

CAFÉ MANAGEMENT

(ระบบจัดการ – ดูแลของใน Café)

เนื้อหา

0. ข้อมูลเบื้องต้นสำหรับโปรเจค.....	ค
1. บทนำ และที่มาของโปรเจค	1
2. รายละเอียดของโปรแกรม	2
ส่วนที่ 1: Login Form (ส่วนของการเข้าถึง)	2
1. คำอธิบาย:	2
2. โปรแกรม (Code):	4
3. หลักการทำงานของโปรแกรม	6
ส่วนที่ 2: Dash Board (ส่วนของการใช้งาน)	7
1. คำอธิบาย:	8
2. โปรแกรมหลัก (Code) :	9
3. หลักการทำงานของโปรแกรม	13
3. ฐานข้อมูล (Database).....	14
1. โครงสร้างข้อมูล (Structure of Data):	14
2. ฐานข้อมูล (Database):	15
3. คลาส (Function) ที่ใช้ในการทำงานร่วมกับฐานข้อมูล:	16
4. หน้าจอใช้งานร่วม (User Control)	17
ส่วนที่ 1: Place Order (สั่ง - ออกบิล รายการอาหาร)	17
โปรแกรม (Code):	18
ส่วนที่ 2: Add Items (เพิ่มข้อมูลรายการอาหาร)	24
โปรแกรม (Code):	25
ส่วนที่ 3: Update Items (อัปเดต / เขียนทับข้อมูลลงไป)	26
โปรแกรม (Code):	27
ส่วนที่ 4: Remove Items (ลบข้อมูลออกจากฐานข้อมูล)	29
โปรแกรม (Code):	30

สารบัญภาพ

ภาพที่ 1 : โปรแกรม และรูปแบบของโปรแกรม	1
ภาพที่ 2.1.1. : Login Form	2
ภาพที่ 2.1.2. : รูปแบบการกรอกรหัส	3
Flowchart สำหรับโปรแกรมส่วน 1	6
ภาพที่ 2.2.1. : Dash Board (Admin Login)	7
ภาพที่ 2.2.2. : Dash Board (Guest Login)	7
ภาพที่ 2.3.1. : การทำงานของปุ่มเปลี่ยนสี	13
ภาพที่ 3.2.1 : ลักษณะตาราง	15
ภาพที่ 3.3.1 : เชื่อมต่อกับฐานข้อมูล	16
ภาพที่ 3.3.2: คำสั่ง getData สำหรับ Return ข้อมูลมา	16
ภาพที่ 3.3.3 : คำสั่ง setData สำหรับแก้ไขตาราง	16
ภาพที่ 4.1.1 : หน้าจอ Place Order	17
ภาพที่ 4.1.2 : การกรอกรหัสในการเลือกอาหาร	19
ภาพที่ 4.1.3 : การเลือกอาหาร และมีข้อมูลปรากฏ	19
ภาพที่ 4.1.4 : การกด Add to Cart	22
ภาพที่ 4.1.5 : ตัวอย่างการพิมพ์ใบเสร็จ	23
ภาพที่ 4.2.1 : หน้าจอ Add Items	24
ภาพที่ 4.3.1 : หน้าจอ Update Item	26
ภาพที่ 4.4.1 : หน้าจอ Remove Items	29
ภาพที่ 4.4.2 : หน้าต่างแจ้งเตือน เมื่อมีการกดเซลล์ลบ	29

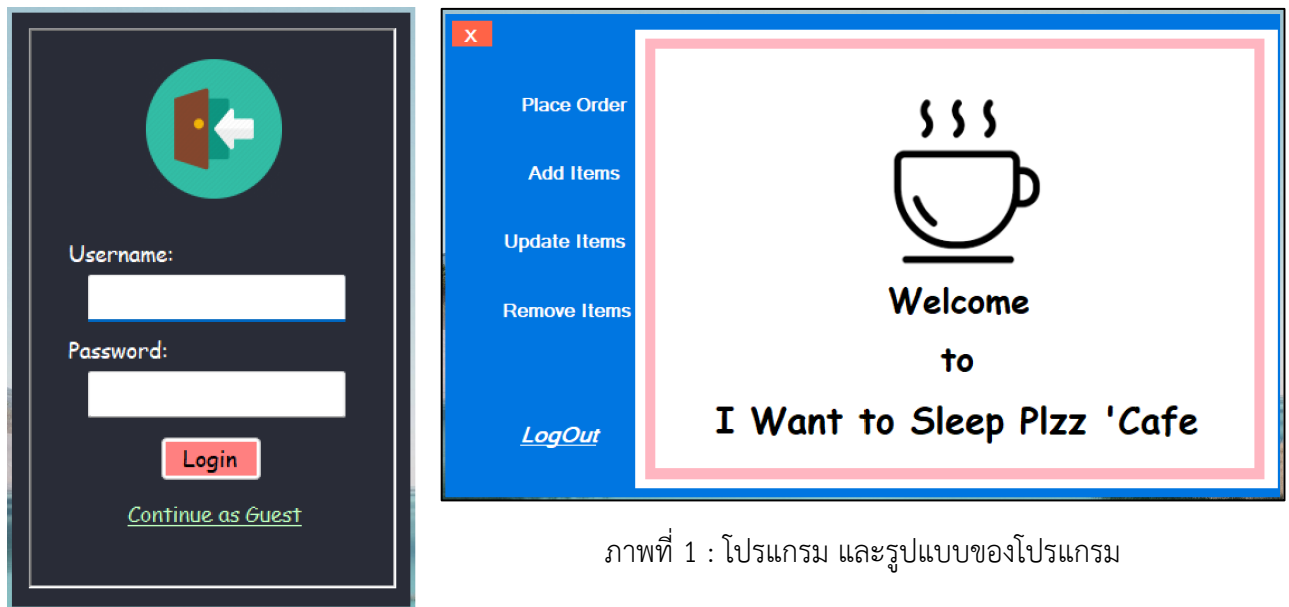
0. ข้อมูลเบื้องต้นสำหรับโปรเจก

- ชื่อโปรเจก : Café Management
- จัดทำโดย : นายธนาท กุลพันธ์ (รหัสนิสิต 6530182121 ภาควิศวกรรมอุตสาหกรรม)
- รายวิชา : 2104255 (Application Development for Industrial Engineering)
- ปีการศึกษา : ภาคต้น ปีการศึกษา 2566 ชั้นปีที่ 2
- ภาษาที่ใช้เขียนโปรแกรม : C# และ SQL (สำหรับสร้างฐานข้อมูล)
- โปรแกรมที่ใช้งาน : Microsoft Word
Visual Studio 2022
SQL Server Management 2019
- จุดประสงค์ของโปรเจก :
 - เพื่อพัฒนา และสร้างระบบการจัดการรายการอาหารในร้านกาแฟ
 - เพื่อนำความรู้การเขียนโปรแกรม C# พื้นฐาน มาพัฒนาเป็นโปรแกรมเบื้องต้น
 - ออกแบบ และปรับปรุงดีไซน์ของโปรแกรมด้วยตนเอง
- ระยะเวลาดำเนินการ : ประมาณ 1 เดือน
- ช่องทางต่าง ๆ (เพิ่มเติม)

