



PROGRAMMING IN .NET

Assignment 2

Duong Xuan Trung – GCH15332

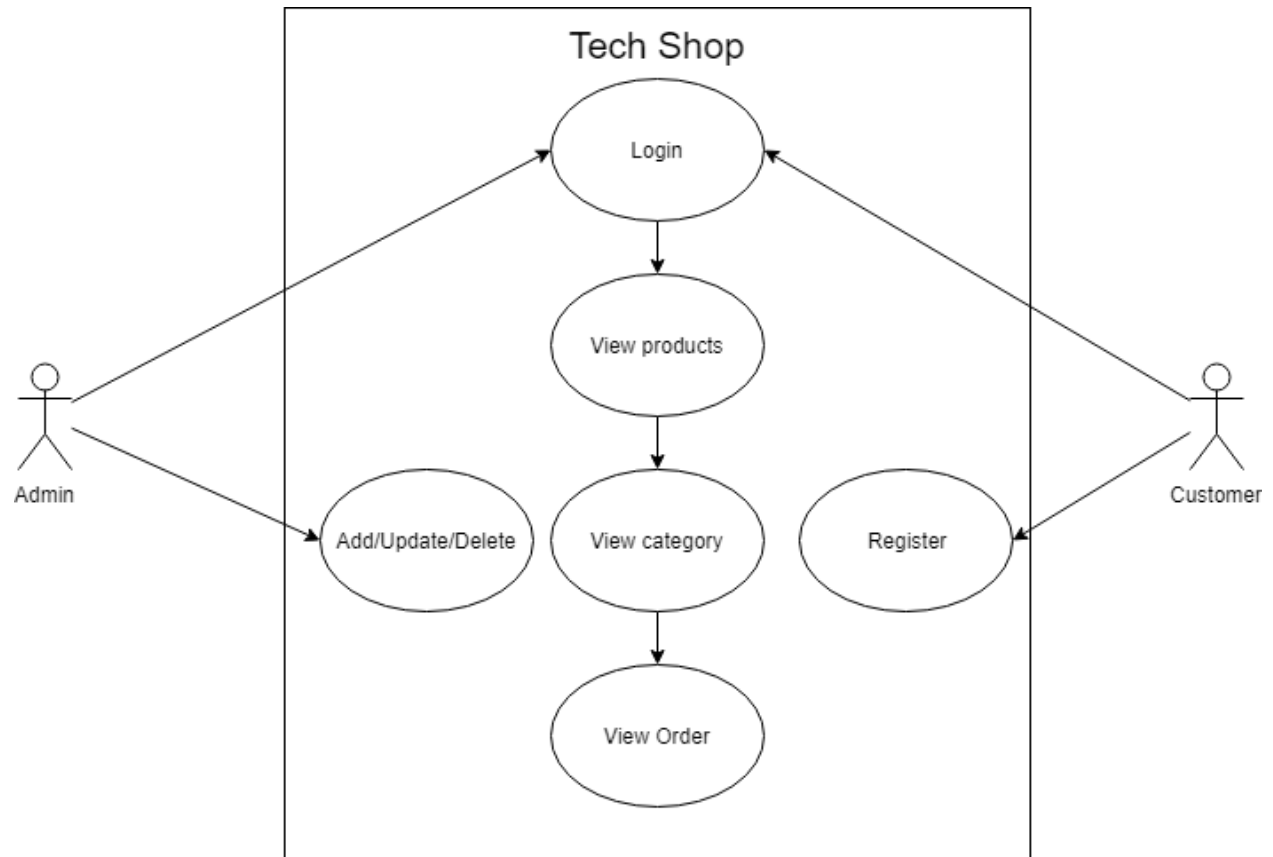
Trung Duong
[Email address]

Contents

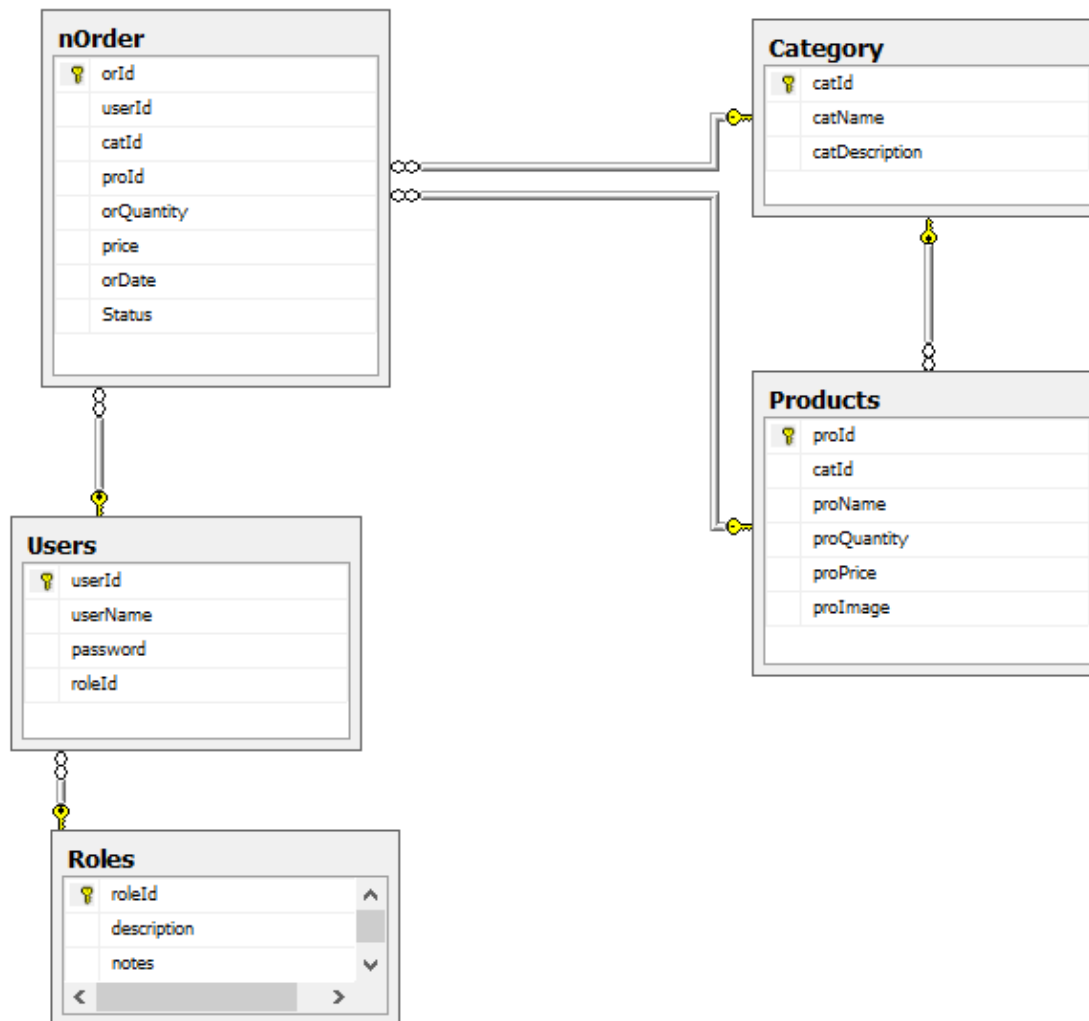
| | |
|--|----|
| Task 1 | 2 |
| I. Use case diagram | 2 |
| II. ERD..... | 3 |
| III. Class diagram | 4 |
| IV. GUI design..... | 5 |
| Task 2 | 9 |
| I. Files structure of the program | 9 |
| LoginForm.cs | 10 |
| AddUserForm.cs..... | 11 |
| DashBoardForm.cs | 12 |
| AddCategoryForm.cs..... | 15 |
| AddForm.cs | 16 |
| OrderForm.cs | 17 |
| II. Graphical components..... | 20 |
| Login Form | 20 |
| DashBoard Form | 21 |
| AddUser Form | 22 |
| AddCategory Form | 23 |
| Add Form (For product) | 24 |
| Order Form..... | 25 |
| Task 3 | 25 |
| Task 4 | 27 |
| I. Program screen..... | 27 |
| Login Form | 27 |
| Register Form | 27 |
| DashBoard Form | 27 |
| AddUser Form | 29 |
| AddCategory Form | 29 |
| Add Form (For product) | 30 |
| Order Form..... | 30 |
| II. Error handling | 31 |

