**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:

{A}

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

{

string name = txbTableName.Text;

//P1

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

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

{

MessageBox.Show("Sửa bàn thành công"); //P2

LoadListTable(); //P3

if (updateTable != null) //E2

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

}

else

{

MessageBox.Show("Có lỗi khi sửa bàn"); //P4

}

}

{B}

\* Mệnh đề vào, ra, bất biến:

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

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

\* Kế hoạch kiểm thử:

Kiểm tra {A} P {B}

{A}

P1; //{A} P1 {A1}

E1

{

// {A1, E1}

P2; // {A1, E1} P2 {A2}

P3; // {A2} P3 {A3}

E2

{

// {A3, E2}

P3; // {A3, E2} P3 {B}

}

// {A3, !E2} = L > {B}

}

// {A1,!E1}

P5; {A1, !E1} P5 {B}

{B}

b. Chương trình chính:

{AI}

{

string query = string.Format("UPDATE dbo.TableFood SET name = N'{0}' WHERE id = {1}", name, idTable); //{P6}

int result = DataProvider.Instance.ExecuteNonQuery(query); //{P7}

return result > 0; //{E3}

}

{BI}

\* Mệnh đề vào, ra, bất biến:

- Mệnh đề vào: {AI: idTable, name, result; idTable € int, name € string, result € bool}.

- Mệnh đề ra: {BI: result > 0?; result € bool}

\* Kế hoạch kiểm thử:

Bước 1: Kiểm tra

{AI}

P6; //{AI} P6 {AI1}

P7; //{AI1} P7 {AI2}

E3 // {AI2, E3} = L > B

// {AI2, !E3} = L > B

{BII}

c. Chương trình truy vấn:

{AII}

{

object[] parameter = null;

//P8

int data = 0;

using (SqlConnection connection = new SqlConnection(connectionSTR)) //P9

{

connection.Open(); //P10

SqlCommand command = new SqlCommand(query, connection); //P11

if (parameter != null) //E4

{

string[] listPara = query.Split(' '); //P12

int i = 0; //P13

// first = &ListPara.First();

// last = &ListPara.Last();

// item = \*first

//{C}

foreach (string item in listPara) //first NOT EQUAL last

{

if (item.Contains('@')) //E5

{

command.Parameters.AddWithValue(item, parameter[i]); //P14

i++; //P15

}

}

}

data = command.ExecuteNonQuery(); //P16

connection.Close(); //P17

}

return data; //P18

}

{BII}

\* Mệnh đề vào, ra, bất biến:

Mệnh đề vào: {AII: query, parameter [], data; query € string, parameter [] € object, data € int}.

Mệnh đề ra: {BII: data; data € int}

\* Kế hoạch kiểm thử.

Bước 1: Dự trù mệnh đề bất biến {C}

Bước 2: Kiểm tra.

**{AII}**

**P8; //{AII1} P8 {AII2}**

**P9 //{AII2} P9 {AII3}**

**{**

**P10; //{AII3} P10 {AII4}**

**P11; //{AII4} P11 {AII5}**

**E4**

**{**

**// {AII5, E4}**

**P12 // {AII5, E4} P12 {AII6}**

**P13 // {AII6} P13 {AII7}**

// first = &ListPara.First();

// last = &ListPara.Last();

// item = \*first;

{C}

Bước 3: Kiểm tra {C} là mệnh đề bất biến:

{C}

**first NOT EQUAL last**

**{**

**// {C, first NOT EQUAL last}**

**E5**

**{**

**// {C, first NOT EQUAL last E5}**

**P14;**

**// {C, first NOT EQUAL last E5} P14, P15 {C}**

**P15;**

**Increase (first)**

**}**

**// {C, first NOT EQUAL last !E5}**

**}**

**{C, first EQUAL last}**

Bước 4: Kiểm tra:

**// {C, first EQUAL last}**

**{**

**P16; {**C, first EQUAL last} P16 {C1}

**P17;** {C1} P17 {C2}

}

**P18; //{C2} P18 {B}**

**{BII}**

**5. Các ý kiến khác (nếu có):**

**6. Kết luận**

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