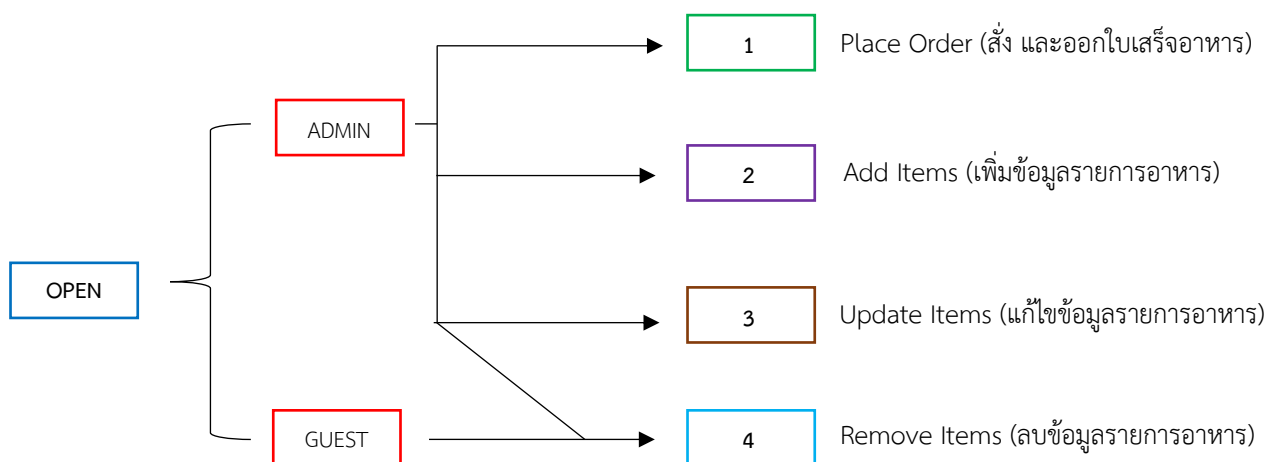
หัวข้อ	หัวข้อย่อย (ถ้ามี)	ลิงค์
Google drives (รวม)	-	[ลิงค์ : Google Drives]
DGV Printer	-	[ลิงค์ : DGV Printer]
คลิปวิธีการใช้งาน	การล็อกอิน	[ลิงค์ Clip : Login]
	Place Order	[ลิงค์ Clip : Place Order]
	Add Items	[ลิงค์ Clip : Add Items]
	Update Items	[ลิงค์ Clip : Update Items]
	Remove Items	[ลิงค์ Clip : Remove Items]

1. บทนำ และที่มาของโปรเจค

ในยุคปัจจุบัน, ค่านิยมสมัยใหม่ของคนไทยส่วนหนึ่ง มีความนิยมในการไปเที่ยวหรือเยี่ยมชมร้านค้ากาแฟตามสถานที่ต่าง ๆ จำนวนมาก อาจมีจุดประสงค์อื่น ๆ ประกอบด้วย อาทิ การนัดเจอผู้คนตามสถานที่, การประชุม, การหาที่พักอ่านหนังสือ หรือترنتตามกระแสสังคม เป็นต้น ซึ่งในบริเวณใกล้เคียงมหาวิทยาลัยนั้นก็มีมากมายเช่นกันตามห้างสรรพสินค้า, ดิกซ์ซ่า หรือแม้กระทั่งในมหาวิทยาลัยเอง



ภาพที่ 1 : โปรแกรม และรูปแบบของโปรแกรม



ในโปรเจกนี้ได้จัดทำ ระบบการจัดการร้านค้ากาแฟ ซึ่งมีระบบอยู่ 4 หมวดหมู่ ได้แก่ 1.การรับออเดอร์สินค้า และออกใบเสร็จ, 2.การเพิ่มรายการอาหาร, 3.การเปลี่ยนแปลงข้อมูลรายการอาหาร และ 4.การลบข้อมูลรายการอาหาร ซึ่งเป็นระบบพื้นฐานสำหรับการให้บริการภายในร้านทั่ว ๆ ไป โดยมีระบบการเข้าใช้งาน (Login) เพื่อใช้จำแนกรูปแบบการใช้งานอีกด้วย

2. รายละเอียดของโปรแกรม

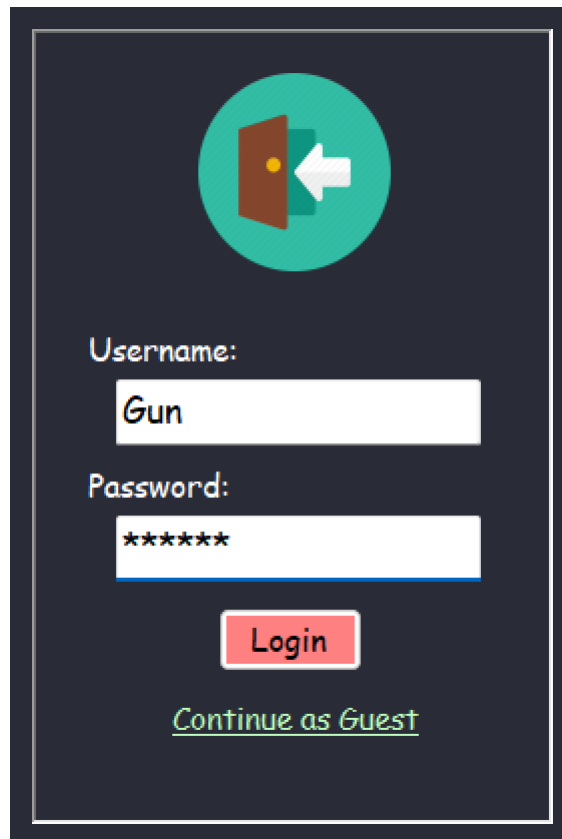
ส่วนที่ 1: Login Form (ส่วนของการเข้าถึง)

The image shows a login form with a dark blue background. At the top center is a circular logo with a teal background, containing a brown door icon and a white arrow pointing left. Below the logo, the text "Username:" is followed by a white text input box. Below that, the text "Password:" is followed by another white text input box. At the bottom center is a red "Login" button. Below the button is a green link that says "Continue as Guest". Four labels with arrows point to specific parts of the form: "username_textbox" points to the username input box, "password_textbox" points to the password input box, "Login_Button_Click" points to the Login button, and an unlabeled arrow points to the "Continue as Guest" link.

ภาพที่ 2.1.1. : Login Form

1. คำอธิบาย:

ในส่วนแรกหลังจากเริ่มทำงานโปรแกรม จะเปิดหน้าต่างเพื่อทำการ Login เข้าใช้งาน ซึ่งสามารถใช้งานได้ 2 รูปแบบ คือ 1. เข้าผ่าน Username ที่มีในระบบ และ 2. เข้าในฐานะ Guest ซึ่งจะแตกต่างกันที่แบบที่ 2. สามารถใช้ระบบ Place Order (สั่ง และออกใบเสร็จสินค้า) ได้อย่างเดียว ในขณะที่แบบที่ 1. จะเป็นรูปแบบของ Admin ซึ่งสามารถเข้าถึงระบบแก้ไขข้อมูลต่าง ๆ ของรายการสินค้าได้ด้วย โดยแต่ละระบบจะอธิบายในส่วนที่ 2



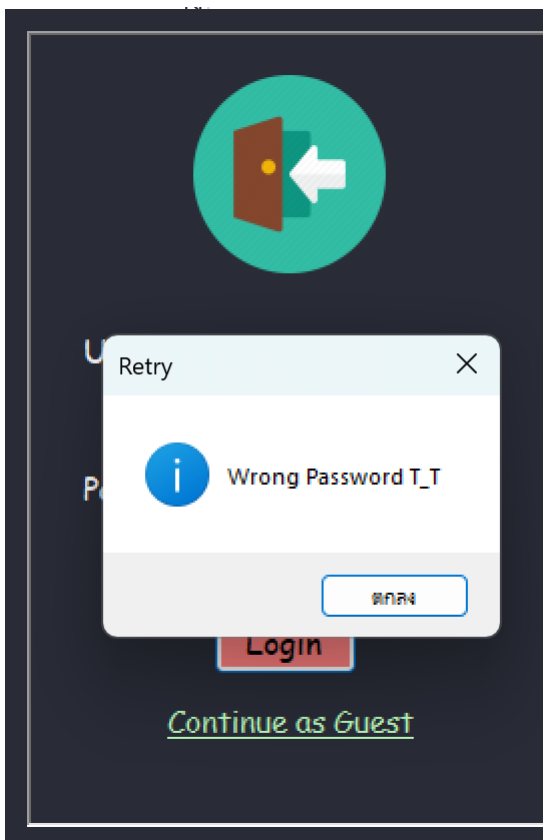
Username:
Gun

Password:

Login

[Continue as Guest](#)

