****

**โปรแกรมจัดการฐานข้อมูลอุปกรณ์คอมพิวเตอร์**

**จัดทำโดย**

นายสันติภาพ ภูรับ 1610900787

ภาควิชาวิศวกรรมคอมพิวเตอร์และหุ่นยนต์

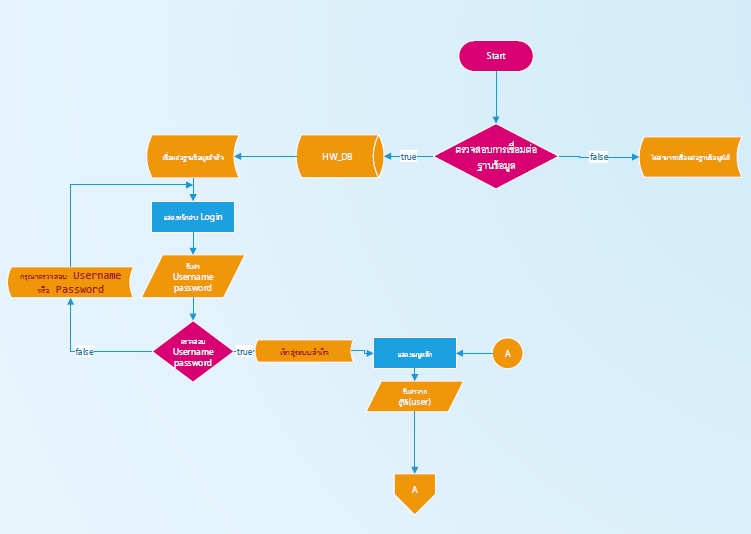
**เสนอ**

อาจารย์ ธิดารัตน์ ต่อสุข

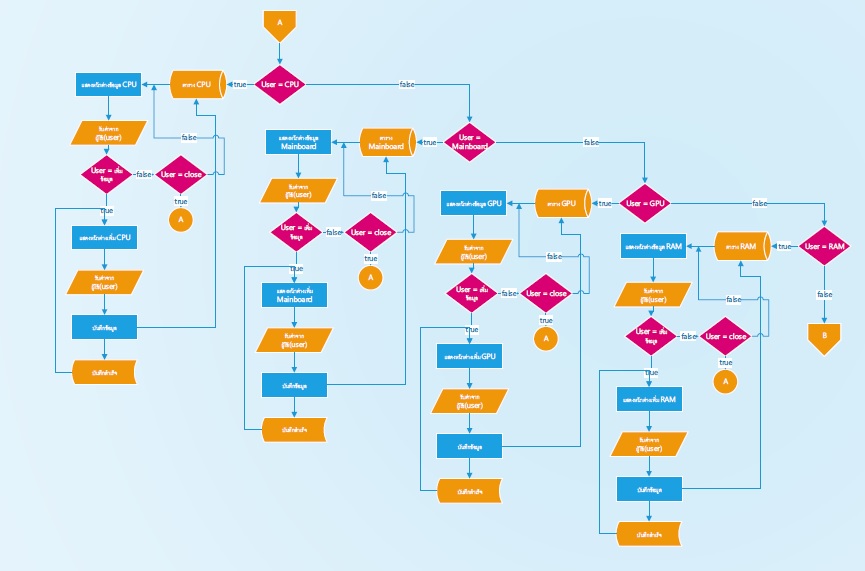
**โครงงานนี้เป็นส่วนหนึ่งของวิชา CE321\_322**

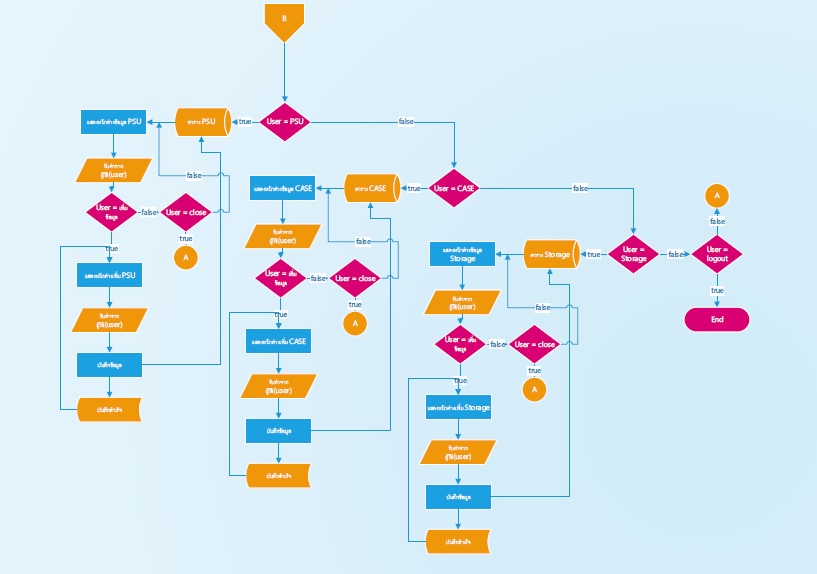
**ภาคการศึกษาที่ 2/2562**

1. **ขอบเขตของโครงงาน**
   1. เขียนโปรแกรมโดยใช้ภาษา VB.net ผ่านโปรแกรม Visual Studio 2019
   2. เชื่อมต่อฐานข้อมูล Microsoft Access 2016
   3. แสดงสถานะในการเชื่อมต่อกับฐานข้อมูลได้
   4. สามารถแสดงข้อมูลที่อยู่ในตารางภายใน Access ได้
   5. สามารถจำกัดผู้ใช้งานโดยผ่านระบบ Login ได้
   6. สามารถจำกัดสิทธิในการเพิ่มชื่อผู้ใช้งานใหม่ได้
   7. สามารถเพิ่ม,ลบ,ค้นหาข้อมูลในตารางฐานข้อมูลได้
2. Flowchart



