**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 “Sửa bàn”

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: “Sửa bàn”.

4.3 Tiến hành nghiệm thu:

4.3.1 Chương trình:

a. Chương trình giao diện:

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

{

string name = txbTableName.Text;

int id = Convert.ToInt32(txbTableId.Text);

if (TableDAO.Instance.UpdateTable(id, name))

{

MessageBox.Show("Sửa bàn thành công");

LoadListTable();

if (updateTable != null)

updateTable(this, new EventArgs());

}

else

{

MessageBox.Show("Có lỗi khi sửa bàn");

}

}

b. Chương trình chính:

public bool UpdateTable(int idTable, string name)

{

string query = string.Format("UPDATE dbo.TableFood SET name = N'{0}' WHERE id = {1}", name, idTable);

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

return result > 0;

}

c. Chương trình truy vấn:

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 bàn:

public DataTable GetListTable()

{

return DataProvider.Instance.ExecuteQuery("SELECT id, name FROM dbo.TableFood");

}

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

void LoadListTable()

{

tableList.DataSource = TableDAO.Instance.GetListTable();

}

4.3.2 Cấu trúc chương trình:

a. Chương trình giao diện:

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

{

string name = txbTableName.Text;

int id = Convert.ToInt32(txbTableId.Text);

//{A}

if (TableDAO.Instance.UpdateTable(id, name)) //{E1}

{ // {A, E1}

MessageBox.Show("Sửa bàn thành công"); //{P1} – L > {B}

LoadListTable(); //{P2}

if (updateTable != null) //{E2}

updateTable(this, new EventArgs()); //{P3}

}

else

{

MessageBox.Show("Có lỗi khi sửa bàn"); //{P4} – L > {B}

}

}

b. Chương trình chính:

public bool UpdateTable(int idTable, string name)

{

string query = string.Format("UPDATE dbo.TableFood SET name = N'{0}' WHERE id = {1}", name, idTable); //{P5}

int result = DataProvider.Instance.ExecuteNonQuery(query); //{P6} - L > {C}

return result > 0; //{E5}

}

c. Chương trình truy vấn:

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

{

int data = 0; //{P7}

using (SqlConnection connection = new SqlConnection(connectionSTR)) //{P8}

{

connection.Open(); //{P9}

SqlCommand command = new SqlCommand(query, connection); //{P10}

if (parameter != null) //{E3}

{

string[] listPara = query.Split(' '); //{P11}

int i = 0; // {P12}

//{C}

// first = &ListPara.First();

// last = &ListPara.Last();

// item = \*first

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

{

if (item.Contains('@'))

{

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

i++;

} //{C,E4} P13 {C}

// Increase (first);

} //C

}

data = command.ExecuteNonQuery(); //{P14}

connection.Close(); //{P15}

}

return data; //{P16}

}

d. Chương trình lấy danh sách bàn:

public DataTable GetListTable()

{

return DataProvider.Instance.ExecuteQuery("SELECT id, name FROM dbo.TableFood"); //P17

}

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

void LoadListTable()

{

tableList.DataSource = TableDAO.Instance.GetListTable(); //P18

}

Cấu trúc chương trình

{A}

{A, E1}

{ P5 // {A, E1} P5 A1

P6 {

Parameter = null

P7 //A1 P7 A2

P8 {

P9 //A2 P9 A3

P10 //A3 P10 A4

{E3, A4}

{

P11 //A4 P11 A5

P12 //A5 P12 A6

first = &ListPara.First();

last = &ListPara.Last();

item = \*first

//C

{C, first NOT EQUAL last}

P13

Increase (first)

// {C, first EQUAL last}

}

P14 // {C, first EQUAL last} P14 C1

P15 // C1 P15 C2

}

P16 // C2 P16 C3

}

E5 – L > E1 // {C, E5} – L > B

P1 – L > B

P2 { P18 { P17 } }

E2

P3

}

P4 – L > B

{B}

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

Mệnh đề vào {A: id, name; id € int, name € string}.

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

Mệnh đề bất biến 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 {C} bất biến của vòng lặp.

Bước 2: Kiểm tra.

{A}

{A, E1}

{ P5 // {A, E1} P5 {1}

P6 {  
 P7 // A1 P7 A2

P8 {

P9 // A2 P9 A3

P10 //A3 P10 A4

{A4, E3}

{

P11 // A4 P11 A5

P12 // A5 P12 A6

first = &ListPara.First();

last = &ListPara.Last();

item = \*first

//C

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

{C}

{C, first NOT EQUAL last}

P13

Increase (first)

}

{C}

// {C, first EQUAL last}

Bước 4: Kiểm tra

// {C, first EQUAL last}

P14 // {C, first EQUAL last} P14 {C1}

P15 // {C1} P15 {C2}

}

P16 // {C2} P16 {C3}

}

E5 – L > E1 // {C, E5} – L > {B}

P1 – L > B

P2

E2

P3

}

P4 – L > B

{B}

4.3.5 Tiến hành kiểm thử.

Gọi mệnh đề thể hiện tính chất dữ liệu vào của chương trình {A} và mệnh đề thể hiện tính chất dữ liệu ra cần có {B}, ta có:

{A: id, name; id € int, name € string} và {B: MessageBox; MessageBox € enums}.

Ta cần chứng tỏ {A} P {B}.

+ Xét mệnh đề bất biến {C: id, name, command, listPara, i; id € int, name € string, command € SQLcommand, listPara € string [], i € int}.

+ Ta có {A, E1} P5, P6, P7, P8, P9, P10 {A4}

{A4, E3} P11, P12, first, last, item {C}

+ Để chứng tỏ {C} là đoạn trình bất biến

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

{

if (item.Contains('@'))

{

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

i++;

}

}

Ta cần có {C, first NOT EQUAL last} P13 {C}.

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>:

* Khi xem xét E1 ta thấy, việc kiểm tra điều kiện của E1 thông qua việc thực hiện P5, P6, E1 với:

+ P5 là đoạn trình tạo nên đầu vào cho P6. Hay {A} P5 {A1}. Tại đây {A} thay đổi tính chất từ {A: id, name; id € int, name € string} thành {A1: query, query € string}.

+ P6 là đoạn trình truy vấn dữ liệu dựa trên đầu vào là A1 của P5. Với đầu vào là một query của A1, P6 khởi tạo biến parameter và gọi đến các chương trình P7, P8, P9, P10 và E3: {A1} P7, P8, P9, P10 {A4}.

* Xem xét {A4, E3} ta thấy mệnh đề này không thể thực hiện. Cho nên việc hình thành vòng lặp không thể tồn tại. Từ đó không thể hình thành vòng lặp.

Ngoài ra, nếu ta không xem xét đến E3, P11, P12, P13 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**