เมื่อกรอกรหัสผิด



Retry

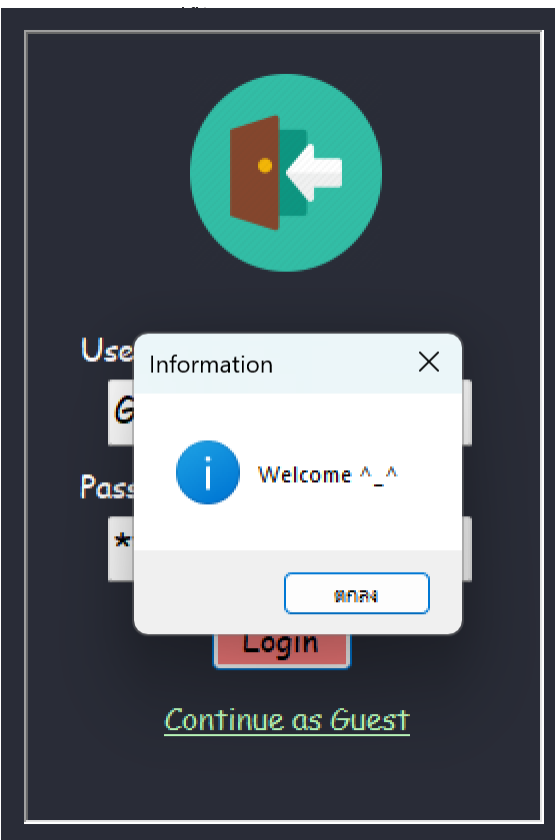
Wrong Password T_T

ตกลง

Login

[Continue as Guest](#)

เมื่อกรอกรหัสถูก



Information

Welcome ^_^

ตกลง

Login

[Continue as Guest](#)

ภาพที่ 2.1.2. : รูปแบบการกรอกรหัส

2. โปรแกรม (Code):

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Net.Sockets;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
```

```
namespace _APP_DEV_PROJECT__CAFE
{
```

```
    public partial class Form1 : Form
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
        }
```

```
        //USERNAME & PASSWORD
```

```
        IDictionary<string, string> Username = new Dictionary<string, string>()
```

```
        {
```

```
            {"Gun", "123456"},
```

```
            {"A", "000" },
```

```
            {"B", "1234" }
```

```
        };
```

สามารถเพิ่ม Username และ Password
เพื่อใช้สำหรับเข้าแบบ Admin ได้

```
        //LOGIN WITH GUEST
```

```
        private void LinkLabel_LinkClicked(object sender,
```

```
        LinkLabelLinkClickedEventArgs e)
```

```
        {
```

```
            MessageBox.Show("Welcome ^_^", "Information",
```

```
            MessageBoxButtons.OK, MessageBoxIcon.Information);
```

```
            Dashboard ds = new Dashboard("Guest");
```

```
            ds.Show();
```

```
        }
```

เข้าแบบ Guest (แบบสอง)

```

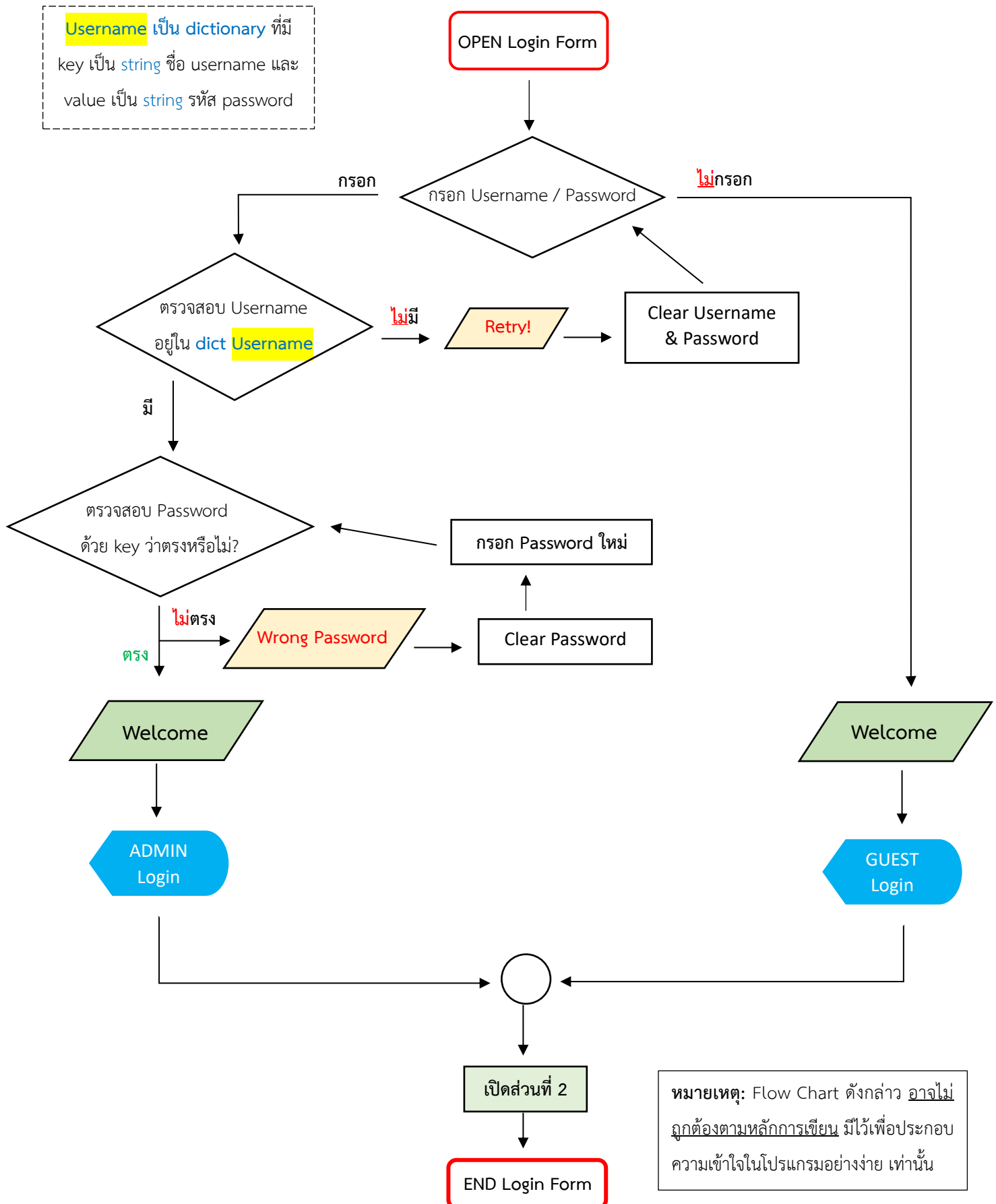
//LOGIN WITH USERNAME
private void Login_Button_Click(object sender, EventArgs e)
{
    -----
    if(username_textbox.Text.Length > 0 &&
    Username.ContainsKey(username_textbox.Text))
    {
        if (password_textbox.Text ==
    Username[username_textbox.Text])
        {
            MessageBox.Show("Welcome ^_^", "Information",
    MessageBoxButtons.OK, MessageBoxIcon.Information);
            Dashboard ds = new Dashboard("Admin");
            ds.Show();
            this.Hide();
        }
        else
        {
            MessageBox.Show("Wrong Password
    T_T", "Retry", MessageBoxButtons.OK, MessageBoxIcon.Information);
            password_textbox.Clear();
        }
    }
    -----
    else
    {
        MessageBox.Show("Error Pls Retry! __*", "Warning",
    MessageBoxButtons.OK, MessageBoxIcon.Error);
        username_textbox.Clear();
        password_textbox.Clear();
    }
}
}
}

```

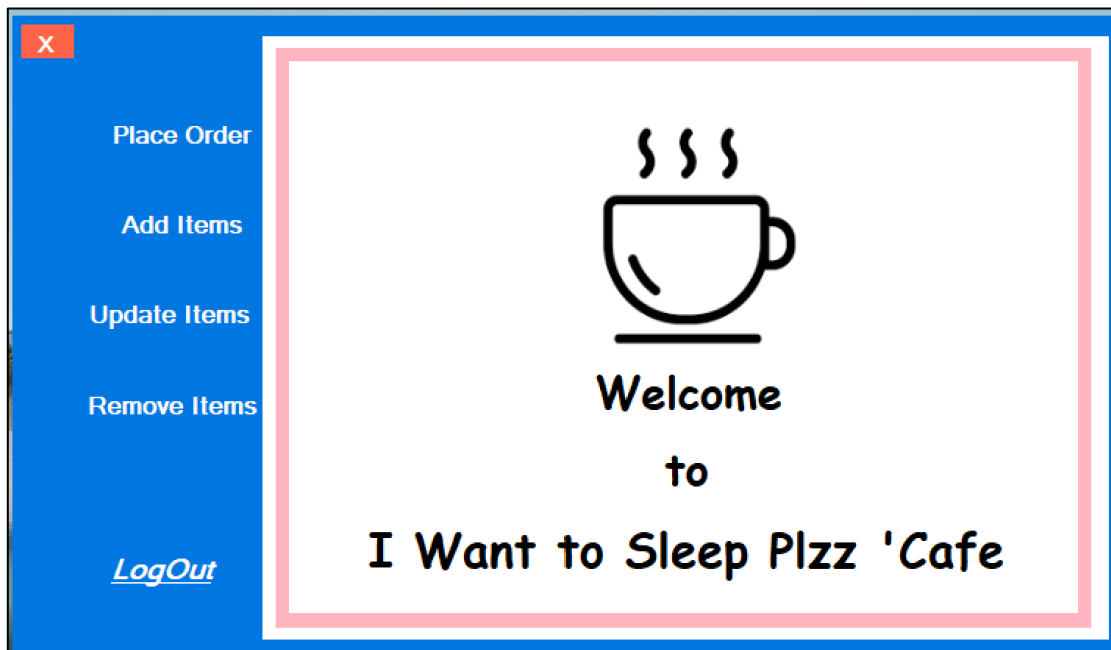
ถ้ามีชื่อในระบบ
เช็ค Password

เข้าแบบแรก

3. หลักการทำงานของโปรแกรม



ส่วนที่ 2: Dash Board (ส่วนของการใช้งาน)



ภาพที่ 2.2.1. : Dash Board (Admin Login)



ภาพที่ 2.2.2. : Dash Board (Guest Login)

1. คำอธิบาย:

ในส่วนนี้จะป็นหน้าจอหลัก สำหรับการใช้งานโปรแกรม ซึ่งรูปแบบการเข้าจะเป็นไปตามการ Login ในส่วนแรก เนื่องจากว่า ในรูปแบบของ Admin มีการทำงานที่มากกว่า และ Place Order ของทั้งสองรูปแบบเหมือนกัน จึงจะอธิบายการทำงานทั้งหมดผ่าน Admin แทน:

โดยมีปุ่มการทำงาน 4 ปุ่ม มีรายละเอียดคร่าว ๆ ดังนี้

1. **Place Order** : สั่งรายการอาหาร, รับ Order ลูกค้า และ Print Bill ออกมาได้
2. **Add Items** : เพิ่มข้อมูลรายการอาหารลงในฐานข้อมูล
3. **Update Items** : แก้ไขข้อมูลรายการอาหารในฐานข้อมูล
4. **Remove Items** : ลบข้อมูลรายการอาหารในฐานข้อมูล

ทั้ง 4 ปุ่มนี้ เมื่อคลิกปุ่มแล้ว จะทำงานผ่าน User Controls ซึ่งจะมีอธิบายทีหลัง (ในบทที่ 4)

2. โปรแกรมหลัก (Code) :

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace _APP_DEV_PROJECT__CAFE
{
    public partial class Dashboard : Form
    {
        public Dashboard()
        {
            InitializeComponent();
        }

        public Dashboard(string name)
        {
            InitializeComponent();

            if (name == "Guest")
            {
                additems.Hide();
                updateitems.Hide();
                removeitems.Hide();
            }

            else if (name == "Admin")
            {
                additems.Show();
                updateitems.Show();
                removeitems.Show();
            }
        }
    }
}