Task 1

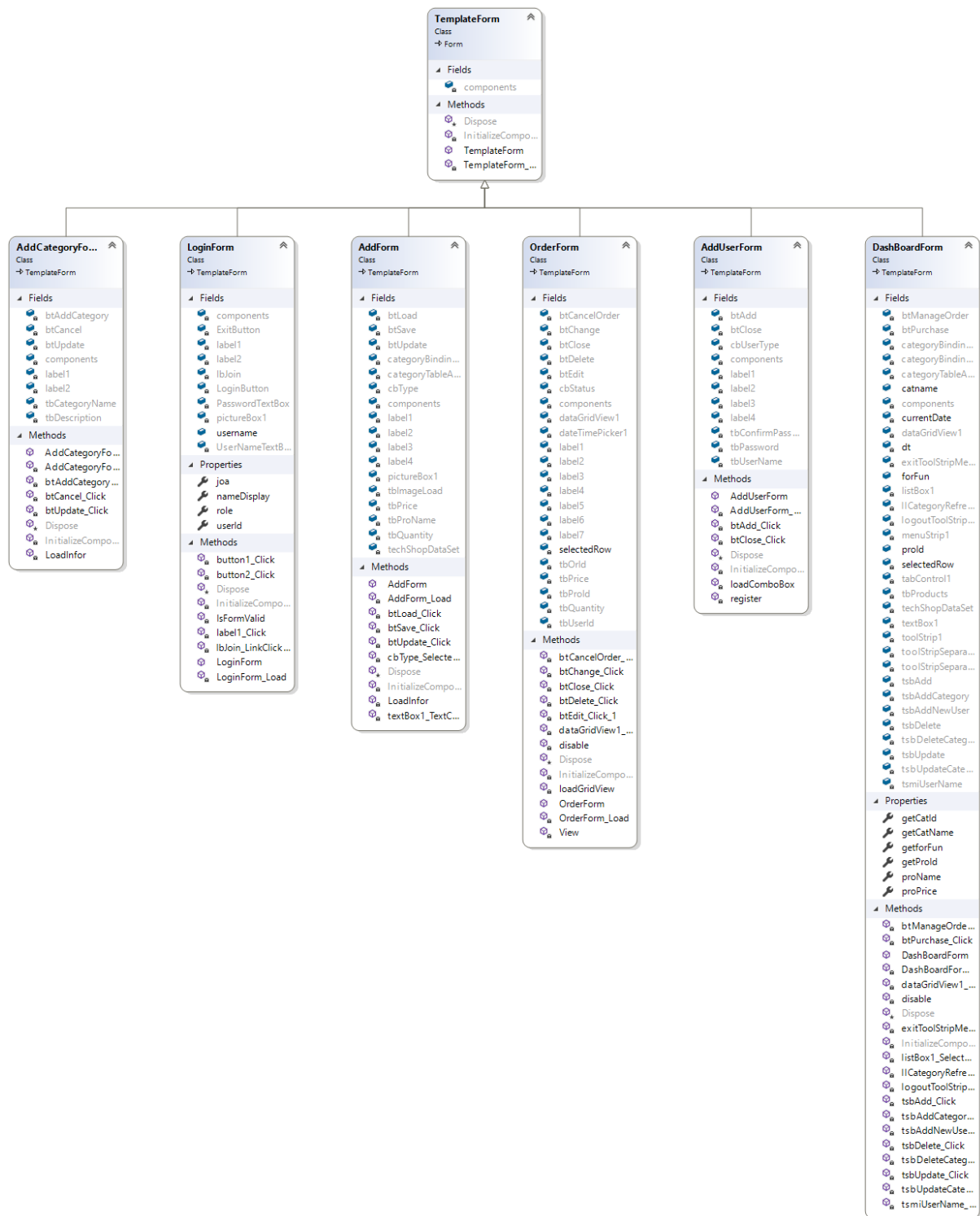
I. Use case diagram

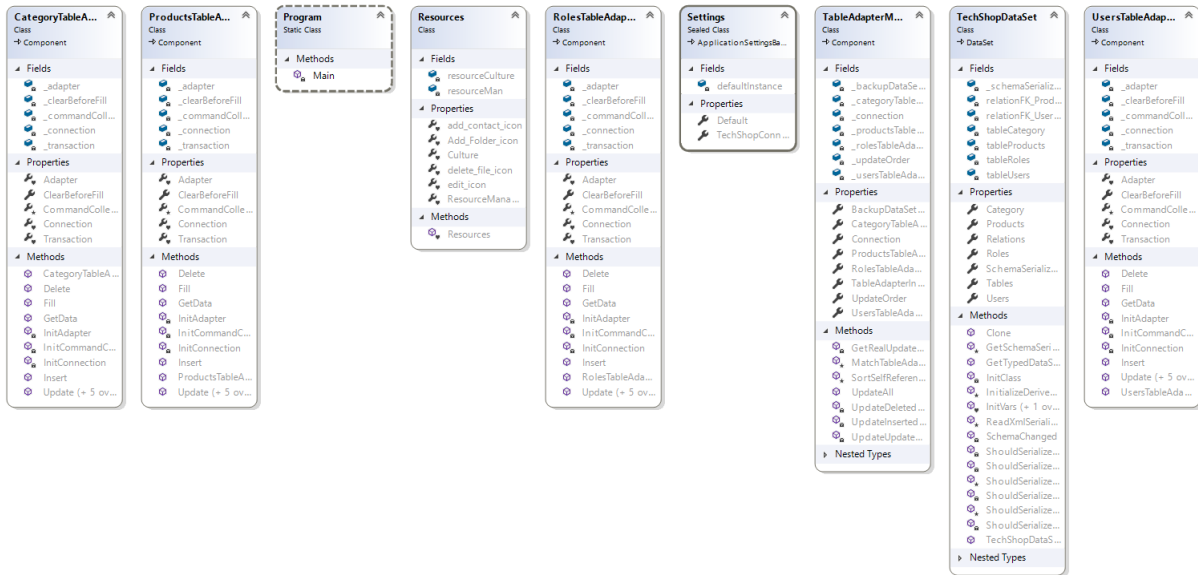


II. ERD



III. Class diagram





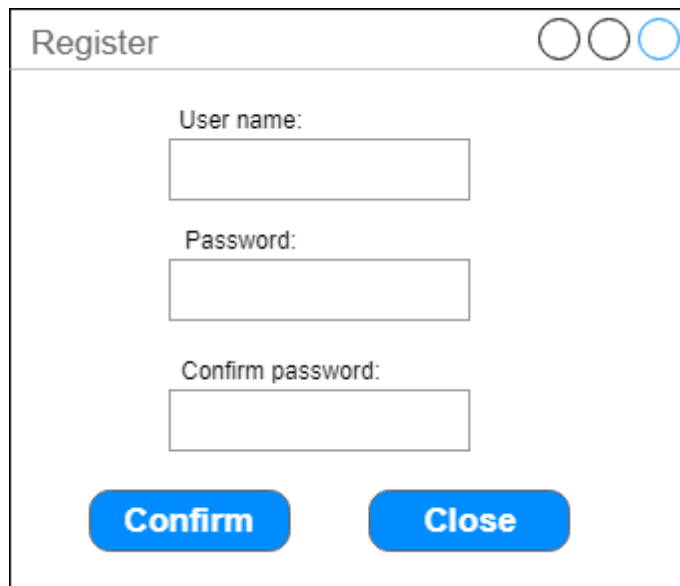
IV. GUI design

This will show the function also user interface that I'm going to design for my program

Login form:

Before go further user would have to login first.

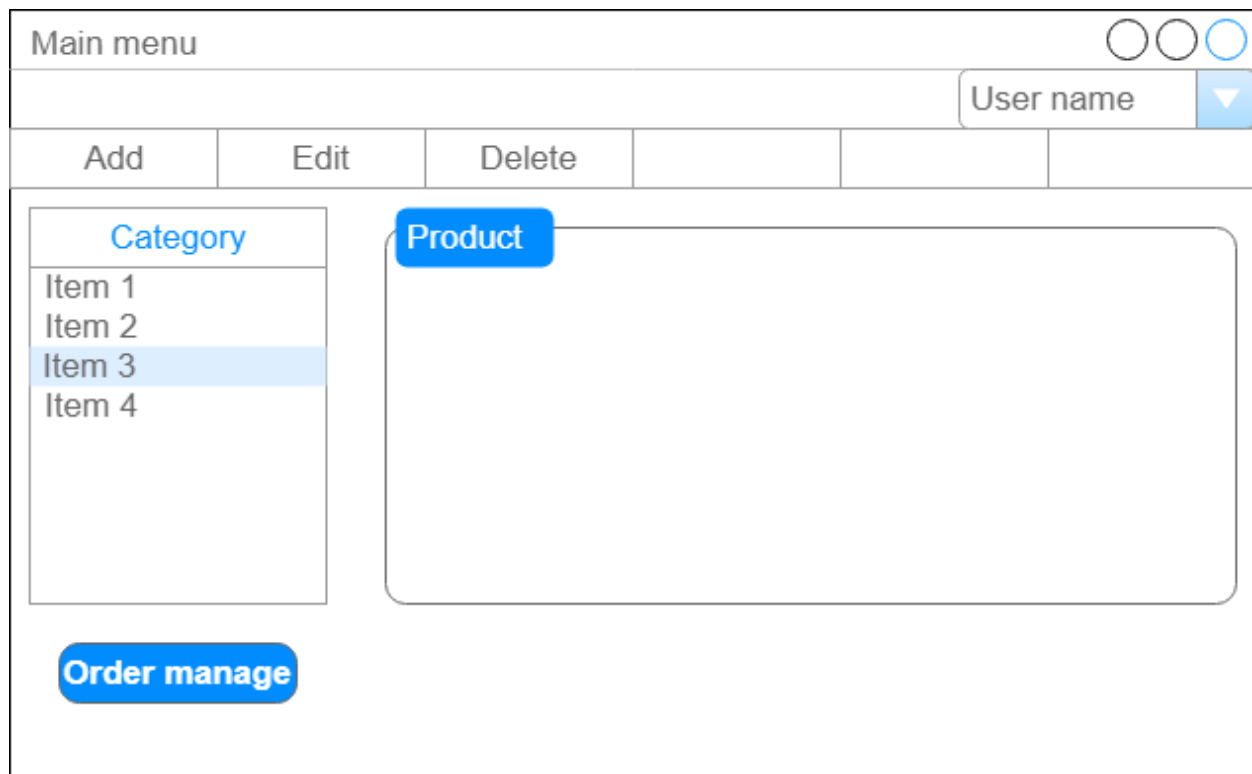
Sign up form:



A window titled "Register" with three input fields for "User name:", "Password:", and "Confirm password:". Below the fields are two blue buttons labeled "Confirm" and "Close".

