tra:

{A}

P1; // {A} P1 {A1}

{A1, E1};

{

P7; //{A1} P7 {A2}

P8 //{A2} P8 {}

{

P9; //{A2} P9 {A3}

P10;

P11;

P12; //{A3} P12 {A4}

E3{

// {A4, E3}

P13; // {A4, E3} P13 {A5}

P14; // {A5} P14 {A6}

first = &ListPara.First();

Last = &ListPara.Last();

item = \*first;

{C}

Bước 3: Kiểm tra {C} là bất biến vòng lặp.

{C}

{C, first NOT EQUAL last)

P15;

P16;

{C}

Bước 4: Kiểm tra

{C, first EQUAL last}

P17; // {C, first EQUAL last} P17 {C1}

P18;

}

P19;

}

E2;

}

// {C, E1}

{

P2;

P3{

P26 {P20, P21, P22, P23, P24, P25}

}

//E4

// {C, E4} P5 {B}

P5

// {C, !E4} P6 {B}

P6

}

{B}

4.3.5 Thực hiện kiểm thử.

Tiến hành kiểm thử ta thấy trong điều kiện chức năng hiện tại E3 vốn không thể thực hiện và từ đó việc khởi tạo vòng lặp không thể xảy ra.

Lí do cho việc E3 được xem là không thể thực hiện bởi tại <A, E1>:

* Xem xét P7, P8, E2 ta thấy, với:

+ P7 là đoạn trình tạo nên đầu vào cho P8. Tại đây {A} thay đổi tính chất từ {A: name, categoryId, price; name € string, categoryId € int, price € float} thành {A1: query, query € string}.

+ P8 là đoạn trình truy vấn dữ liệu dựa trên đầu vào là kết quả của P7. Với đầu vào là một query, P8 khởi tạo biến parameter mang giá trị mặc định null và gọi đến các chương trình P11, P12, P9, P10 và E3.

+ Xem xét {A, E3} ta thấy mệnh đề này không thể thực hiện. Vì với parameter là null không thể thỏa mãn điều kiện để thực hiện vòng lặp. Cho nên việc hình thành vòng lặp không thể tồn tại. Từ đó không thể hình thành vòng lặp. Từ đó ta có thể kết luận {A} P {B} không hoàn đúng.

Ngoài ra, nếu ta không xem xét đến E3, P13, P14, P15, P16 thì việc chứng mình {A} P {B} là đúng.

Với điều trên, ta nhận thấy rằng, {A} P {B} không hoàn toàn đúng, cũng không hoàn sai.

**5. Các ý kiến khác (nếu có):**

**6. Kết luận**

**ĐẠI DIỆN ĐƠN VỊ NGHIỆM THU PHẦN MỀM**