```

ผลจากส่วนที่ 1 : Login Form

```

private void logout_linklabel_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    Form1 fm = new Form1();
    this.Hide();
    fm.Show();
}
//ใช้เปลี่ยนสีปุ่ม เมื่อคลิก
private void change_color_when_click(Button button)
{
    //additems / placeorder / removeitems / updateitems
    //สีพื้นหลัง: #0076E1

    if (button == additems)
    {
        additems.BackColor = Color.White;
        additems.ForeColor = Color.Black;

        placeorder.BackColor = ColorTranslator.FromHtml("#0076E1");
        placeorder.ForeColor = Color.White;
        removeitems.BackColor =
ColorTranslator.FromHtml("#0076E1");
        removeitems.ForeColor = Color.White;
        updateitems.BackColor =
ColorTranslator.FromHtml("#0076E1");
        updateitems.ForeColor = Color.White;
    }
    if (button == removeitems)
    {
        removeitems.BackColor = Color.White;
        removeitems.ForeColor = Color.Black;

        placeorder.BackColor = ColorTranslator.FromHtml("#0076E1");
        placeorder.ForeColor = Color.White;
        additems.BackColor = ColorTranslator.FromHtml("#0076E1");
        additems.ForeColor = Color.White;
        updateitems.BackColor =
ColorTranslator.FromHtml("#0076E1");
        updateitems.ForeColor = Color.White;
    }

    if (button == updateitems)
    {
        updateitems.BackColor = Color.White;
        updateitems.ForeColor = Color.Black;

        placeorder.BackColor = ColorTranslator.FromHtml("#0076E1");

```

```

        placeorder.ForeColor = Color.White;
        removeitems.BackColor =
ColorTranslator.FromHtml("#0076E1");
        removeitems.ForeColor = Color.White;
        additems.BackColor = ColorTranslator.FromHtml("#0076E1");
        additems.ForeColor = Color.White;
    }

    if (button == placeorder)
    {
        placeorder.BackColor = Color.White;
        placeorder.ForeColor = Color.Black;

        additems.BackColor = ColorTranslator.FromHtml("#0076E1");
        additems.ForeColor = Color.White;
        removeitems.BackColor =
ColorTranslator.FromHtml("#0076E1");
        removeitems.ForeColor = Color.White;
        updateitems.BackColor =
ColorTranslator.FromHtml("#0076E1");
        updateitems.ForeColor = Color.White;
    }
}

```

```

//โดยปกติตอนเปิด
private void Dashboard_Load(object sender, EventArgs e)
{
    uC_Additems1.Visible = false;
    uC_PlaceOrder2.Visible = false;
    uC_UpdateItems1.Visible = false;
    uC_RemoveItems1.Visible = false;
}

```

เมื่อเปิดหน้าจอ User
Control อื่น ๆ จะปิด
อยู่ เพื่อให้เห็น

```

private void additems_Click(object sender, EventArgs e)
{
    change_color_when_click(additems);
    uC_Additems1.Visible = true;
    uC_Additems1.BringToFront();
}

private void placeorder_Click(object sender, EventArgs e)
{
    change_color_when_click(placeorder);
    uC_Welcome1.SendToBack();
    uC_PlaceOrder2.Visible = true;
    uC_PlaceOrder2.BringToFront();
}