Incase user don't have the account to login. They can sign-up one.

Main menu form for admin:



A window titled "Main menu" with a header bar containing a "User name" dropdown menu. Below the header is a table with columns "Add", "Edit", "Delete", and two empty columns. The table has four rows: "Category", "Item 1", "Item 2", "Item 3", and "Item 4". "Item 3" is highlighted. To the right of the table is a large blue button labeled "Product". Below the table is a blue button labeled "Order manage".

This include Add, update, delete for category and product and also can add user.

Add/update category:

Add category

Category name

Description

Add

Close

Add/update product:

Add/Update Product

Category

Product name

Quantity

Price

Browser

Add

Close

Orders management form for admin:

Order manage

Order ID

User name

Product

Quantity

Price

Date

Status

Change

Edit

Cancel

Close

Delete

Orders

Main menu for User:

Main menu

User name

Category

Item 1

Item 2

Item 3

Item 4

Product

Order manage

Order

Order products form for user:

Order manage

Order ID

User name

Product

Quantity

Price

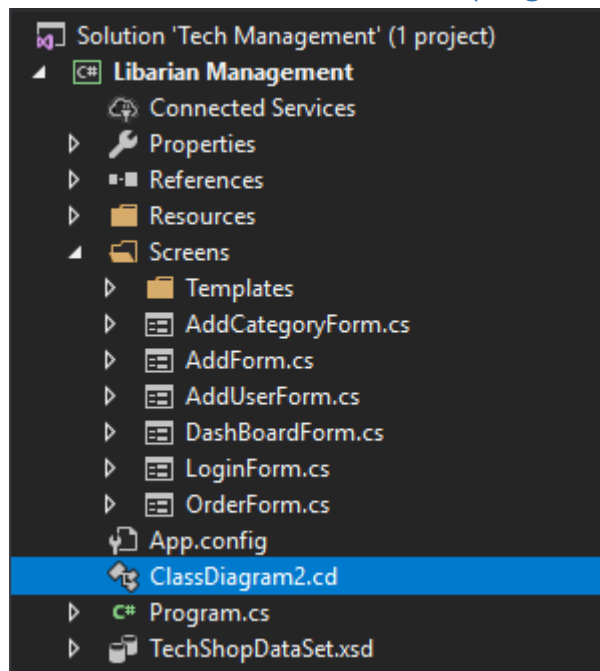
Date

Status

Orders

Task 2

I. Files structure of the program



The program contains 7 .cs files.

When the program starts, Program.cs file creates new Login form as starting point.

```

1  using Librarian_Management.Screens;
2  using System;
3  using System.Collections.Generic;
4  using System.Linq;
5  using System.Threading.Tasks;
6  using System.Windows.Forms;
7
8  namespace Librarian_Management
9  {
10     static class Program
11     {
12         /// <summary>
13         /// The main entry point for the application.
14         /// </summary>
15         [STAThread]
16         static void Main()
17         {
18             Application.EnableVisualStyles();
19             Application.SetCompatibleTextRenderingDefault(false);
20             Application.Run(new LoginForm());
21         }
22     }
23 }
24

```

LoginForm.cs

```

private void button1_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    String query = "Select * from Users where userName = '"+UserNameTextBox.Text.Trim()+"' and password='"+PasswordTextBox.Text.Trim()+"'";
    SqlDataAdapter sda = new SqlDataAdapter(query, conn);
    DataTable result = new DataTable();
    sda.Fill(result);
    if (IsValidForm())
    {
        if (result.Rows.Count == 1)
        {
            userId = result.Rows[0][0].ToString();
            role = result.Rows[0][3].ToString();
            nameDisplay = UserNameTextBox.Text;
            this.Hide();
            DashBoardForm db = new DashBoardForm();
            db.Show();
        }
        else
        {
            MessageBox.Show("User name/Password is not correct.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
            UserNameTextBox.Focus();
        }
    }
}

```

In here when user input user name and password into text box then click on login button, it will perform checking form data base if the value exist it will go to main menu form.

```

private bool IsFormValid()
{
    if (UserNameTextBox.Text.Trim() == String.Empty)
    {
        MessageBox.Show("User name is required.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        UserNameTextBox.Clear();
        UserNameTextBox.Focus();
        return false;
    }

    if (PasswordTextBox.Text.Trim() == String.Empty)
    {
        MessageBox.Show("Password is required.", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);
        PasswordTextBox.Clear();
        PasswordTextBox.Focus();
        return false;
    }

    return true;
}

```

Also incase user doesn't input any values into the textbox this will check and require user to input them.

AddUserForm.cs

```

private void btAdd_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    String query = "Select * from Users where userName = '" + tbUserName.Text.Trim() + "'";
    SqlDataAdapter sda = new SqlDataAdapter(query, conn);
    DataTable result = new DataTable();
    sda.Fill(result);
    if (result.Rows.Count == 1)
    {
        MessageBox.Show("Username already used!");
    }
    else
    {
        if (tbPassword.Text.Equals(tbConfirmPassword.Text))
        {
            String query1 = "Insert into Users values('" + tbUserName.Text.Trim() + "','" + tbPassword.Text.Trim() + "','" + cbUserType.SelectedValue + "')";
            SqlCommand cmd = new SqlCommand(query1, conn);
            try
            {
                conn.Open();
                cmd.ExecuteNonQuery();
                conn.Close();
                MessageBox.Show("Add success");
                this.Close();
                LoginForm.joa = 0;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
        else
        {
            MessageBox.Show("Confirm password not correct!");
        }
    }
}

```

This is the function for add user new user into the system. This option just available for admin so they are the only one can use this. In here when admin input all information into text box then click on add button, it will perform command to add the information onto data base.

DashBoardForm.cs

```
private void DashBoardForm_Load(object sender, EventArgs e)
{
    // TODO: This line of code loads data into the 'techShopDataSet.Category' table. You can move, or remove it, as needed.
    this.categoryTableAdapter.Fill(this.techShopDataSet.Category);
    textBox1.Text = LoginForm.role;
    textBox1.Enabled = false;
    tsmiUserName.Text = LoginForm.username;
    tsbUpdate.Enabled = false;
    tsbDelete.Enabled = false;
    btPurchase.Enabled = false;
    //listBox1.SelectedIndex = 1;
    listBox1.SelectedIndex = 0;
    if (LoginForm.role == "2")
    {
        toolStrip1.Visible = false;
        btPurchase.Visible = true;
        btManageOrder.Text = "Your Order";
    }
    else if(LoginForm.role == "1")
    {
        toolStrip1.Visible = true;
        btPurchase.Visible = false;
        btManageOrder.Text = "Manage Order";
    }
}
```

Because the dashboard form is available for both user type, so as we see there are few rule in dashboard form for each user type (Admin and customer). After login this form will get username, user ID and also user role that 1 is admin and 2 if customer. For customer, they can't see the admin tool to add/update/delete for category and product, they just able to give order to a product. For admin they can see and use the admin tool to add/update/delete for category and product, but there are no order button for them to give order to a product. So it been set to "false" and the admin tool is set to "true".

```
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    catname = listBox1.Text;
    dt = new DataTable();
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    //getquery = listBox1.SelectedValue;
    getCatId = listBox1.SelectedValue;
    //String qry = ("select proid as 'Product ID', proName as 'Name', proQuantity as 'Quantity', proPrice as 'Price' from Products where catid = '" + getquery + "'");
    String qry = ("select proid as 'Product ID', proImage as 'Image', proName as 'Name', proQuantity as 'Quantity', proPrice as 'Price' from Products where catid = '" + getCatId + "'");
    SqlDataAdapter sdal = new SqlDataAdapter(qry, conn);
    sdal.Fill(dt);
    dataGridView1.DataSource = dt;
    BindingSource bSource = new BindingSource();
    bSource.DataSource = dt;
    dataGridView1.DataSource = bSource;
    sdal.Update(dt);
}
```

This function of category listbox, when in the dashboard form user can see the category so they can select them. Each time they select a category the product table will show that category products. This function could be like the search.

```
private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    selectedRow = e.RowIndex;
    if(selectedRow >= 0)
    {
        tsbUpdate.Enabled = true;
        tsbDelete.Enabled = true;
        btPurchase.Enabled = true;
        DataGridViewRow row = this.dataGridView1.Rows[selectedRow];
        //textBox1.Text = row.Cells["Product ID"].Value.ToString();
        proId = row.Cells["Product ID"].Value.ToString();
        proName = row.Cells["Name"].Value.ToString();
        proPrice = row.Cells["Price"].Value.ToString();
    }
}
```

This function is for admin only. The admin can see all the add/update/delete function but they are not able to use update and delete function yet because they have to select a product or a category first. When the product table showed products, then admin select a product the button update/delete will be available to use.

```
private void tsbUpdate_Click(object sender, EventArgs e)
{
    if (proId != "")
    {
        forFun = 2;
        AddForm af = new AddForm();
        af.Show();
    }
    else
    {
        MessageBox.Show("Please select a product");
    }
}
```

This function will open a form to admin manage a change of the product.

```
private void tsbDelete_Click(object sender, EventArgs e)
{
    DialogResult dr = MessageBox.Show("Do you really want to remove this product?", "Title", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);
    if (dr == DialogResult.Yes)
    {
        SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
        String query = "Delete from Products where proId='" + proId + "'";
        SqlCommand cmd = new SqlCommand(query, conn);
        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Deleted");
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}
```

When click on the delete button. This execute a command to database to remove a product that admin selected from database.

```

private void btnPurchase_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    //String query = "Insert into pOrder(userID, proId, orQuantity, price, orDate, Status) values('" + LoginForm.userId + "','" + getProId + "','1,'" + proPrice + "','" + currentDat
    String query = "Insert into newOrder(userName, catName, proName, orQuantity, price, orDate, Status) values('" + LoginForm.nameDisplay + "','" + DashBoardForm.getCatName + "','"
    SqlCommand cmd = new SqlCommand(query, conn);
    try
    {
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Purchased");
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

For order button, this will make a execute command into database to create an order of customer. It will get all the information of category, product and the user.