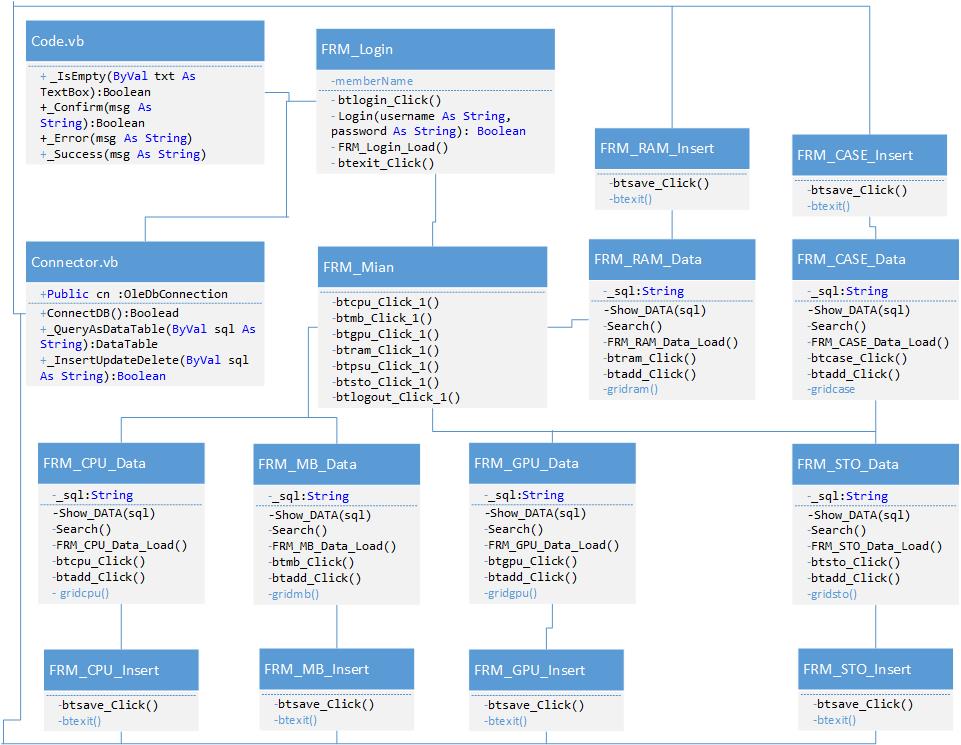
เมื่อเริ่มโปรแกรมจะเชื่อมต่อกับฐานข้อมูลถ้าเชื่อมต่อสำเร็จจะแสดงหน้าต่างเข้าสู่ระบบขึ้นมาพร้อมแสดงสถานะบอก



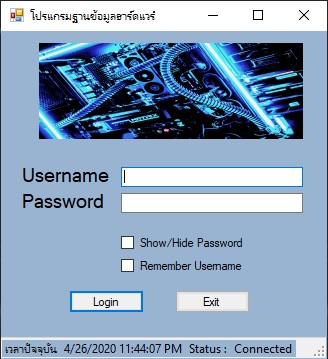


เมื่อเข้าสู่ระบบสำเร็จผู้ใช้จะทำการเลือกอุปกรณ์ที่ต้องการจะดู เช่น เมื่อผู้ใช้กดปุ่ม CPU โปรแกรมจะทำการดึงข้อมูลจากตาราง tbl\_cpu ที่อยู่ในฐานข้อมูลขึ้นมา และแสดงหน้าต่างแสดงข้อมูล cpu ขึ้นมา และเมื่อผู้ใช้กดเพิ่มข้อมูลและกรอกข้อมูลและกดบันทึกข้อมูลจะถูกเก็บลงตาราง

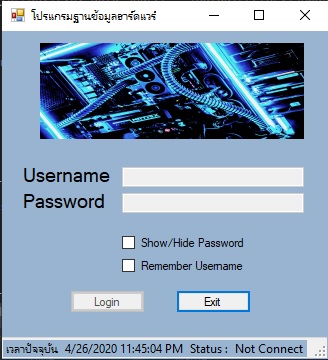
1. UML Diagram

class FRM\_Login จะทำการดึงฟังก์ชั่นจาก module Connector เพื่อทำการเชื่อมต่อกับฐานข้อมูล โดย Module Code จะมีหน้าที่เก็บฟังก์ชั่นที่จะมีการเรียกใช้งานบ่อย ๆ เช่น การแสดงกล่องข้อความแจ้งเตือน โดยคลาสเมนูหลักจะมีการเรียกใช้ฟังก์ชั่นในคลาสที่ผู้ใช้ต้องการจะเรียกดูข้อมูล และการเพิ่มข้อมูลคลาสที่เพิ่มข้อมูลจะมีการเชื่อมต่อกับโมดูล Connector เพื่อทำการใช้งานฟังก์ชั่นที่ใช้ติดต่อกับฐานข้อมูล

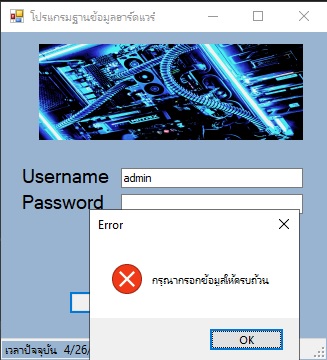
1. **การทำงานของโปรแกรม**
   1. **หน้าต่าง Login(กรณีเชื่อมต่อฐานข้อมูลสำเร็จ)**

****

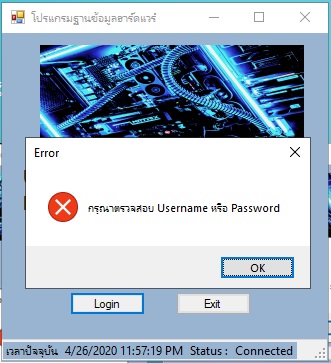
* 1. **หน้าต่าง Login(กรณีเชื่อมต่อฐานข้อมูลไม่สำเร็จ)**