```



```
private void removeitems_Click(object sender, EventArgs e)
{
    change_color_when_click(removeitems);
    uC_RemoveItems1.Visible = true;
    uC_RemoveItems1.BringToFront();
}

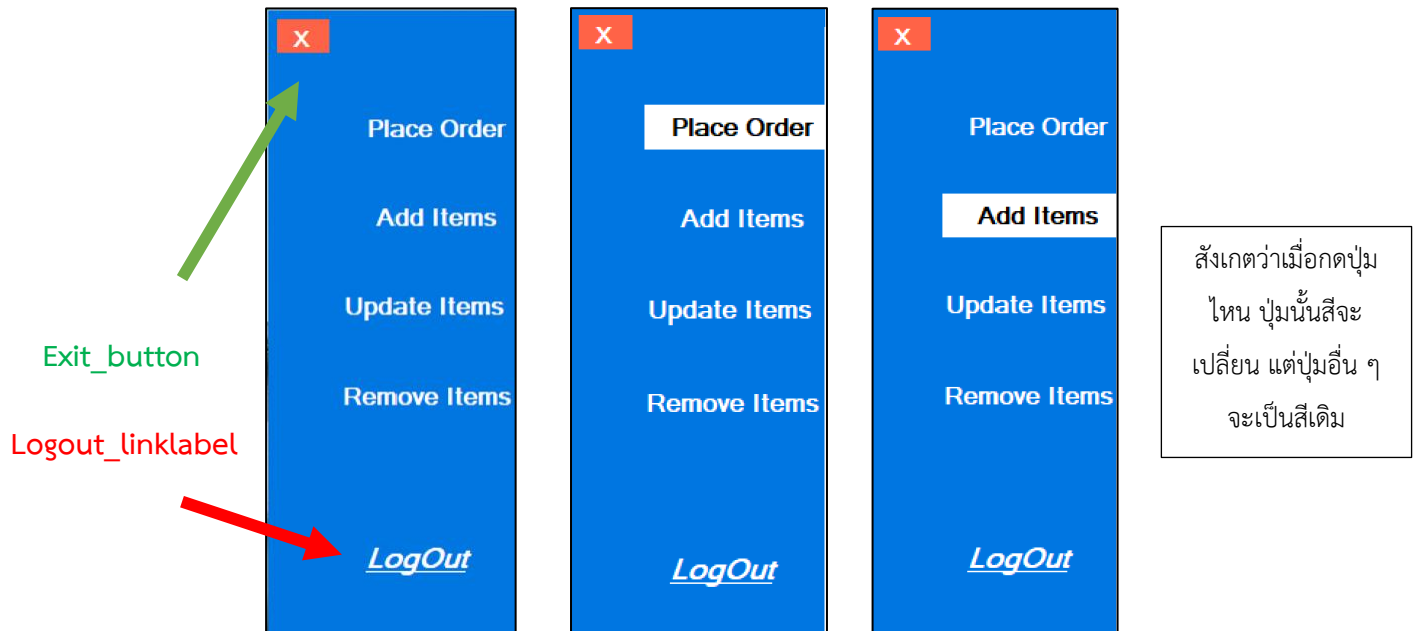
private void updateitems_Click(object sender, EventArgs e)
{
    change_color_when_click(updateitems);
    uC_UpdateItems1.Visible = true;
    uC_UpdateItems1.BringToFront();
}

private void exit_button_Click_1(object sender, EventArgs e)
{
    Application.Exit();
}
}
```

ปุ่มปิดโปรแกรม

3. หลักการทำงานของโปรแกรม

- `change_color_when_click(Button button)` : แทนคำสั่งที่จะรับค่า “ปุ่ม” เพื่อที่ว่า เมื่อกดปุ่มดังกล่าวแล้ว สีของปุ่มจะเปลี่ยนเป็นสีขาว ตัวอักษรสีดำ โดยที่ปุ่มที่เหลือจะเป็นสีพื้นหลัง อักษรสีขาวตามภาพ



ภาพที่ 2.3.1. : การทำงานของปุ่มเปลี่ยนสี

- โดยปกติแล้ว เมื่อเข้ามาในส่วนหน้าจอนี้ จะมีการตั้งให้แสดงให้ User Control ของแต่ละปุ่ม = false อยู่ เพื่อให้เห็นหน้าจอ Welcome ตามภาพ 2.1.1 และ 2.1.2 แต่หากกดปุ่มทั้ง 4 แล้ว จะแสดง User Control เหล่านั้นแทน และสีพื้นหลัง กับสีอักษรจะเปลี่ยนไปจากคำสั่ง `change_color_when_click(Button button)`
- `exit_button_Click_1` : ปุ่มกากบาท ที่จะทำหน้าที่ปิดโปรแกรม
- `logout_linklabel_LinkClicked` : Logout กลับสู่ Form1 (Login Form)

3. ฐานข้อมูล (Database)

1. โครงสร้างข้อมูล (Structure of Data):

เนื่องจากภาษา C# สามารถใช้คำสั่งเพื่อทำงาน ร่วมกับฐานข้อมูลได้ ในที่นี้จึงจะอธิบายเสียก่อนว่า ข้อมูลที่ใช้มีอะไรบ้าง ดังนี้

- **Name** : ชื่อรายการอาหาร (Type: **String**)
- **Category** : ประเภทของอาหาร (Type: **String**)

โดยมีประเภทดังนี้

1. Cake
2. Bakery
3. Meal
4. Soft Drink
5. South Italian
6. Tea & Milk
7. Dessert



Name: Mango Bingsu
Category: Dessert
Price: 80

- **Price** : ราคาของอาหาร (Type: **Int**)

2. ฐานข้อมูล (Database):

ในที่นี้ใช้ภาษา SQL ด้วยโปรแกรม SQL Server Management Studio 19 ทำการสร้างตารางที่ชื่อ **New_table_for_project** ด้วยคำสั่งดังนี้

```
create table New_table_for_project
(
    iid int identity(1,1) primary key,
    names varchar(50) not null,
    category varchar(50) not null,
    price bigint not null
)
```

iid	names	category	price
-----	-------	----------	-------

ภาพที่ 3.2.1 : ลักษณะตาราง

3. คลาส (Function) ที่ใช้ในการทำงานร่วมกับฐานข้อมูล:

เชื่อมต่อฐานข้อมูล

```
internal class Function
{
    2 references
    protected SqlConnection getConnection()
    {
        SqlConnection con = new SqlConnection();
        con.ConnectionString = "data source = ;database = ;integrated security = True";
        return con;
    }
}
```

ภาพที่ 3.3.1 : เชื่อมต่อกับฐานข้อมูล

```
public DataSet getData(String query)
{
    SqlConnection con = getConnection();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = con;
    cmd.CommandText = query;
    SqlDataAdapter da = new SqlDataAdapter(cmd);
    DataSet ds = new DataSet();
    da.Fill(ds);
    return ds;
}
```

ภาพที่ 3.3.2 : คำสั่ง getData สำหรับ Return ข้อมูลมา

query เป็น string ที่ทำหน้าที่รับคำสั่งที่ใช้ในการจัดการตาราง ด้วยจะมีการแปลง string ดังกล่าวเป็นภาษา sql อีกหน เพื่อดึงข้อมูลมา

```
public void setData(String query)
{
    SqlConnection con = getConnection();
    SqlCommand cmd = new SqlCommand();
    cmd.Connection = con;
    con.Open();
    cmd.CommandText = query;
    cmd.ExecuteNonQuery();
    con.Close();

    MessageBox.Show("Data Processed Successfully.",
        "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);
}
```

ภาพที่ 3.3.3 : คำสั่ง setData สำหรับแก้ไขตาราง

หากทำงานเรียบร้อย ให้ขึ้นหน้าต่างเพื่อแจ้งว่า
แก้ไขตารางสำเร็จ

4. หน้าจอใช้งานร่วม (User Control)

หลังกดปุ่ม 1 ใน 4 ปุ่มในส่วนที่ 2 : Dash Board แล้ว จะมีการทำงานที่แตกต่างกันไป 4 รูปแบบ โดยจะกล่าวแต่ละ User Control ดังนี้

ส่วนที่ 1: Place Order (สั่ง - ออกบิล รายการอาหาร)

Place Order

Category:

Search

Place Order

Item Name: Price:

Quantity: Total:

Discount(%) :

Item Name	Unit Price	Quantity	Discount (%)	Price

Grand Total

ภาพที่ 4.1.1 : หน้าจอ Place Order

- วิธีการใช้งาน : ในคลิป หรือผ่านลิงค์ [ลิงค์ Clip: Place Order](#)

โปรแกรม (Code):

1. โปรแกรมหลัก:

```

public partial class UC_PlaceOrder : UserControl
{
    Function fn = new Function();
    String query;
    2 references
    public UC_PlaceOrder()...
    //show ชื่อรายการอาหารทั้งหมดในหมวดหมู
    2 references
    private void showItemList(String query)
    {
        listBox1.Items.Clear();
        DataSet ds = fn.getData(query);

        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            listBox1.Items.Add(ds.Tables[0].Rows[i][0].ToString());
        }
    }
}