```

private void btnManageOrder_Click(object sender, EventArgs e)
{
    OrderForm of = new OrderForm();
    of.Show();
}

1 reference
private void tsbAddNewUser_Click(object sender, EventArgs e)
{
    AddUserForm auf = new AddUserForm();
    auf.Show();
}

1 reference
private void tsbUpdateCategory_Click(object sender, EventArgs e)
{
    if (getCatId != "")
    {
        forFun = 2;
        AddCategoryForm acf = new AddCategoryForm();
        acf.Show();
    }
    else
    {
        MessageBox.Show("Please select a product");
    }
}

```

```

1 reference
private void tsbAddCategory_Click(object sender, EventArgs e)
{
    forFun = 1;
    AddCategoryForm acf = new AddCategoryForm();
    acf.Show();
}

```

Each button will open another form such as order form, add user form, add/update category.

```

private void tsbDeleteCategory_Click(object sender, EventArgs e)
{
    DialogResult dr = MessageBox.Show("Do you really want to remove " + getCatName + " category?", "Title", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);
    if (dr == DialogResult.Yes)
    {
        SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
        String query = "Delete from Category where catId = '" + getCatId + "'";
        SqlCommand cmd = new SqlCommand(query, conn);
        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Deleted");
            conn.Close();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

In conclusion, this form will show category list, product list when you selected a category from category list. Also with the admin it will show and allow admin to perform add/update/delete form selected value.

AddCategoryForm.cs

```

private void btAddCategory_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    String query = "Insert into Category values('" + tbCategoryName.Text + "', '" + tbDescription.Text + "')";
    SqlDataAdapter sda = new SqlDataAdapter(query, conn);
    DataTable result = new DataTable();
    sda.Fill(result);
    SqlCommand cmd = new SqlCommand(query, conn);
    try
    {
        conn.Open();
        cmd.ExecuteNonQuery();
        conn.Close();
        MessageBox.Show("Add success");
        this.Close();
        DashboardForm.forFun = 0;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//reference
private void btUpdate_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    String query = "update Category set catName='" + this.tbCategoryName.Text + "', catDescription='" + this.tbDescription.Text + "' where catId = '" + DashboardForm.getCatId + "'";
    SqlCommand cmd1 = new SqlCommand(query, conn);
    try
    {
        conn.Open();
        cmd1.ExecuteNonQuery();
        MessageBox.Show("Update success");
        this.Close();
        DashboardForm.forFun = 0;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

As the form is use for both add and update category, so the add and update button will show-up or invisible in different situation. When admin select add new category the add button will appear and update button will disappear. So when admin select update category the add button will disappear and the update button will appear.

The add button will add new category into database and the update button will make change of current category that admin selected.


```

private void LoadInfor()
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    String query = "select * from Category where catId='" + DashBoardForm.getCatId.ToString() + "'";
    SqlCommand cmd = new SqlCommand(query, conn);
    SqlDataReader reader;
    try
    {
        conn.Open();
        reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            tbCategoryName.Text = (reader["catName"].ToString());
            tbDescription.Text = (reader["catDescription"].ToString());
        }
        conn.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

This form show-up when admin choose Add or Update category Function. If choose add new form will appear. If choose update the form will get information then fill text box. Each function will appear different button Add or Delete button.

AddForm.cs

```

private void btSave_Click(object sender, EventArgs e)
{
    byte[] imageBt = null;
    if (tbImageLoad.Text == "")
    {
        MessageBox.Show("Please add picture");
    }
    else
    {
        FileStream fstream = new FileStream(this.tbImageLoad.Text, FileMode.Open, FileAccess.Read);
        BinaryReader br = new BinaryReader(fstream);
        imageBt = br.ReadBytes((int)fstream.Length);

        SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
        String query = "Insert into Products(catId, proName, proQuantity, proPrice, proImage) values('" + this.cbType.SelectedValue + "','" + this.tbProName.Text + "','" + this.tbProQuantity.Text + "','" + this.tbProPrice.Text + "','" + imageBt.ToString() + "')";
        SqlCommand cmd1 = new SqlCommand(query, conn);
        try
        {
            conn.Open();
            cmd1.Parameters.Add(new SqlParameter("@IMG", imageBt));
            cmd1.ExecuteNonQuery();
            conn.Close();
            MessageBox.Show("Saved");
            this.Close();
            DashBoardForm.forFun = 0;
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

```

private void btUpdate_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    String query = "Update Products set catId='" + this.cbType.SelectedValue + "',proName='" + this.tbProName.Text + "',proQuantity='" + this.tbProQuantity.Text + "',proPrice='" + this.tbProPrice.Text + "'";
    SqlCommand cmd1 = new SqlCommand(query, conn);
    try
    {
        conn.Open();
        //cmd1.Parameters.Add(new SqlParameter("@IMG", imageBt));
        cmd1.ExecuteNonQuery();
        MessageBox.Show("Update success");
        this.Close();
        DashBoardForm.forFun = 0;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

Same as add or update category. This form will show-up for admin to add new product or edit information of a product. As this form is for both add new and update current product so it will show with different button.

```
private void LoadInfor()
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    String query = "Select * from Products where proId = '" + DashBoardForm.getProId + "'";
    SqlCommand cmd = new SqlCommand(query, conn);
    SqlDataReader reader;
    try
    {
        conn.Open();
        reader = cmd.ExecuteReader();
        if (reader.Read())
        {
            tbProName.Text = (reader["proName"].ToString());
            tbQuantity.Text = (reader["proQuantity"].ToString());
            tbPrice.Text = (reader["proPrice"].ToString());
            byte[] imgg = (byte[])(reader["proImage"]);
            if (imgg == null)
            {
                pictureBox1.Image = null;
            }
            else
            {
                MemoryStream mstream = new MemoryStream(imgg);
                pictureBox1.Image = System.Drawing.Image.FromStream(mstream);
            }
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

When admin select update a product this will get the information if that product then fill it into the textbox so user can edit then then click update.

OrderForm.cs

For the order form, it is use for both user type (admin and customer) the form will based on the user role to disable and enable the function.

```

private void dataGridView1_CellContentClick(object sender, DataGridViewCellEventArgs e)
{
    selectedRow = e.RowIndex;
    if (selectedRow >= 0)
    {
        DataGridViewRow row = this.dataGridView1.Rows[selectedRow];
        /*tbOrderId.Text = row.Cells["orderId"].Value.ToString();
        tbUserId.Text = row.Cells["userId"].Value.ToString();
        tbProId.Text = row.Cells["proId"].Value.ToString();
        tbQuantity.Text = row.Cells["orQuantity"].Value.ToString();
        tbPrice.Text = row.Cells["price"].Value.ToString();
        dateTimePicker1.Text = row.Cells["orDate"].Value.ToString();
        cbStatus.Text = row.Cells["Status"].Value.ToString();*/
        tbOrderId.Text = row.Cells["orderId"].Value.ToString();
        tbUserId.Text = row.Cells["userName"].Value.ToString();
        tbProId.Text = row.Cells["proName"].Value.ToString();
        tbQuantity.Text = row.Cells["orQuantity"].Value.ToString();
        tbPrice.Text = row.Cells["price"].Value.ToString();
        dateTimePicker1.Text = row.Cells["orDate"].Value.ToString();
        cbStatus.Text = row.Cells["Status"].Value.ToString();
    }
}

//reference
private void loadGridView()
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    //String qry = "select * from pOrder";
    //String qry1 = "select * from pOrder where userId = '"+LoginForm.userId+"'";
    String qry = "select * from newOrder";
    String qry1 = "select * from newOrder where userName = '"+LoginForm.nameDisplay+"'";
    SqlDataAdapter sda = new SqlDataAdapter();
    DataTable dt = new DataTable();
    if (LoginForm.role == "1")
    {
        {
            sda = new SqlDataAdapter(qry, conn);
            sda.Fill(dt);
            dataGridView1.DataSource = dt;
        }
    }
    else if (LoginForm.role == "2")
    {
        {
            sda = new SqlDataAdapter(qry1, conn);
            sda.Fill(dt);
            dataGridView1.DataSource = dt;
        }
    }
}

```

When select a order from order table, users can see all the information on the textbox at their left screen. For Customer, they just only see their order but with admin they can see all the order of all customers.