****

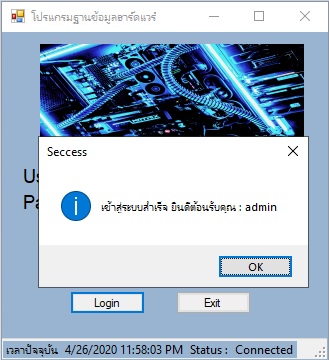
* 1. **เมื่อใส่ข้อมูลไม่ครบ**

****

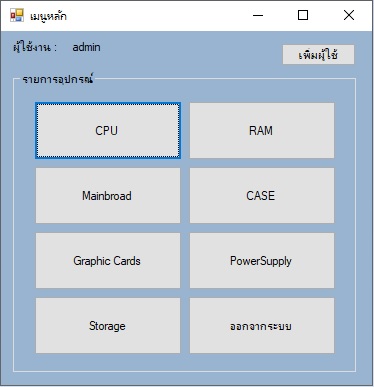
* 1. **เมื่อใส่ข้อมูลไม่ถูกต้อง**

****

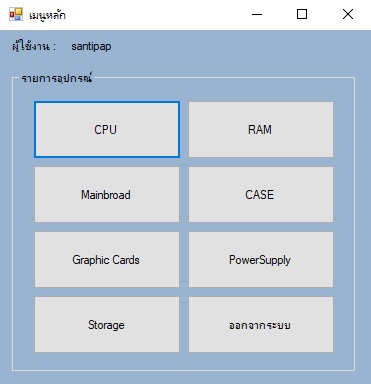
* 1. **เมื่อใส่ข้อมูลถูกต้อง**

****

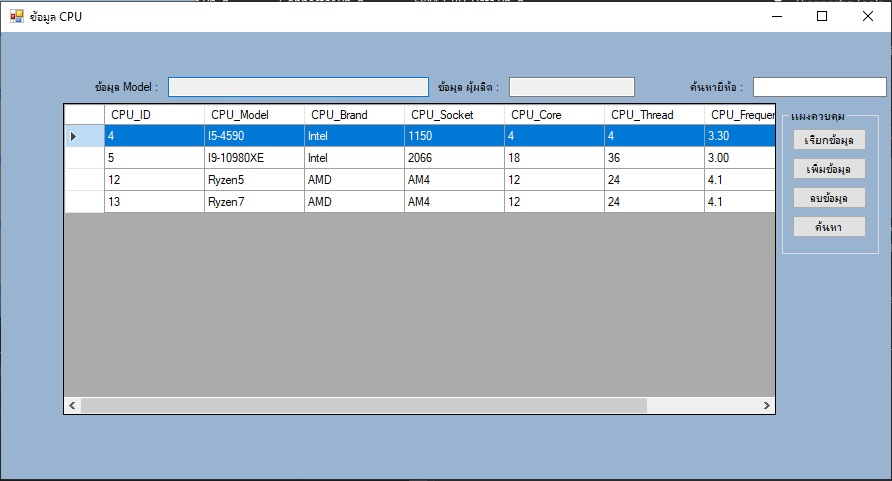
* 1. **หน้าต่างเมนูหลัก(กรณี Admin)**

****

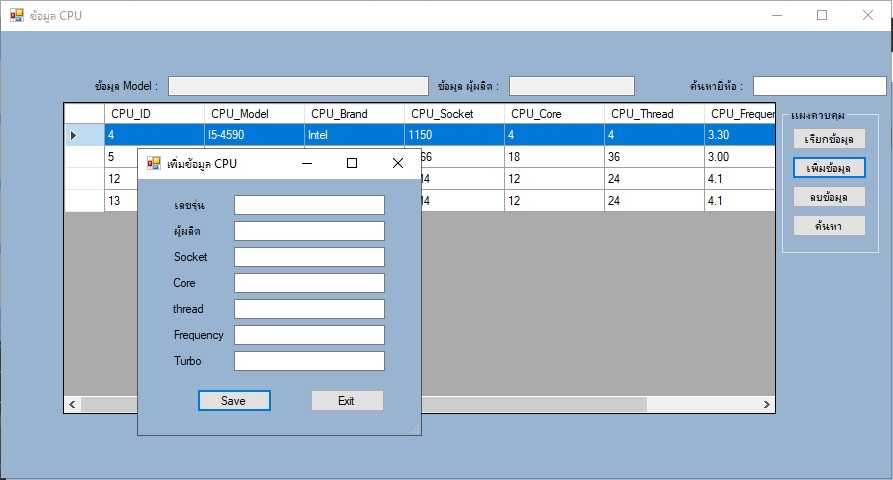
* 1. **หน้าต่างเมนูหลัก(กรณีไม่ใช่ Admin)**

****

* 1. **หน้าต่างแสดงข้อมูลอุปกรณ์ต่าง ๆ**

****

* 1. **หน้าต่างเพิ่มข้อมูลอุปกรณ์ต่าง ๆ**

****

1. **Source Code**
   1. **Connector**

Imports System.Data.OleDb

Module Connector

Public cn As New OleDbConnection("Provider=Microsoft.ACE.OLEDB.12.0; Data Source = |DataDirectory|\HW\_DB.accdb; Persist Security Info = false;") 'ประกาศตัวแปรสำหรับเก็บที่อยู่ของฐานข้อมูล

Public Function ConnectDB() As Boolean 'สร้างฟังก์ชั่นสำหรับเชื่อมต่อฐานข้อมูล

Try

If cn.State = ConnectionState.Closed Then 'ตรวจสอบว่าที่อยู่ของฐานข้อมูลว่ามีอยู่จริงหรือไม่

cn.Open() 'ทำการเปิดฐานข้อมูลที่อยู่ในตัวแปร cn

End If

Return True 'ส่งค่าคืนเป็น true

Catch ex As Exception

Return False 'ส่งค่าคืนเป็น false ในกรณีที่เปิดฐานข้อมูลไม่ได้

End Try

End Function

Public Function \_QueryAsDataTable(ByVal sql As String) As DataTable 'สร้างฟังก์ชั่นสำหรับใช้แสดงข้อมูลในตาราง