```

showItemsList : เป็นฟังก์ชันรับ **string** query

ทำหน้าที่ในการดึงรายชื่อรายการอาหารทั้งหมด หลังจากเลือก Category แล้ว ซึ่งจะปรากฏที่ ListBox

```

//เลือกหมวดหมู่อาหาร แล้วใช้ showItemList ดึงชื่อทั้งหมดออกมา
1 reference
private void comboCatagory_SelectedIndexChanged(object sender, EventArgs e)
{
    String category_ = comboCatagory.Text;
    query = "select names from New_table_for_project where category =" +
        category_ + "'";
    showItemList(query);
}

//ค้นหาชื่อ
1 reference
private void txtSearch_TextChanged(object sender, EventArgs e)
{
    String category_ = comboCatagory.Text;
    query = "select names from New_table_for_project where category =" +
        category_ + "' and names like '" + txtSearch.Text + "%'";
    showItemList(query);
}

```

จุดต่าง โดยจะค้นชื่อเพิ่มเติมด้วย (query 2 ชั้น)

Place Order

Category:
Tea & Milk

Search

Lemon Tea
Thai Tea
Milk Caramel
Hot Milk
Hot Tea

→

Place Order

Category:
Tea & Milk

H

Hot Milk
Hot Tea

ภาพที่ 4.1.2 : การกรองข้อมูลในการเลือกอาหาร

Place Order

Category:
Tea & Milk

H

Hot Milk
Hot Tea

Item Name: Hot Milk

Price: 32

Quantity: 2

Total: 64

Discount(%) : 0

Add to Cart

Item Name	Unit Price	Quantity	Discount (%)	Price

Grand Total

0.00B

Remove

Print

ใส่ได้ในช่วง 0 - 100
หากเกินช่วงนี้ จะมี
MessageBox แจ้ง

ภาพที่ 4.1.3 : การเลือกอาหาร และมีข้อมูลปรากฏ


```
//เลือกรายการอาหาร
1 reference
private void listBox1_SelectedIndexChanged(object sender, EventArgs e)
{
    txtQuantityUpDown.ResetText();
    txtTotal.Clear();

    String text = listBox1.GetItemText(listBox1.SelectedItem);
    txtItemName.Text = text;
    query = "select price from New_table_for_project where names = '" + text + "'";
    DataSet ds = fn.getData(query);
    txtQuantityUpDown.Text = "0";
    txtTotal.Text = "0.00";
    txtDisCount.Text = "0";

    try
    {
        txtPrice.Text = ds.Tables[0].Rows[0][0].ToString();
    }
    catch {}
}
```

ds ที่ได้จะมีข้อมูลแค่ตารางแถวเดียว เป็นของรายการอาหารนั้นเลย

เมื่อเลือกรายการอาหารที่ ListBox แล้วในส่วนของ Item Name กับ Price จะปรากฏข้อมูลขึ้นมา

```
//เลือกจำนวน
1 reference
private void txtQuantityUpDown_ValueChanged(object sender, EventArgs e)
{
    Int64 quan = Int64.Parse(txtQuantityUpDown.Value.ToString());
    Int64 price = Int64.Parse(txtPrice.Text);
    txtTotal.Text = (quan*price).ToString("#.##");
}
```

เมื่อเปลี่ยนตัวเลขใน Quantity แล้ว Total จะมีการเปลี่ยนแปลง

```
//กดเพิ่มข้อมูลลงตาราง
protected int n, total = 0;
1 reference
private void btnAddtoCart_Click(object sender, EventArgs e)
{
    /*DataGridView
    Name      Unit      Quantity      Discount      Total/Price
    A          50         5             10             250 - 10% = 225
    */
    if (Int64.Parse(txtDisCount.Text) <= 100 && Int64.Parse(txtDisCount.Text) >= 0)
    {
        if (txtTotal.Text != "0" && txtTotal.Text != "" && txtQuantityUpDown.Text != "0")
        {
            n = dataGridView1.Rows.Add();
            dataGridView1.Rows[n].Cells[0].Value = txtItemName.Text;
            dataGridView1.Rows[n].Cells[1].Value = txtPrice.Text;
            dataGridView1.Rows[n].Cells[2].Value = txtQuantityUpDown.Value;

            if (txtDisCount.Text == "0")
            {
                dataGridView1.Rows[n].Cells[3].Value = "-";
            }
            else
            {
                dataGridView1.Rows[n].Cells[3].Value = "-" + txtDisCount.Text;
            }

            dataGridView1.Rows[n].Cells[4].Value = Math.Ceiling(Int64.Parse(txtTotal.Text) *
                (1 - 0.01 * Int64.Parse(txtDisCount.Text))).ToString();

            total = total + int.Parse(Math.Ceiling(Int64.Parse(txtTotal.Text) *
                (1 - 0.01 * Int64.Parse(txtDisCount.Text))).ToString());
            txtTotalAmount.Text = total.ToString("#.##") + ".00 B";
        }
        else
        {
            MessageBox.Show("Minimum Quantity need to be 1", "Information",
                MessageBoxButtons.OK, MessageBoxIcon.Information);
        }
    }
    else
    {
        MessageBox.Show("Plz input discount in range (0,100)", "Warning",
            MessageBoxButtons.OK, MessageBoxIcon.Warning);
        txtDisCount.Clear();
    }
}
```

- **If - else (แดง)** : ตรวจสอบว่า txtdiscount อยู่ในช่วง 0 - 100 หรือไม่
- **If - else (น้ำเงิน)** : ตรวจสอบว่ามีข้อมูลครบถ้วนหรือไม่ (เพราะตอนสร้างตารางตอนเริ่มต้นตั้งค่าว่า not null) และทำการเพิ่มข้อมูลใหม่ลงตาราง มาแสดงในตารางข้างล่างสีชมพู
- **If - else (เขียว)** : ใช้การใส่ตัวเลขของ discount ถ้า = 0 ไม่ต้องใส่ ให้ใส่ "-" พอ

Place Order

Category: Tea & Milk

H

Hot Milk
Hot Tea

Item Name: Hot Milk Price: 32

Quantity: 2 Total: 64

Discount(%): 0 Add to Cart

Item Name	Unit Price	Quantity	Discount (%)	Price
Hot Milk	32	2	-	64

Grand Total

Remove 64.00 B Print

ภาพที่ 4.1.4 : การกด Add to Cart

```
//เตรียมจะลบข้อมูล
int amount;
1 reference
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (dataGridView1.Rows[dataGridView1.CurrentCell.RowIndex] != null)
    {
        try
        {
            amount = int.Parse(dataGridView1.Rows[e.RowIndex].Cells[4].Value.ToString());
        }
        catch { }
    }
}

//ลบข้อมูล
1 reference
private void btnRemove_Click(object sender, EventArgs e)
{
    try
    {
        int rowIndex = dataGridView1.CurrentCell.RowIndex;
        dataGridView1.Rows.RemoveAt(rowIndex);
    }
    catch { }
    total -= amount;
    txtTotalAmount.Text = total.ToString("#.##") + ".00 B";
}
```

2. โปรแกรมพิมพ์ใบเสร็จ (DGVPrinter – DataGridView Printer) :

ในส่วนนี้จะมีการแนบ Source Code สำหรับ DGVPrinter ไปด้วย เนื่องจาก Source Code ดังกล่าวมี Code สำหรับการพิมพ์ตารางที่ทำให้การทำโปรเจกต์สะดวกมากขึ้น

```
private void btnPrint_Click(object sender, EventArgs e)
{
    DGVPrinter printer = new DGVPrinter();
    //Text//
    printer.Title = "Customer Bill";
    printer.SubTitle = "I Want to Sleep Plzz 'Cafe \n\r"
        + "Chula Soi 999, Bangjungah BKK 10500" + "\n\r"
        + string.Format("Date: {0}", DateTime.Now) + "\n\r";
    printer.SubTitleFormatFlags = StringFormatFlags.LineLimit | StringFormatFlags.NoClip;
    printer.PageNumbers = true;
    printer.PageNumberInHeader = false;
    printer.PortportionalColumns = true;
    printer.HeaderCellAlignment = StringAlignment.Near;
    printer.Footer = "Total Payable Amount : " + txtTotalAmount.Text
        + "\n\r Thank you, Good bye ^_^";
    printer.FooterSpacing = 45;
    printer.PrintDataGridView(dataGridView1);

    total = 0;
    dataGridView1.Rows.Clear();
    txtTotalAmount.Text = "0.00B";
}
```