```

private void btEdit_Click_1(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    //String query = "update pOrder set Status='" + cbStatus.Text + "'where orId = '" + tbOrId.Text + "'";
    String query = "update newOrder set Status='" + cbStatus.Text + "'where orId = '" + tbOrId.Text + "'";
    SqlCommand cmd = new SqlCommand(query, conn);
    try
    {
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Update success");
        cbStatus.Enabled = false;
        btEdit.Visible = false;
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

//reference
private void btChange_Click(object sender, EventArgs e)
{
    if (cbStatus.Text == "")
    {
        MessageBox.Show("Please select a row");
    }
    else
    {
        cbStatus.Enabled = true;
        btEdit.Enabled = true;
    }
}

```

```

private void btDelete_Click(object sender, EventArgs e)
{
    SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
    //String query = "Delete from pOrder where orId = '" + tbOrId.Text + "'";
    String query = "Delete from newOrder where orId = '" + tbOrId.Text + "'";
    SqlCommand cmd = new SqlCommand(query, conn);
    try
    {
        conn.Open();
        cmd.ExecuteNonQuery();
        MessageBox.Show("Deleted");
        conn.Close();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

```

Both edit and delete button will just appear for admin to make change. Each button will do different function base on their name such add make change of the order and remove that order form database.

```

private void btClose_Click(object sender, EventArgs e)
{
    this.Close();
}

1 reference
private void btCancelOrder_Click(object sender, EventArgs e)
{
    DialogResult dr = MessageBox.Show("Do you really want to cancel the order?", "Title", MessageBoxButtons.YesNoCancel, MessageBoxIcon.Information);
    if(dr == DialogResult.Yes)
    {
        String Cancel = "Cancel!";
        SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
        //String query = "update pOrder set Status=" + Cancel + "where orId = " + tbOrId.Text + "";
        String query = "update newOrder set Status=" + Cancel + "where orId = " + tbOrId.Text + "";
        SqlCommand cmd = new SqlCommand(query, conn);
        try
        {
            conn.Open();
            cmd.ExecuteNonQuery();
            MessageBox.Show("Cancel success");
            dataGridView1.Refresh();
        }
        catch (Exception ex)
        {
            MessageBox.Show(ex.Message);
        }
    }
}

```

This form show all the order information of the user. For admin it will show all the order information of all users.

II. Graphical components

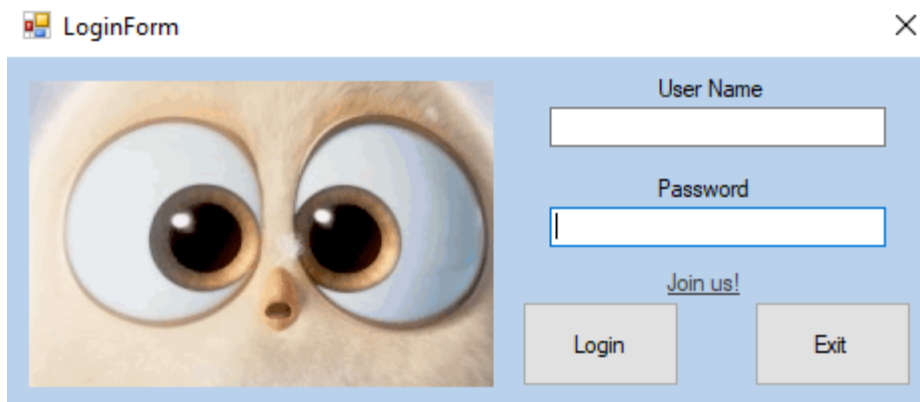
Login Form

```

private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.TextBox UserNameTextBox;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.TextBox PasswordTextBox;
private System.Windows.Forms.Button LoginButton;
private System.Windows.Forms.Button ExitButton;
private System.Windows.Forms.LinkLabel lbJoin;

```

In login form there are 1 picture box, 2 label to user know to enter which text type into 2 text box. There are 2 button one for login and the other to close the program. The last once is linkLabel that allow to click on it.



DashBoard Form

```
private System.Windows.Forms.MenuStrip menuStrip1;
private System.Windows.Forms.ToolStripMenuItem tsmiUserName;
private System.Windows.Forms.ToolStripMenuItem exitToolStripMenuItem;
private System.Windows.Forms.ToolStrip toolStrip1;
private System.Windows.Forms.ListBox listBox1;
private System.Windows.Forms.DataGridView dataGridView1;
private System.Windows.Forms.TextBox textBox1;
private TechShopDataSet techShopDataSet;
private System.Windows.Forms.BindingSource categoryBindingSource;
private TechShopDataSetTableAdapters.CategoryTableAdapter categoryTableAdapter;
private System.Windows.Forms.BindingSource categoryBindingSource1;
private System.Windows.Forms.ToolStripButton tsbDelete;
private System.Windows.Forms.ToolStripButton tsbUpdate;
private System.Windows.Forms.ToolStripButton tsbAdd;
private System.Windows.Forms.ToolStripButton tsbAddNewUser;
private System.Windows.Forms.Button btPurchase;
private System.Windows.Forms.TabControl tabControl1;
private System.Windows.Forms.TabPage tbProducts;
private System.Windows.Forms.Button btManageOrder;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;
private System.Windows.Forms.ToolStripButton tsbAddCategory;
private System.Windows.Forms.ToolStripButton tsbUpdateCategory;
private System.Windows.Forms.ToolStripButton tsbDeleteCategory;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator2;
private System.Windows.Forms.ToolStripMenuItem logoutToolStripMenuItem;
private System.Windows.Forms.LinkLabel llCategoryRefresh;
```



As Dashboard form is main menu and allow user to use almost function. With mean it have 1 list box to show category and 1 gridview to show products. There is a toolStrip that contain 3 two toolStripMenuItem and a menuStrip that contain 9 ToolStrip. There also 2 button and a tabControl which contain a tabPage. About 2 bindingSource, it is used for the ListBox and GridView.

DashboardForm

admin

Add new User
 Add Category
 Update category
 Delete category
 Add
 Update
 Delete

Products

| Product ID | Image | Name | Quantity | Price |
|------------|---|-------|----------|-------|
| 24 |  | ASUS | 1 | 1000 |
| 25 |  | F555L | 1 | 1000 |

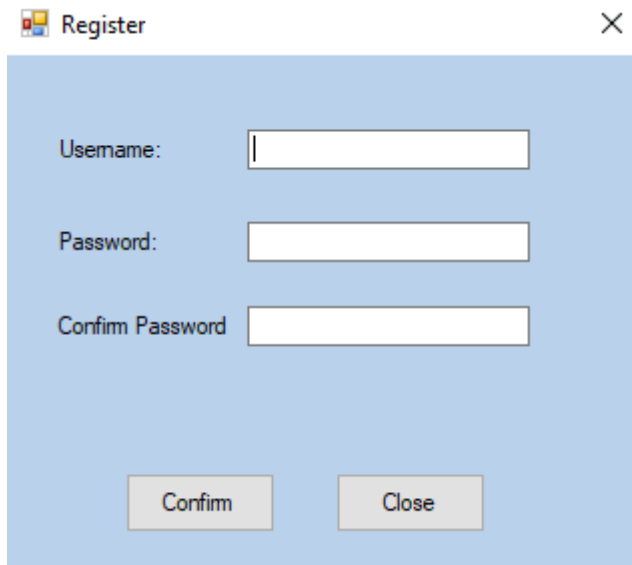
ASUS
Lenovo

Manage Order

AddUser Form

```
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.TextBox tbUserName;
private System.Windows.Forms.TextBox tbPassword;
private System.Windows.Forms.ComboBox cbUserType;
private System.Windows.Forms.Button btAdd;
private System.Windows.Forms.Button btClose;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.TextBox tbConfirmPassword;
```

This form contain 3 label that show direction to user know what to enter in 3 textbox. There also 2 button.

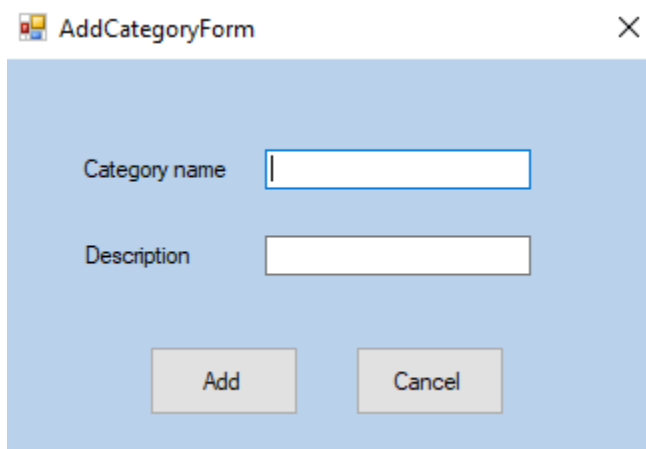


A screenshot of a Windows application window titled "Register". The window has a light blue background and a standard Windows title bar with a close button (X) in the top right corner. Inside the window, there are three text input fields arranged vertically. The first field is labeled "Username:", the second "Password:", and the third "Confirm Password". Below the text fields, there are two buttons: "Confirm" on the left and "Close" on the right.

AddCategory Form