Dim da As New OleDbDataAdapter(sql, cn) 'ประกาศตัวแปรโดยมีพารามิเตอร์เป็นคำสั่ง sql ที่มาจากฟอร์มอื่น และ ที่อยู่ของฐานข้อมูล

Dim ds As New DataSet 'ประกาศตัวแปรสำหรับเก็บข้อมูลจากตารางฐานข้อมูล

da.Fill(ds, "Result") 'เก็บผลลัพท์ที่ได้จากคำสั่ง sql ลงใน ds

Return ds.Tables("Result") 'ส่งค่าผลลัพท์กลับไป

End Function

Public Function \_InsertAdd(ByVal sql As String) As Boolean 'เรียกใช้ฟังก์ชั่นสำหรับการเพิ่มข้อมูลลงตาราง

ConnectDB() 'ใช้งานฟังก์ชั่นการเชื่อมต่อ

Dim cmd As New OleDbCommand(sql, cn) 'ประกาศตัวแปรโดยมีพารามิเตอร์เป็นคำสั่ง sql ที่มาจากฟอร์มอื่น และ ที่อยู่ของฐานข้อมูล

Return cmd.ExecuteNonQuery() > 0 'ส่งค่าการบันทึกกลับไป

End Function

Public Function \_Delete(ByVal sql As String) As Boolean 'เรียกใช้ฟังก์ชั่นสำหรับการลบข้อมูลลงตาราง

ConnectDB() 'ใช้งานฟังก์ชั่นการเชื่อมต่อ

Dim cmd As New OleDbCommand(sql, cn) 'ประกาศตัวแปรโดยมีพารามิเตอร์เป็นคำสั่ง sql ที่มาจากฟอร์มอื่น และ ที่อยู่ของฐานข้อมูล

Return cmd.ExecuteNonQuery() > 0 'ส่งค่าการบันทึกกลับไป

End Function

End Module

* 1. **Code**

Imports System.Runtime.CompilerServices

Module code

<Extension()> Public Function \_IsEmpty(ByVal txt As TextBox) As Boolean 'สร้างฟังกชั่นสำหรับตรวจสอบค่าว่าในtextbox

Return String.IsNullOrWhiteSpace(txt.Text) 'ส่งคำสั่งเช็คค่่าค่าว่างกลับไป

End Function

<Extension()> Public Function \_isEmpty(ByVal str As String) As Boolean 'สร้างฟังกชั่นสำหรับตรวจสอบค่าว่าในtextbox

Return String.IsNullOrWhiteSpace(str) 'ส่งคำสั่งเช็คค่่าค่าว่างกลับไป

End Function

Public Function \_Confirm(msg As String) As Boolean 'สร้างฟังก์ชั่นสำหรับกล่องข้อความยืนยัน

Return MessageBox.Show(msg, "ยืนยัน", MessageBoxButtons.YesNo, MessageBoxIcon.Question) = DialogResult.Yes 'ส่งค่าโครงสร้างของกล่องข้อความกลับไป

End Function

Public Sub \_Error(msg As String) 'สร้างฟังก์ชั่นสำหรับกล่องข้อความผิดพลาด

MessageBox.Show(msg, "Error", MessageBoxButtons.OK, MessageBoxIcon.Error) 'ส่งค่าโครงสร้างของกล่องข้อความกลับไป

End Sub

Public Sub \_Success(msg As String) 'สร้างฟังก์ชั่นสำหรับกล่องข้อความสำเร็จ

MessageBox.Show(msg, "Seccess", MessageBoxButtons.OK, MessageBoxIcon.Information) 'ส่งค่าโครงสร้างของกล่องข้อความกลับไป

End Sub

End Module

* 1. **FRM\_Login**

Public Class FRM\_Login

Private Sub chkshowhide\_CheckedChanged(sender As Object, e As EventArgs) Handles chkshowhide.CheckedChanged

txtpassword.PasswordChar = If(chkshowhide.Checked, "", "\*") 'เมื่อchekboxถูกกดให้แสดงรหัสผ่านที่กรอกเป็นข้อความ

End Sub

Private Sub btlogin\_Click(sender As Object, e As EventArgs) Handles btlogin.Click

If txtusername.\_IsEmpty Or txtpassword.\_IsEmpty Then 'ตรวจสอบว่าช่องกรอก Username และ Password ว่างหรือไม่

\_Error("กรุณากรอกข้อมูลให้ครบถ้วน") 'ในกรณีที่ว่างให้เรียกใช้ฟังก์ชั่น \_Error เพื่อแสดงกล่องข้อความแจ้งเตือน

Else

Dim isLoginCorrect As Boolean = Login(txtusername.Text, txtpassword.Text) 'ประกาศตัวแปรและกำหนดค่าจากฟังก์ชั่น Login

If isLoginCorrect Then 'ตรวจสอบตัวแปร

\_Success("เข้าสู่ระบบสำเร็จ ยินดีต้อนรับคุณ : " + txtusername.Text) 'เรียกใช้ฟังก์ชั่นในกรณีที่ username กับ password ถูกต้อง

FRM\_MAIN.Show() 'เปิด Form MAIN

Me.Hide() 'ซ่อนฟอร์มนี้

Else

\_Error("กรุณาตรวจสอบ Username หรือ Password") 'เรียกใช้ฟังก์ชั่นในกรณีที่ username กับ password ผิด

End If

If chkremember.Checked Then 'ตรวจสอบchekboxว่าถูกเลือกไหม

My.Settings.username = txtusername.Text 'เก็บค่าที่อยู่ใน textbox Username ไว้ในการตั้งค่าของโปรแกรม

My.Settings.Save() 'บันทึกการตั้งค่า

End If

End If

End Sub

Private Function Login(username As String, password As String) As Boolean

Dim sql As String = $"SELECT \* FROM tbl\_admin WHERE Username = '{username}' AND password = '{password}'" 'ประกาศตัวแปรและเก็บค่าคำสั่ง sql