[\[Link : DGV Printer\]](#)

Customer Bill

I Want to Sleep Plzz 'Cafe
Chula Soi 999, Bangjungah BKK 10500
Date: 17/11/2566 18:37:12

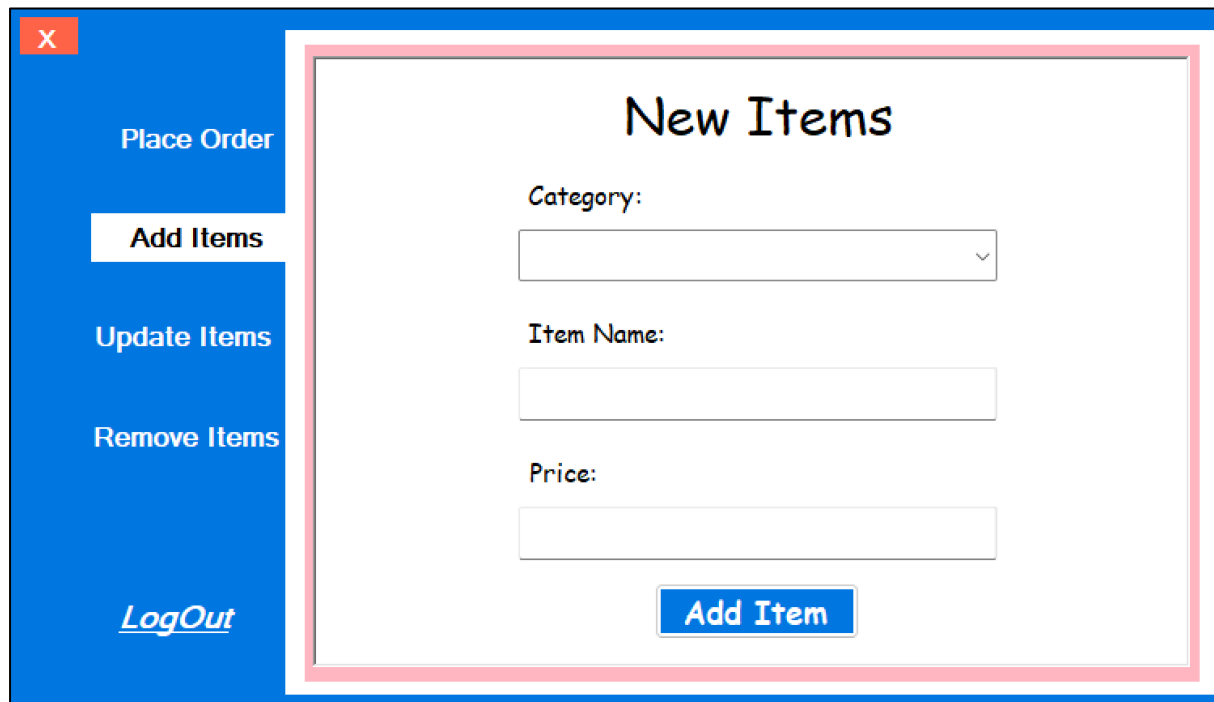
Item Name	Price (/unit)	Quantity	Discount	Total (B)
Melon Pie	50	1	-	50
Melon Pie	50	3	-	150
Blue Hawaii	30	1	-	30

Total Payable Amount : 230.00 B
Thank you, Good bye ^_^

Page 1

ภาพที่ 4.1.5 : ตัวอย่างการพิมพ์ใบเสร็จ

ส่วนที่ 2: Add Items (เพิ่มข้อมูลรายการอาหาร)



New Items

Category:

Item Name:

Price:

Add Item

ภาพที่ 4.2.1 : หน้าจอ Add Items

- วิธีการใช้งาน : ในคลิป หรือผ่านลิงค์ [ลิงค์ Clip : Add Items](#)

โปรแกรม (Code):

```

public partial class UC_Additems : UserControl
{
    Function fn = new Function();
    String query;
    1 reference
    public UC_Additems()
    {
        InitializeComponent();
    }

    //ไม่ Add Item
    1 reference
    private void button1_Click(object sender, EventArgs e)
    {
        query = "insert into New_table_for_project (names,category,price) " +
            "values ('" + txt_ItemName.Text + "','" + txtCategory.Text + "','" + txt_Price.Text + ")";
        fn.setData(query);
        clearAll();
    }
}

```

ใช้คำสั่ง Insert เพิ่มข้อมูลในตาราง

```

private void clearAll()
{
    txtCategory.SelectedIndex = -1; //Reset Category
    txt_ItemName.Clear();
    txt_Price.Clear();
}
1 reference
private void UC_Additems_Leave(object sender, EventArgs e)
{
    clearAll();
}

```

ส่วนที่ 3: Update Items (อัปเดต / เขียนทับข้อมูลลงไป)

Update Item

Item Name:

	iid	names	category	price
▶	2	Lemon Tea	Tea & Milk	45
	3	Bingsu	Cake	500
	4	Sushi	Meal	100
	5	Thai Tea	Tea & Milk	45

Category:

Price:

Item Name:

ภาพที่ 4.3.1 : หน้าจอ Update Items

- วิธีการใช้งาน : ในคลิป หรือผ่านลิงค์ [\[ลิงค์ Clip : Update Items\]](#)

โปรแกรม (Code):

```

public partial class UC_UpdateItems : UserControl
{
    Function fn = new Function();
    String query;
    1 reference
    public UC_UpdateItems()
    {
        InitializeComponent();
    }
    1 reference
    private void UC_UpdateItems_Load(object sender, EventArgs e)
    {
        query = "select * from New_table_for_project";
        loadData(query);
    }
    4 references
    private void loadData(string query)
    {
        DataSet ds = fn.getData(query);
        dataGridView1.DataSource = ds.Tables[0];
    }
}

```

ทุกครั้งที่มีการเปิด
หน้าจอก็จะทำงาน

เรียกตาราง
ออกมาแสดงผล


```

private void txtSearchItem_TextChanged(object sender, EventArgs e)
{
    query = "select * from New_table_for_project where names like '"
        + txtSearchItem.Text + "%'";
    loadData(query);
}

int id;
1 reference
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    try
    {
        id : แทนแถวตารางที่กด
        id = int.Parse(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());
        String category = dataGridView1.Rows[e.RowIndex].Cells[2].Value.ToString();
        String name = dataGridView1.Rows[e.RowIndex].Cells[1].Value.ToString();
        int price = int.Parse(dataGridView1.Rows[e.RowIndex].Cells[3].Value.ToString());

        txtCategory.ForeColor = Color.Black;
        txtCategory.Text = category;

        txtName.ForeColor = Color.Black;
        txtName.Text = name;

        txtPrice.ForeColor = Color.Black;
        txtPrice.Text = price.ToString();
    }
    catch { }
}