```
private System.Windows.Forms.Label label1;  
private System.Windows.Forms.Label label2;  
private System.Windows.Forms.TextBox tbCategoryName;  
private System.Windows.Forms.TextBox tbDescription;  
private System.Windows.Forms.Button btAddCategory;  
private System.Windows.Forms.Button btCancel;  
private System.Windows.Forms.Button btUpdate;
```

This form contain 2 label that show direction to user know what to enter in 2 textbox. There also 3 button.

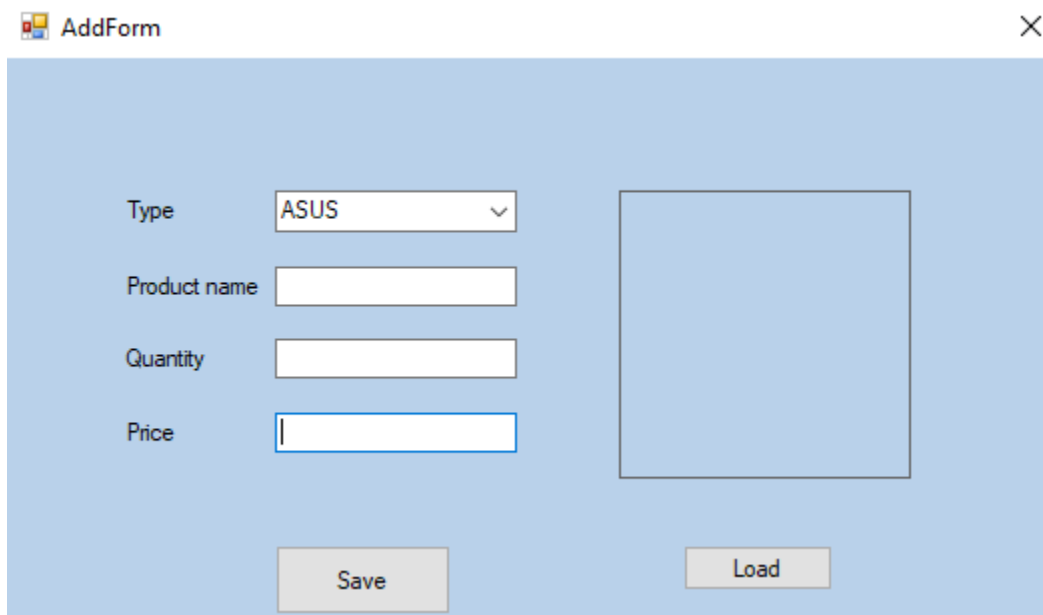


A screenshot of a Windows application window titled "AddCategoryForm". The window has a light blue background and a standard Windows title bar with a close button (X) in the top right corner. Inside the window, there are two text input fields arranged vertically. The first field is labeled "Category name" and the second "Description". Below the text fields, there are two buttons: "Add" on the left and "Cancel" on the right.

Add Form (For product)

```
private System.Windows.Forms.Label label1;  
private System.Windows.Forms.TextBox tbProName;  
private System.Windows.Forms.ComboBox cbType;  
private System.Windows.Forms.Label label2;  
private System.Windows.Forms.Label label3;  
private System.Windows.Forms.Label label4;  
private System.Windows.Forms.PictureBox pictureBox1;  
private System.Windows.Forms.Button btLoad;  
private System.Windows.Forms.Button btSave;  
private System.Windows.Forms.Button btUpdate;  
private TechShopDataSet techShopDataSet;  
private TechShopDataSetTableAdapters.CategoryTableAdapter categoryTableAdapter;  
private System.Windows.Forms.BindingSource categoryBindingSource;  
private System.Windows.Forms.TextBox tbQuantity;  
private System.Windows.Forms.TextBox tbPrice;  
private System.Windows.Forms.TextBox tbImageLoad;
```

This form contain 4 label that show direction to user know what to enter in 3 textbox and what is in the comboBox. There also 3 button 2 to perform that action and one other is to load image into the picturebox.



The screenshot shows a Windows application window titled "AddForm" with a close button (X) in the top right corner. The window has a light blue background. On the left side, there are four labels: "Type", "Product name", "Quantity", and "Price". Each label is followed by a corresponding input field. The "Type" field is a dropdown menu with "ASUS" selected. The "Product name", "Quantity", and "Price" fields are text boxes. To the right of these input fields is a large, empty rectangular area, likely a placeholder for an image. At the bottom of the window, there are two buttons: "Save" and "Load".

Order Form

```
private System.Windows.Forms.DataGridView dataGridView1;  
private System.Windows.Forms.TextBox tbOrId;  
private System.Windows.Forms.Label label1;  
private System.Windows.Forms.TextBox tbUserId;  
private System.Windows.Forms.Label label2;  
private System.Windows.Forms.TextBox tbProId;  
private System.Windows.Forms.Label label3;  
private System.Windows.Forms.TextBox tbQuantity;  
private System.Windows.Forms.Label label4;  
private System.Windows.Forms.DateTimePicker dateTimePicker1;  
private System.Windows.Forms.Label label5;  
private System.Windows.Forms.TextBox tbPrice;  
private System.Windows.Forms.Label label6;  
private System.Windows.Forms.Label label7;  
private System.Windows.Forms.ComboBox cbStatus;  
private System.Windows.Forms.Button btClose;  
private System.Windows.Forms.Button btDelete;  
private System.Windows.Forms.Button btEdit;  
private System.Windows.Forms.Button btChange;  
private System.Windows.Forms.Button btCancelOrder;
```

This form contain 7 label that show direction to user know what is in 5 textbox, DateTimePicker, and ComboBox. There also 5 button and a GridView.

| | orId | userName | catName | proName | orQuantity | price |
|---|------|----------|---------|---------|------------|-------|
| ▶ | 1 | admin | ASUS | ASUS | 1 | 1000 |
| | 2 | trung3 | ASUS | F555L | 1 | 1000 |
| | 3 | trung3 | ASUS | F555L | 1 | 1000 |
| * | | | | | | |

Task 3

First, we need to differentiate the difference between the two types of applications: Windows and the Web. When the first Web application was created, one would distinguish the two types of applications from the following: The Windows app runs on the Desktop or on a local-Area network, while the WEB application is run on a remote Server and accessed using a Web browser (Web browser). This difference is no longer obvious because Windows apps now tend to use WEB services. For example, Outlook software converts the mail through a WEB connection.

Desktop-based application is an application that runs stand-alone in a desktop or laptop computer. Different than "Web-based application," which requires the Web browser to run. The Desktop-based application runs on a PC operating system such as Windows, Mac, Linux, etc. it has a graphical user interface and it does not run inside a web browser. The term that may be used to contrast desktop applications with mobile applications that run in smartphones and tablets. Between the Windows and the Web application which one owns the UI?

In my opinion, the application uses the browser to display or the application is built into the program that can run on the Desktop. There are some advantages to the WEB application, the application can be accessed by any browser that connects to the Server, the calibration is done on the Server, there is no need to distribute the new dynamic link library (DynamicLinkLibraries-DLLs) needed to run the application for Users. Dot NET also has this distinguished, typically having the appropriate toolkits for each type of application: Windows or Web. Both types are based on the Form template and use the controls (Control) such as Buttons, ListBox, and Text... The current Windows and WEB application has many similarities, and .NET should be bundled into a generic toolkit for both Windows and WEB applications in the coming version.

Web App is a Web-based application that performs a number of basic functions such as sharing images, computing, shopping... That you can easily reach through the Internet or the Internet. Applications can be as simple as a message board or complex as a multiplayer gaming app. These apps are becoming more popular in these days because they don't require user to installation, manual updates, or hardware upgrades.

Mobile APP is a mobile application that allows you to access the content you desire on devices such as mobile phones.

In this scenario, Windows Form is chosen for develop the application. It gives user easier access to the software and its functionalities. Moreover, a desktop is the most stable among the three types of delivery environments.

To exchange data with database, Technology [ADO.NET](#) are used. It is the model that allows programmers to manipulate multiple types of databases in the same way. DATABASES can be SQL Server, MS Access, Oracle ... Each DB type will be the appropriate .NET Framework Data Provider. Includes objects of the Data Provider classes used to manipulate the DATABASE

To start connection to database you would have to follow these code:

First enter this code to access the sql command data:

```
using System.Data.SqlClient;
```

Then enter this code inside the method to connect into the database of Microsoft SQL server:

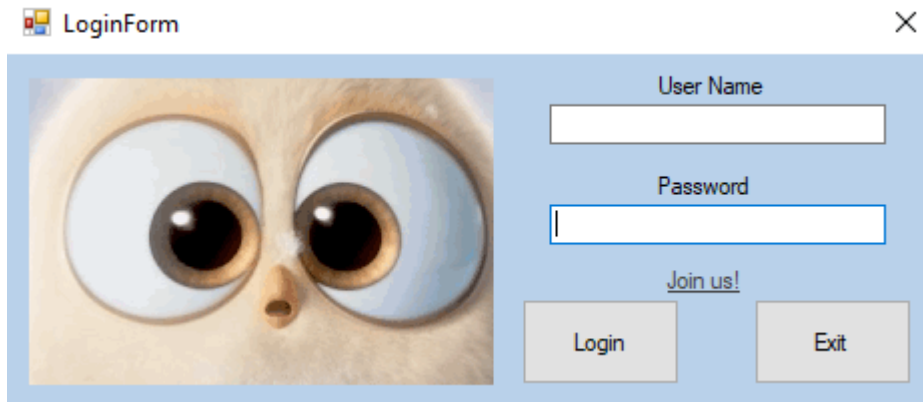
```
SqlConnection conn = new SqlConnection(@"Data Source=trung;Initial Catalog=TechShop;Integrated Security=True");
```

Task 4

I. Program screen

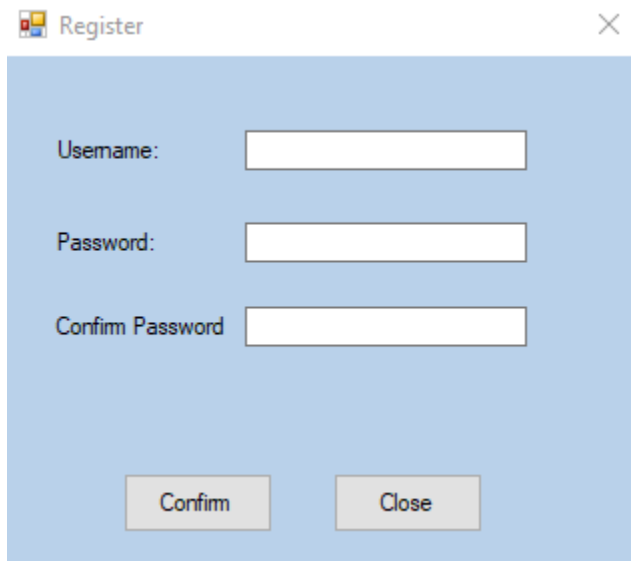
Login Form

First when run the program, user would have to enter their account which is user name and password to go further. If they does not have an account yet they can register by click on Join us!.



Register Form

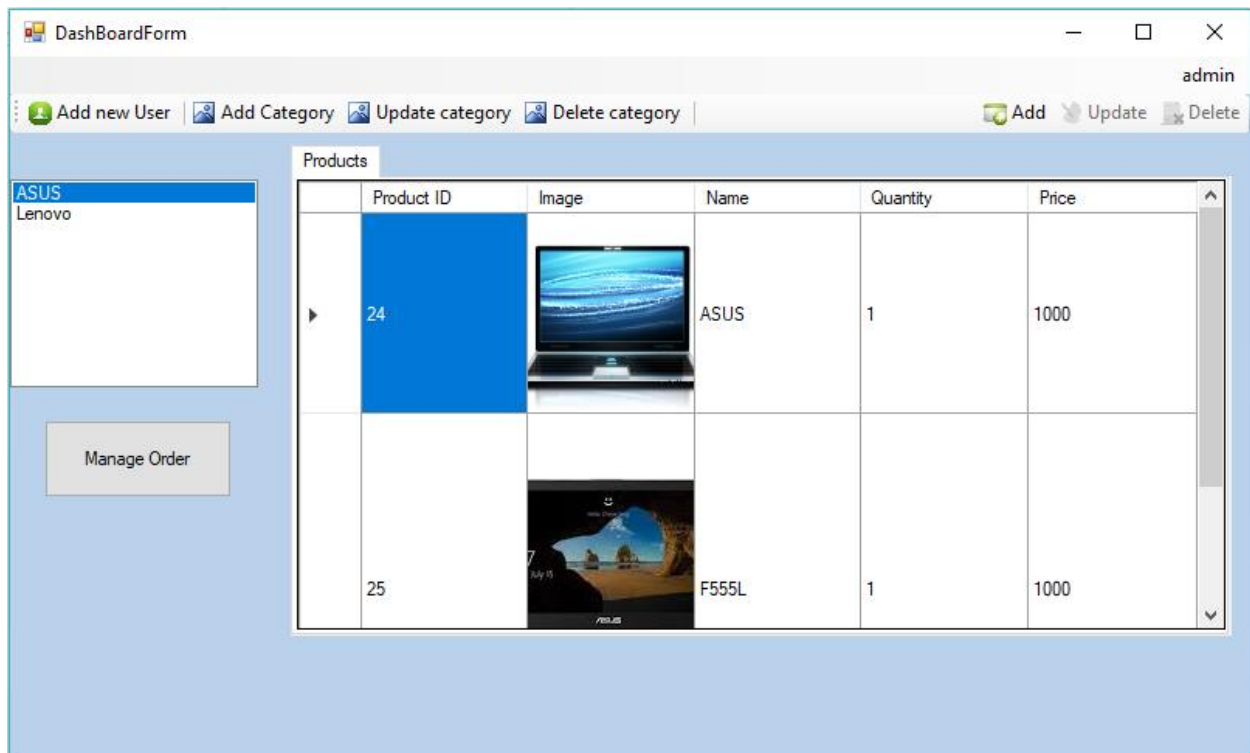
User would have to input user name, password and re-type the password then select confirm. After have an account they can enter it at login screen to move to main menu



DashBoard Form

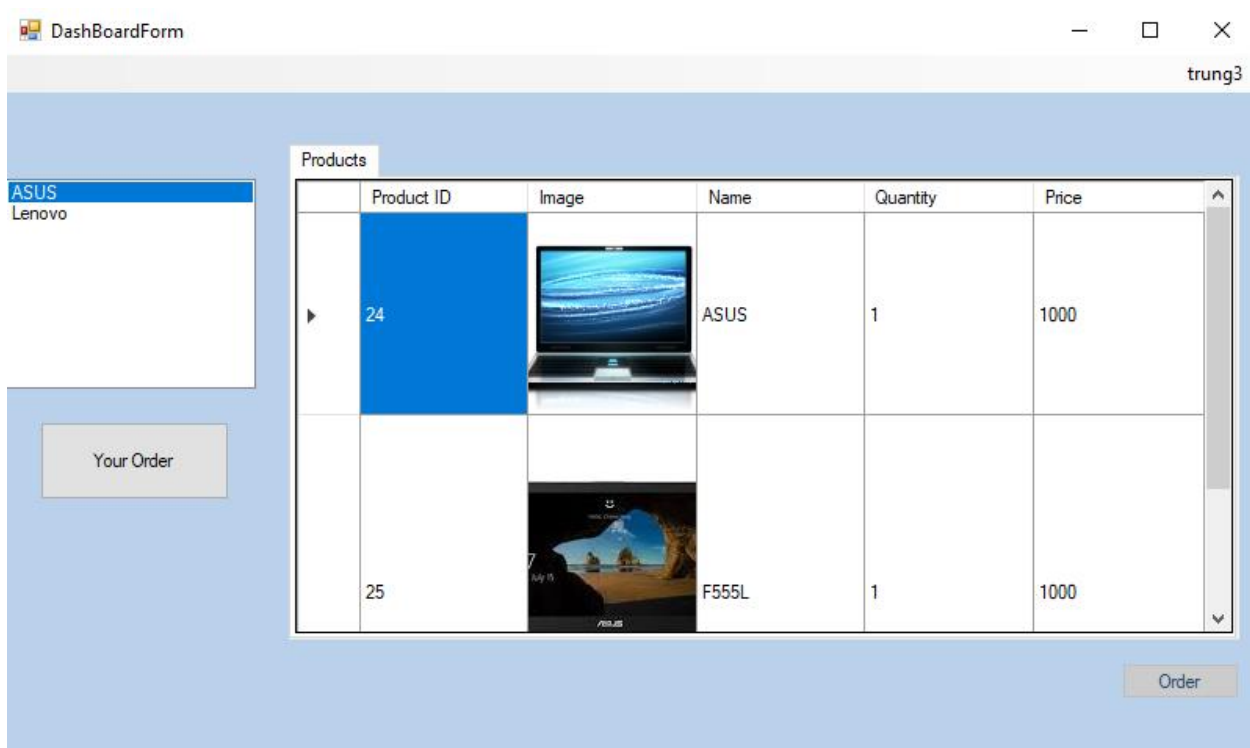
Known as Main menu, this form will allow user to see Categories and products, also do some action with some button. The button will appear depend on the account role. There are 2 role Admin and Customer.

For admin:



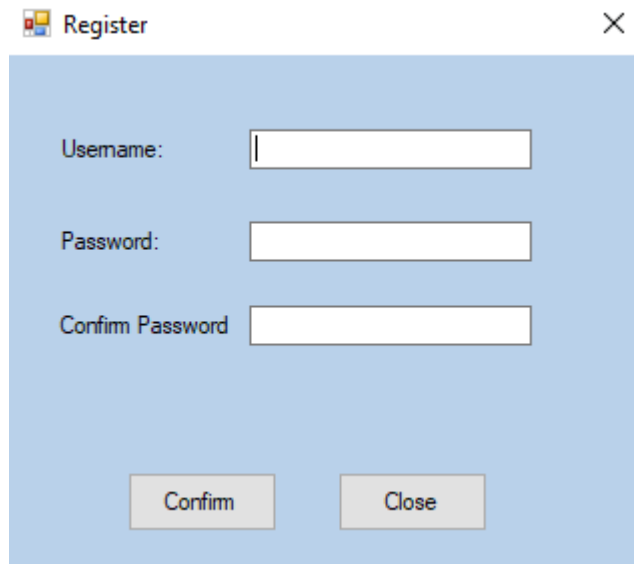
Admin are allow to make change to category and product. They can add new, change information and also remove them.

For Customer



The customer are allow to make order from the product they choose.

AddUser Form



A Windows-style dialog box titled "Register" with a close button (X) in the top right corner. The dialog has a light blue background. It contains three text input fields: "Username:", "Password:", and "Confirm Password:". Below the fields are two buttons: "Confirm" and "Close".

Register

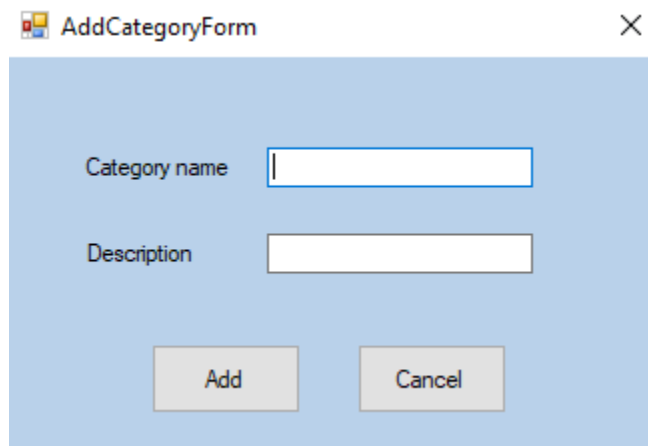
Username:

Password:

Confirm Password

Confirm Close

AddCategory Form



A Windows-style dialog box titled "AddCategoryForm" with a close button (X) in the top right corner. The dialog has a light blue background. It contains two text input fields: "Category name" and "Description". Below the fields are two buttons: "Add" and "Cancel".

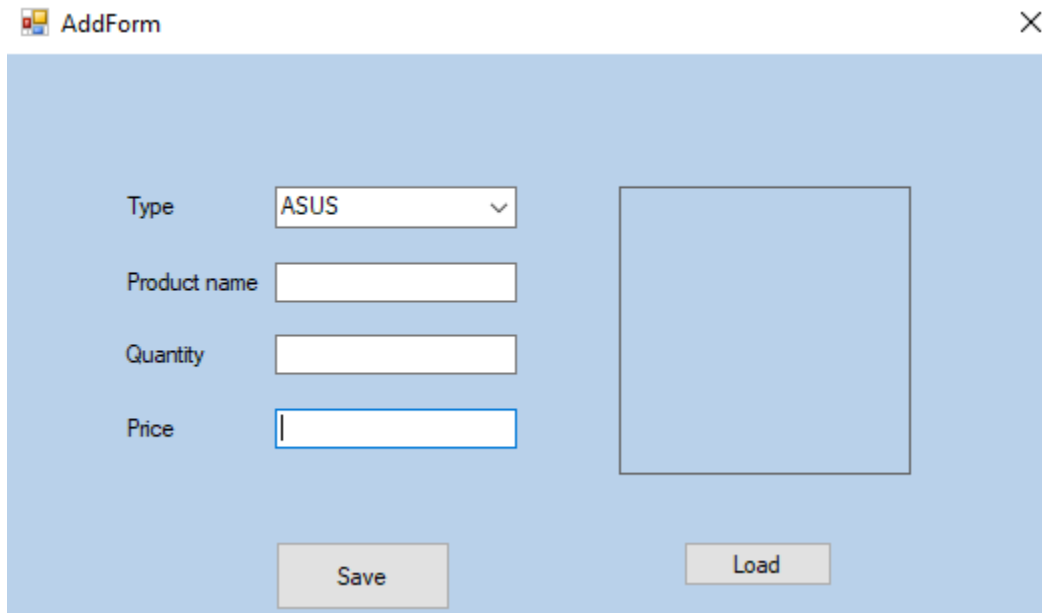
AddCategoryForm

Category name

Description

Add Cancel

Add Form (For product)

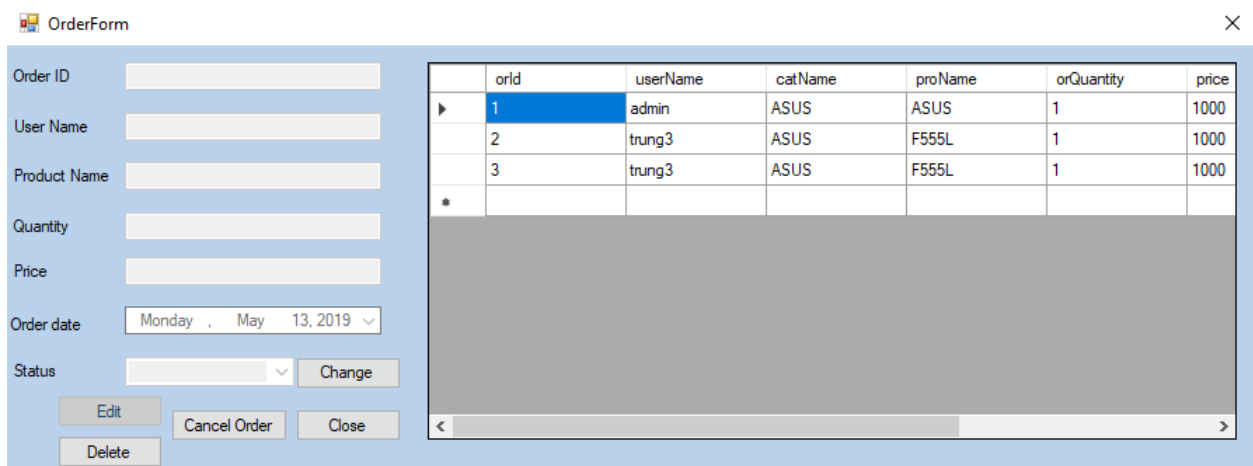


The AddForm window is a light blue dialog box with a title bar containing a small icon and the text "AddForm" and a close button (X). The main area contains four input fields on the left: "Type" with a dropdown menu showing "ASUS", "Product name" with a text box, "Quantity" with a text box, and "Price" with a text box. To the right of these fields is a large empty square box. At the bottom are two buttons: "Save" and "Load".

Order Form