Dim da As New OleDb.OleDbDataAdapter(sql, cn) 'ประกาศตัวแปรโดยมีพารามิเตอร์เป็นคำสั่ง sql และ ที่อยู่ของฐานข้อมูล

Dim ds As New DataSet 'ประกาศตัวแปรสำหรับเก็บข้อมูลจากตารางฐานข้อมูล

da.Fill(ds, "Result") 'เก็บผลลัพท์ที่ได้จากคำสั่ง sql ลงใน ds

If ds.Tables("Result").Rows.Count > 0 Then 'ตรวจสอบว่า ds ได้มีการรับข้อมูลไหม

Return True 'ส่งค่าคืนเป็น true

Else

Return False 'ส่งค่าคืนเป็น False

End If

End Function

Private Sub FRM\_Login\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

If Not My.Settings.username.\_isEmpty Then 'ตรวจสอบว่าค่า username ในการตั้งค่าว่าไหม

txtusername.Text = My.Settings.username 'ถ้ามีค่าอยู่ในการตั้งค่า ให้ textbox Username เท่ากับค่าที่อยู่ในระบบ

chkremember.Checked = True 'ทำการเลือก Cheakbox จำผู้ใช้ไว้

End If

Dim isConnected As Boolean = ConnectDB() 'สร้างตัวแปรสำหรับรับค่าจากฟังก์ชั่นการเชื่อมต่อ

If isConnected Then

status.Text = "Connected" 'ให้ขึ้นข้อความนี้เมื่อมีการรับค่า true มาจากฟังก์ชั่น

Else

status.Text = "Not Connect" 'ให้ขึ้นข้อความนี้เมื่อมีการรับค่า false มาจากฟังก์ชั่น

txtusername.Enabled = False 'ปิดการทำงานของวัตถุ

txtpassword.Enabled = False 'ปิดการทำงานของวัตถุ

btlogin.Enabled = False 'ปิดการทำงานของวัตถุ

End If

End Sub

Private Sub chkremember\_CheckedChanged(sender As Object, e As EventArgs) Handles chkremember.CheckedChanged

If chkremember.Checked = False Then 'ตรวจสอบchekboxว่าถูกเลือกไหม

My.Settings.username = "" 'เก็บค่าว่าง ไว้ในการตั้งค่าของโปรแกรม

My.Settings.Save() 'บันทึกการตั้งค่า

End If

End Sub

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

If \_Confirm("ต้องการออกจากโปรแกรมหรือไม่") Then 'เรียกใช้ฟังก์ชั่นเพื่อแสดงกล่องข้อความยืนยัน

Close() 'ปิดฟอร์ม

End 'จบการทำงาน

End If

End Sub

Private Sub tmclock\_Tick(sender As Object, e As EventArgs) Handles tmclock.Tick

lbclock.Text = Date.Now 'เก็บค่าเวลาปัจจุบันเพื่อแสดงในหน้าฟอร์ม

End Sub

End Class

* 1. **FRM\_Main**

Public Class FRM\_MAIN

Private Sub btcpu\_Click\_1(sender As Object, e As EventArgs) Handles btcpu.Click

FRM\_CPU\_Data.Show\_DATA("SELECT \* FROM tbl\_cpu") 'เปิดฟอร์ม CPU พร้อมเรียกใช้ฟังก์ชั่นโดยกำหนดพารามิเตอร์เป็นคำสั่ง sql

End Sub

Private Sub btmb\_Click\_1(sender As Object, e As EventArgs) Handles btmb.Click

FRM\_MB\_Data.Show\_DATA("SELECT \* FROM tbl\_mainboard") 'เปิดฟอร์ม Mainboard พร้อมเรียกใช้ฟังก์ชั่นโดยกำหนดพารามิเตอร์เป็นคำสั่ง sql

End Sub

Private Sub btgpu\_Click\_1(sender As Object, e As EventArgs) Handles btgpu.Click

FRM\_GPU\_Data.Show\_DATA("SELECT \* FROM tbl\_gpu") 'เปิดฟอร์ม Mainboard พร้อมเรียกใช้ฟังก์ชั่นโดยกำหนดพารามิเตอร์เป็นคำสั่ง sql

End Sub

Private Sub btram\_Click(sender As Object, e As EventArgs) Handles btram.Click

FRM\_RAM\_Data.Show\_DATA("SELECT \* FROM tbl\_ram") 'เปิดฟอร์ม RAM พร้อมเรียกใช้ฟังก์ชั่นโดยกำหนดพารามิเตอร์เป็นคำสั่ง sql

End Sub

Private Sub btcase\_Click(sender As Object, e As EventArgs) Handles btcase.Click

FRM\_CASE\_Data.Show\_DATA("SELECT \* FROM tbl\_case") 'เปิดฟอร์ม CASE พร้อมเรียกใช้ฟังก์ชั่นโดยกำหนดพารามิเตอร์เป็นคำสั่ง sql

End Sub

Private Sub btpsu\_Click(sender As Object, e As EventArgs) Handles btpsu.Click

FRM\_PSU\_Data.Show\_DATA("SELECT \* FROM tbl\_psu") 'เปิดฟอร์ม PSU พร้อมเรียกใช้ฟังก์ชั่นโดยกำหนดพารามิเตอร์เป็นคำสั่ง sql

End Sub

Private Sub btstorage\_Click(sender As Object, e As EventArgs) Handles btstorage.Click

FRM\_STO\_Data.Show\_DATA("SELECT \* FROM tbl\_storage") 'เปิดฟอร์ม Storage พร้อมเรียกใช้ฟังก์ชั่นโดยกำหนดพารามิเตอร์เป็นคำสั่ง sql

End Sub

Private Sub FRM\_MAIN\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Me.Label2.Text = FRM\_Login.txtusername.Text 'ให้ข้อความใน label เหมือนกับข้อความ username ในฟอร์ม Login