```

ทุกครั้งที่มีการคลิกที่เซลล์ จะขึ้นชื่อ หมวดหมู่
และราคาของรายการนั้น ๆ ขึ้นมา

ปุ่มอัปเดต หรือเขียนทับข้อมูลนั้น ๆ ใหม่

```

private void btnUpdate_Click(object sender, EventArgs e)
{
    query = "update New_table_for_project set names = '" + txtName.Text + "'" +
        ",category= '" + txtCategory.Text + "'" +
        ",price = " + txtPrice.Text + " " +
        " where iid = " + id + " ";

    fn.setData(query);
    loadData("select * from New_table_for_project");

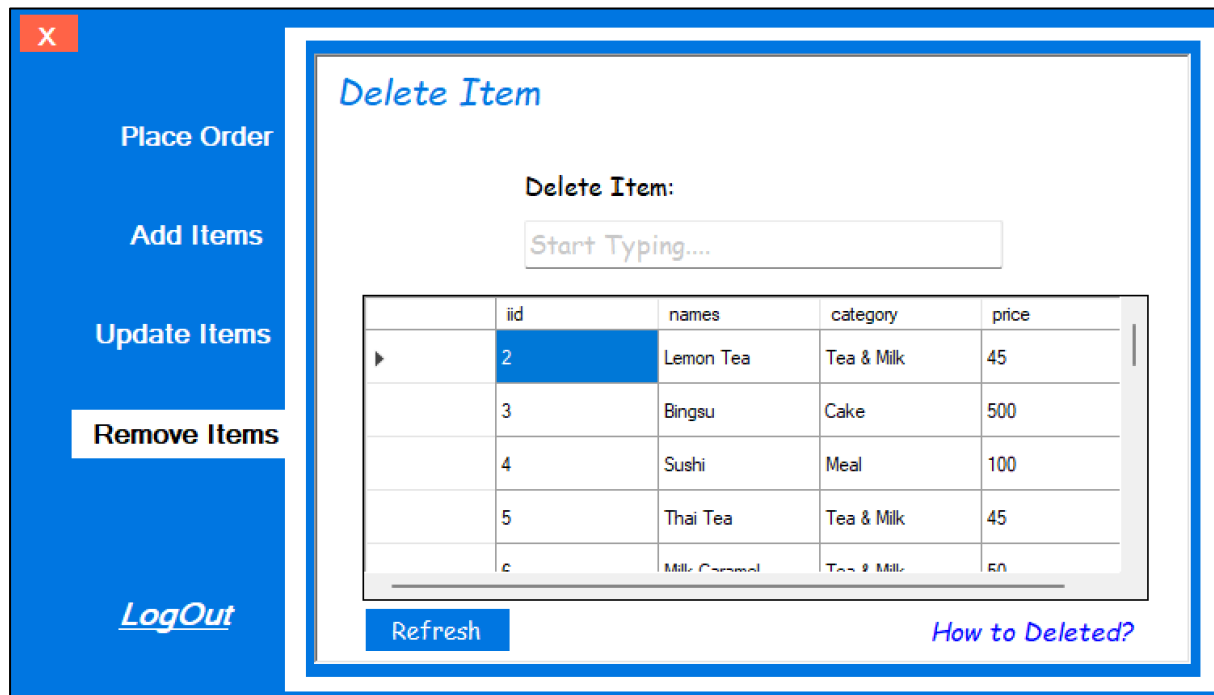
    txtCategory.Text = "    Category";
    txtCategory.ForeColor = Color.Silver;
    txtName.Text = "    Item Name";
    txtName.ForeColor = Color.Silver;
    txtPrice.Text = "    Price";
    txtPrice.ForeColor = Color.Silver;
}

1 reference
private void btnRefresh_Click(object sender, EventArgs e)
{
    loadData("select * from New_table_for_project");
}

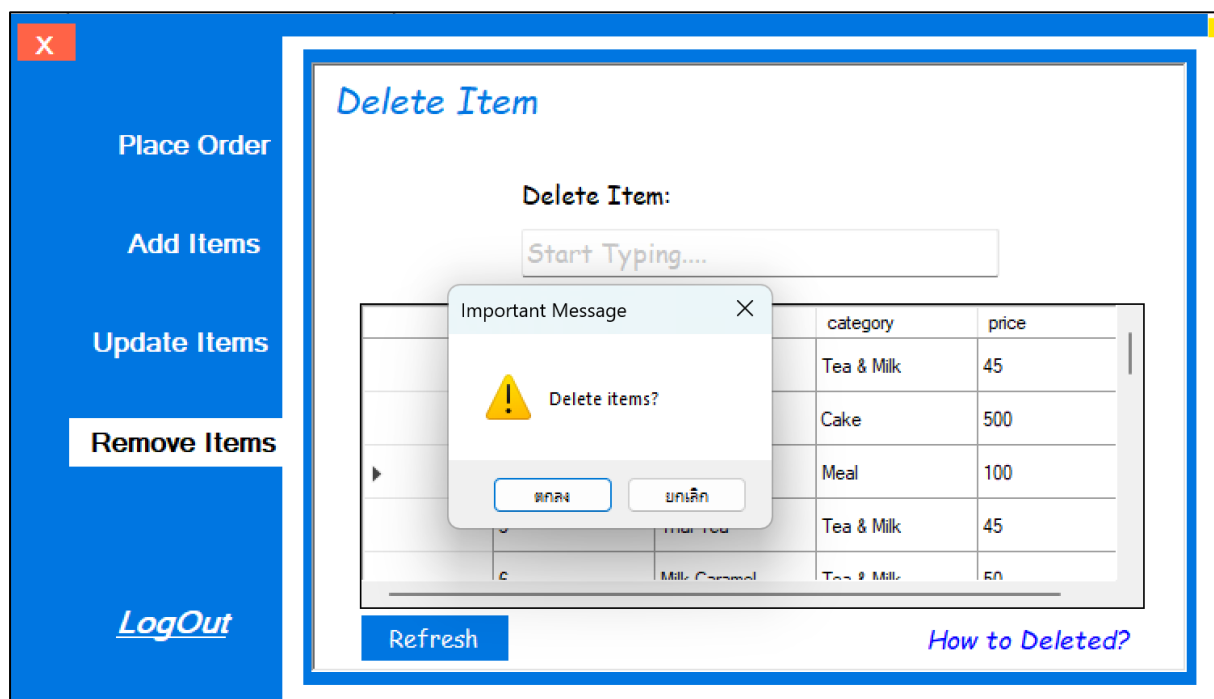
```

เมื่ออัปเดตตารางแล้ว ให้เปลี่ยน
ช่องข้อมูลที่แสดงเป็นสีเทา

ส่วนที่ 4: Remove Items (ลบข้อมูลออกจากฐานข้อมูล)



ภาพที่ 4.4.1 : หน้าจอ Remove Items



ภาพที่ 4.4.2 : หน้าต่างแจ้งเตือน การลบข้อมูล

- วิธีการใช้งาน : ในคลิป หรือผ่านลิงค์ [\[ลิงค์ Clip : Remove Items\]](#)

โปรแกรม (Code):

```

public partial class UC_RemoveItems : UserControl
{
    Function fn = new Function();
    String query;
    1 reference
    public UC_RemoveItems()
    {
        InitializeComponent();
    }

    1 reference
    private void UC_RemoveItems_Load(object sender, EventArgs e)
    {
        DelLabel.Text = "How to Deleted?";
        DelLabel.ForeColor = Color.Blue;

        → loadData("select * from New_table_for_project");
    }
    5 references
    private void loadData(string query)
    {
        DataSet ds = fn.getData(query);
        dataGridView1.DataSource = ds.Tables[0];
    }
}

```

เรียกตาราง
ออกมาแสดงผล

```

1 reference
private void txt_ItemName_TextChanged(object sender, EventArgs e)
{
    if (txt_ItemName.Text != "Start Typing...")
    {
        txt_ItemName.ForeColor = Color.Black;
        query = "select * from New_table_for_project where names like '" +
            txt_ItemName.Text + "%'";
        loadData(query);
    }
}

```

เมื่อกดที่เซลล์แล้ว จะมีหน้าต่างแจ้งเตือนก่อนจะลบข้อมูลนั้น และถ้ากดตกลงก็จะทำการลบข้อมูล

```

1 reference
private void dataGridView1_CellClick(object sender, DataGridViewCellEventArgs e)
{
    if (MessageBox.Show("Delete items?", "Important Message", MessageBoxButtons.OKCancel,
        MessageBoxIcon.Warning) == DialogResult.OK)
    {
        int id = int.Parse(dataGridView1.Rows[e.RowIndex].Cells[0].Value.ToString());

        query = "delete from New_table_for_project where iid = " + id + " ";
        fn.setData(query);
        loadData("select * from New_table_for_project");
    }
}

```

เมื่อกดที่ How to Deleted? จะมีวิธีบอก

```

private void DelLabel_Click(object sender, EventArgs e)
{
    DelLabel.ForeColor = Color.Red;
    DelLabel.Text = "*Click on Row to Delete the Item.";
    DelLabel.Location = new Point(786 - 300, 694);
}

```

```

1 reference
private void UC_RemoveItems_Enter(object sender, EventArgs e)
{
    loadData("select * from New_table_for_project");
}

```

```

1 reference
private void btnRefresh_Click(object sender, EventArgs e)
{
    loadData("select * from New_table_for_project");
}

```