The order form are available for both admin and customer but it will show different.

For admin



The OrderForm window for admin is a light blue dialog box with a title bar containing a small icon and the text "OrderForm" and a close button (X). The main area is divided into two sections. On the left is a form with the following fields: "Order ID" (text box), "User Name" (text box), "Product Name" (text box), "Quantity" (text box), "Price" (text box), "Order date" (date picker showing "Monday . May 13, 2019"), and "Status" (dropdown menu with a "Change" button). Below these fields are three buttons: "Edit", "Cancel Order", and "Delete". On the right is a table with the following columns: "orId", "userName", "catName", "proName", "orQuantity", and "price". The table contains three rows of data. The first row is highlighted in blue. Below the table is a large grey rectangular area.

| | orId | userName | catName | proName | orQuantity | price |
|---|------|----------|---------|---------|------------|-------|
| ▶ | 1 | admin | ASUS | ASUS | 1 | 1000 |
| | 2 | trung3 | ASUS | F555L | 1 | 1000 |
| | 3 | trung3 | ASUS | F555L | 1 | 1000 |
| * | | | | | | |

Admin can see all the orders and also make change of their status.

For customer

The OrderForm application window displays a form on the left for entering order details and a table on the right showing existing orders. The form includes fields for Order ID, User Name, Product Name, Quantity, Price, Order date (set to Tuesday, May 14, 2019), and Status. There are 'Cancel Order' and 'Close' buttons at the bottom. The table has columns for orld, userName, catName, proName, orQuantity, and price. It lists two orders for user 'trung3' with product 'ASUS F555L' at a price of 1000 each.

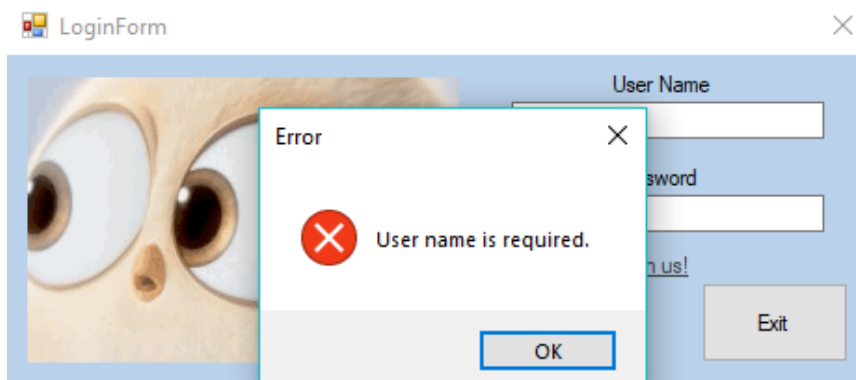
| | orld | userName | catName | proName | orQuantity | price |
|---|------|----------|---------|---------|------------|-------|
| ▶ | 2 | trung3 | ASUS | F555L | 1 | 1000 |
| | 3 | trung3 | ASUS | F555L | 1 | 1000 |
| * | | | | | | |

The customer can see all of their orders and also cancel the order.

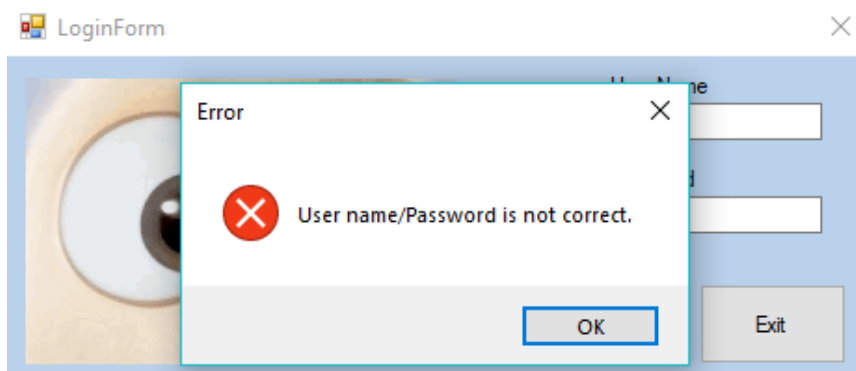
II. Error handling

In login form:

When user input user name and password, if one of the username or password are blank message box will show up and tell them to input it.



Also if the account user input does not exist or they enter wrong password, message box will show up.



Register form:

When user try to register if they account already exist the message box will appear. Also as they have to enter the password twice to make sure both are the same but if one of those are not the same the message box will appear