If Label2.Text = "admin" Then 'ตรวจสอบข้อความของ Label ว่าเป็น admin หรือไม่

btadduser.Visible = True 'กรณีที่เป็น admin ให้แสดงปุ่มเพิ่มผู้ใช้ขึ้นมา

Else

btadduser.Visible = False 'กรณีที่ไม่ใช่ admin ให้ซ่อนปุ่มเพิ่มผู้ใช้

End If

End Sub

Private Sub btlogout\_Click(sender As Object, e As EventArgs) Handles btlogout.Click

If \_Confirm("ต้องการออกจากระบบหรือไม่") Then 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความยืนยันพร้อมกำหนดข้อความในพารามิเตอร์

Close() 'ปิดฟอร์ม

FRM\_Login.Show() 'แสดงหน้าต่าง Login ขึ้นมา

End If

End Sub

Private Sub btadduser\_Click(sender As Object, e As EventArgs) Handles btadduser.Click

FRM\_ADDUSER.ShowDialog() 'เปิดหน้าต่างเพิ่มผู้ใช้

End Sub

End Class

* 1. **FRM\_CPU\_Data**

Imports System.Data.OleDb

Imports System.Data

Public Class FRM\_CPU\_Data

Dim da As OleDbDataAdapter 'สร้างตัวแปรสำหรับควบคุมคำสั่ง sql

Dim dt As DataTable 'สร้างตัวแปรสำหรับเก็บข้อมูลในตาราง

Dim ds As DataSet 'สร้างตัวแปรสำหรับเก็บเช็ตในตาราง

Private \_sql As String 'ประกาศตัวแปรสำหรับเก็บค่า sql จากภายนอก

Public Sub Show\_DATA(ByVal sql As String) 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

\_sql = sql 'กำหนดให้ตัวแปรที่สร้างขึ้นมามีค่าเท่ากับ sql ที่ได้รับจากภายนอก

Show() 'แสดงข้อมูล

End Sub

Private Sub Search() 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

gridcpu.DataSource = \_QueryAsDataTable(\_sql) 'ให้วัตถุแสดงข้อมูลในตารางฐานข้อมูลโดยการเรียกใช้ฟังก์ชั่นพร้อมใส่พารามิเตอร์ที่ได้รับค่ามาแล้ว

End Sub

Private Sub FRM\_CPU\_Data\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Search() 'เรียกใช้ฟังก์ชั่นเมื่อหน้าต่างถูกเปิดขึ้นมา

End Sub

Private Sub btdelete\_Click\_1(sender As Object, e As EventArgs) Handles btdelete.Click

If \_Confirm("ต้องการลบหรือไม่") Then 'แสดงข้อความยืนยันพร้อมกำหนดคำในพารามิเตอร์

Dim sql As String = "DELETE FROM [tbl\_cpu] WHERE CPU\_Model = '" & txtModel.Text & "'" 'ประกาศตัวแปรสำหรับก็บคำสั่งลบข้อมูลในตารางฐานข้อมูล

\_Delete(sql) 'เรียกใช้ฟังก์ชั่นลบข้อมูลโดยกำหนดตัวแปรที่เก็บคำสั่งในพารามิเตอร์

\_Success("ลบข้อมูลเรียบร้อย") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความพร้อมกำหนดคำในพารามิเตอร์

Search() 'เรียกใช้ฟังก์ชั่นแสดงข้อมูล

End If

End Sub

Private Sub gridcpu\_CellMouseUp(sender As Object, e As DataGridViewCellMouseEventArgs) Handles gridcpu.CellMouseUp

txtModel.Text = gridcpu.Rows.Item(e.RowIndex).Cells(1).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

txtBrand.Text = gridcpu.Rows.Item(e.RowIndex).Cells(2).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

End Sub

Private Sub btSearch\_Click(sender As Object, e As EventArgs) Handles btSearch.Click

dt = New DataTable 'กำหนดคุณสมบัติให้ตัวแปร

ds = New DataSet 'กำหนดคุณสมบัติให้ตัวแปร

ds.Tables.Add(dt) 'เก็บข้อมูลตารางที่ได้รับไว้ในตัวแปร

da = New OleDbDataAdapter($"SELECT \* FROM tbl\_cpu WHERE CPU\_Brand = '" & txtSearch.Text & "'", cn) 'ให้ตัวแปรเก็บคำสั่ง sql สำหรับค้นหาข้อมูล

da.Fill(dt) 'นำตัวแปรที่เก็บคำสั่งใปใช้กับตัวแปรที่เก็บตารางฐานข้อมูล

gridcpu.DataSource = dt.DefaultView 'ให้วัตถุแสดงข้อมูลที่อยู่ในตารางที่ผ่านการกรองมาแล้ว

End Sub

Private Sub btcpu\_Click\_1(sender As Object, e As EventArgs) Handles btcpu.Click

Search() 'เรียกใช้ฟังก์ชั่นสำหรับเรียกข้อมูลมาแสดงเมื่อกดปุ่ม

End Sub

Private Sub btadd\_Click(sender As Object, e As EventArgs) Handles btadd.Click

FRM\_CPU\_Insert.ShowDialog() 'แสดงหน้าต่างสำหรับเพิ่มข้อมูลใหม่เมื่อกดปุ่ม

End Sub

End Class

* 1. **FRM\_CASE\_Data**

Imports System.Data.OleDb 'เพิ่มไลบารี่สำหรับฐานข้อมูล

Imports System.Data 'เพิ่มไลบารี่สำหรับจัดการฐานข้อมูล

Public Class FRM\_CASE\_Data

Dim da As OleDbDataAdapter 'สร้างตัวแปรสำหรับควบคุมคำสั่ง sql

Dim dt As DataTable 'สร้างตัวแปรสำหรับเก็บข้อมูลในตาราง

Dim ds As DataSet 'สร้างตัวแปรสำหรับเก็บเช็ตในตาราง

Private \_sql As String 'ประกาศตัวแปรสำหรับเก็บค่า sql จากภายนอก

Private Sub FRM\_CASE\_Data\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Search() 'เรียกใช้ฟังก์ชั่นเมื่อหน้าต่างถูกเปิดขึ้นมา

End Sub

Private Sub Search() 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

gridcase.DataSource = \_QueryAsDataTable(\_sql) 'ให้วัตถุแสดงข้อมูลในตารางฐานข้อมูลโดยการเรียกใช้ฟังก์ชั่นพร้อมใส่พารามิเตอร์ที่ได้รับค่ามาแล้ว

End Sub

Public Sub Show\_DATA(ByVal sql As String) 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

\_sql = sql 'กำหนดให้ตัวแปรที่สร้างขึ้นมามีค่าเท่ากับ sql ที่ได้รับจากภายนอก

Show() 'แสดงข้อมูล

End Sub

Private Sub btcpu\_Click(sender As Object, e As EventArgs) Handles btcpu.Click

Search() 'เรียกใช้ฟังก์ชั่นสำหรับเรียกข้อมูลมาแสดงเมื่อกดปุ่ม

End Sub

Private Sub btadd\_Click(sender As Object, e As EventArgs) Handles btadd.Click

FRM\_CASE\_Insert.ShowDialog() 'แสดงหน้าต่างสำหรับเพิ่มข้อมูลใหม่เมื่อกดปุ่ม

End Sub

Private Sub btdelete\_Click(sender As Object, e As EventArgs) Handles btdelete.Click

If \_Confirm("ต้องการลบหรือไม่") Then 'แสดงข้อความยืนยันพร้อมกำหนดคำในพารามิเตอร์

Dim sql As String = "DELETE FROM [tbl\_case] WHERE CASE\_Model = '" & txtModel.Text & "'" 'ประกาศตัวแปรสำหรับก็บคำสั่งลบข้อมูลในตารางฐานข้อมูล

\_Delete(sql) 'เรียกใช้ฟังก์ชั่นลบข้อมูลโดยกำหนดตัวแปรที่เก็บคำสั่งในพารามิเตอร์

\_Success("ลบข้อมูลเรียบร้อย") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความพร้อมกำหนดคำในพารามิเตอร์

Search() 'เรียกใช้ฟังก์ชั่นแสดงข้อมูล

End If

End Sub

Private Sub gridcase\_CellMouseUp(sender As Object, e As DataGridViewCellMouseEventArgs) Handles gridcase.CellMouseUp

txtModel.Text = gridcase.Rows.Item(e.RowIndex).Cells(1).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

txtBrand.Text = gridcase.Rows.Item(e.RowIndex).Cells(2).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

End Sub

Private Sub Button1\_Click(sender As Object, e As EventArgs) Handles btSearch.Click

dt = New DataTable 'กำหนดคุณสมบัติให้ตัวแปร

ds = New DataSet 'กำหนดคุณสมบัติให้ตัวแปร

ds.Tables.Add(dt) 'เก็บข้อมูลตารางที่ได้รับไว้ในตัวแปร

da = New OleDbDataAdapter($"SELECT \* FROM tbl\_case WHERE CASE\_Brand = '" & txtSearch.Text & "'", cn) 'ให้ตัวแปรเก็บคำสั่ง sql สำหรับค้นหาข้อมูล

da.Fill(dt) 'นำตัวแปรที่เก็บคำสั่งใปใช้กับตัวแปรที่เก็บตารางฐานข้อมูล

gridcase.DataSource = dt.DefaultView 'ให้วัตถุแสดงข้อมูลที่อยู่ในตารางที่ผ่านการกรองมาแล้ว

End Sub

End Class

* 1. **FRM\_GPU\_Data**

Imports System.Data.OleDb

Imports System.Data

Public Class FRM\_GPU\_Data

Dim da As OleDbDataAdapter 'สร้างตัวแปรสำหรับควบคุมคำสั่ง sql

Dim dt As DataTable 'สร้างตัวแปรสำหรับเก็บข้อมูลในตาราง

Dim ds As DataSet 'สร้างตัวแปรสำหรับเก็บเช็ตในตาราง

Private \_sql As String 'ประกาศตัวแปรสำหรับเก็บค่า sql จากภายนอก

Private Sub FRM\_GPU\_Data\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Search() 'เรียกใช้ฟังก์ชั่นเมื่อหน้าต่างถูกเปิดขึ้นมา

End Sub

Public Sub Show\_DATA(ByVal sql As String) 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

\_sql = sql 'กำหนดให้ตัวแปรที่สร้างขึ้นมามีค่าเท่ากับ sql ที่ได้รับจากภายนอก

Show() 'แสดงข้อมูล

End Sub

Private Sub Search() 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

gridgpu.DataSource = \_QueryAsDataTable(\_sql) 'ให้วัตถุแสดงข้อมูลในตารางฐานข้อมูลโดยการเรียกใช้ฟังก์ชั่นพร้อมใส่พารามิเตอร์ที่ได้รับค่ามาแล้ว

End Sub

Private Sub btcpu\_Click\_1(sender As Object, e As EventArgs) Handles btcpu.Click

Search() 'เรียกใช้ฟังก์ชั่นสำหรับเรียกข้อมูลมาแสดงเมื่อกดปุ่ม

End Sub

Private Sub btadd\_Click(sender As Object, e As EventArgs) Handles btadd.Click

FRM\_GPU\_Insert.ShowDialog() 'แสดงหน้าต่างสำหรับเพิ่มข้อมูลใหม่เมื่อกดปุ่ม

End Sub

Private Sub btdelete\_Click(sender As Object, e As EventArgs) Handles btdelete.Click

If \_Confirm("ต้องการลบหรือไม่") Then 'แสดงข้อความยืนยันพร้อมกำหนดคำในพารามิเตอร์

Dim sql As String = "DELETE FROM [tbl\_gpu] WHERE gpu\_Model = '" & txtModel.Text & "'" 'ประกาศตัวแปรสำหรับก็บคำสั่งลบข้อมูลในตารางฐานข้อมูล

\_Delete(sql) 'เรียกใช้ฟังก์ชั่นลบข้อมูลโดยกำหนดตัวแปรที่เก็บคำสั่งในพารามิเตอร์

\_Success("ลบข้อมูลเรียบร้อย") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความพร้อมกำหนดคำในพารามิเตอร์

Search() 'เรียกใช้ฟังก์ชั่นแสดงข้อมูล

End If

End Sub

Private Sub gridgpu\_CellMouseUp(sender As Object, e As DataGridViewCellMouseEventArgs) Handles gridgpu.CellMouseUp

txtModel.Text = gridgpu.Rows.Item(e.RowIndex).Cells(1).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

txtBrand.Text = gridgpu.Rows.Item(e.RowIndex).Cells(2).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

End Sub

Private Sub btSearch\_Click(sender As Object, e As EventArgs) Handles btSearch.Click

dt = New DataTable 'กำหนดคุณสมบัติให้ตัวแปร

ds = New DataSet 'กำหนดคุณสมบัติให้ตัวแปร

ds.Tables.Add(dt) 'เก็บข้อมูลตารางที่ได้รับไว้ในตัวแปร

da = New OleDbDataAdapter($"SELECT \* FROM tbl\_gpu WHERE GPU\_Brand = '" & txtSearch.Text & "'", cn) 'ให้ตัวแปรเก็บคำสั่ง sql สำหรับค้นหาข้อมูล

da.Fill(dt) 'นำตัวแปรที่เก็บคำสั่งใปใช้กับตัวแปรที่เก็บตารางฐานข้อมูล

gridgpu.DataSource = dt.DefaultView 'ให้วัตถุแสดงข้อมูลที่อยู่ในตารางที่ผ่านการกรองมาแล้ว

End Sub

End Class

* 1. **FRM\_MB\_Data**

Imports System.Data.OleDb

Imports System.Data

Public Class FRM\_MB\_Data

Dim da As OleDbDataAdapter 'สร้างตัวแปรสำหรับควบคุมคำสั่ง sql

Dim dt As DataTable 'สร้างตัวแปรสำหรับเก็บข้อมูลในตาราง

Dim ds As DataSet 'สร้างตัวแปรสำหรับเก็บเช็ตในตาราง

Private \_sql As String 'ประกาศตัวแปรสำหรับเก็บค่า sql จากภายนอก

Private Sub FRM\_MB\_Data\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Search() 'เรียกใช้ฟังก์ชั่นเมื่อหน้าต่างถูกเปิดขึ้นมา

End Sub

Public Sub Show\_DATA(ByVal sql As String) 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

\_sql = sql 'กำหนดให้ตัวแปรที่สร้างขึ้นมามีค่าเท่ากับ sql ที่ได้รับจากภายนอก

Show() 'แสดงข้อมูล

End Sub

Private Sub Search() 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

gridmb.DataSource = \_QueryAsDataTable(\_sql) 'ให้วัตถุแสดงข้อมูลในตารางฐานข้อมูลโดยการเรียกใช้ฟังก์ชั่นพร้อมใส่พารามิเตอร์ที่ได้รับค่ามาแล้ว

End Sub

Private Sub btcpu\_Click(sender As Object, e As EventArgs) Handles btcpu.Click

Search() 'เรียกใช้ฟังก์ชั่นสำหรับเรียกข้อมูลมาแสดงเมื่อกดปุ่ม

End Sub

Private Sub btadd\_Click(sender As Object, e As EventArgs) Handles btadd.Click

FRM\_MB\_Insert.ShowDialog() 'แสดงหน้าต่างสำหรับเพิ่มข้อมูลใหม่เมื่อกดปุ่ม

End Sub

Private Sub btdelete\_Click(sender As Object, e As EventArgs) Handles btdelete.Click

If \_Confirm("ต้องการลบหรือไม่") Then 'แสดงข้อความยืนยันพร้อมกำหนดคำในพารามิเตอร์

Dim sql As String = "DELETE FROM [tbl\_mainboard] WHERE MB\_Model = '" & txtModel.Text & "'" 'ประกาศตัวแปรสำหรับก็บคำสั่งลบข้อมูลในตารางฐานข้อมูล

\_Delete(sql) 'เรียกใช้ฟังก์ชั่นลบข้อมูลโดยกำหนดตัวแปรที่เก็บคำสั่งในพารามิเตอร์

\_Success("ลบข้อมูลเรียบร้อย") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความพร้อมกำหนดคำในพารามิเตอร์

Search() 'เรียกใช้ฟังก์ชั่นแสดงข้อมูล

End If

End Sub

Private Sub gridmb\_CellMouseUp(sender As Object, e As DataGridViewCellMouseEventArgs) Handles gridmb.CellMouseUp

txtModel.Text = gridmb.Rows.Item(e.RowIndex).Cells(1).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

txtBrand.Text = gridmb.Rows.Item(e.RowIndex).Cells(2).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

End Sub

Private Sub btSearch\_Click(sender As Object, e As EventArgs) Handles btSearch.Click

dt = New DataTable 'กำหนดคุณสมบัติให้ตัวแปร

ds = New DataSet 'กำหนดคุณสมบัติให้ตัวแปร

ds.Tables.Add(dt) 'เก็บข้อมูลตารางที่ได้รับไว้ในตัวแปร

da = New OleDbDataAdapter($"SELECT \* FROM tbl\_mainboard WHERE MB\_Brand = '" & txtSearch.Text & "'", cn) 'ให้ตัวแปรเก็บคำสั่ง sql สำหรับค้นหาข้อมูล

da.Fill(dt) 'นำตัวแปรที่เก็บคำสั่งใปใช้กับตัวแปรที่เก็บตารางฐานข้อมูล

gridmb.DataSource = dt.DefaultView 'ให้วัตถุแสดงข้อมูลที่อยู่ในตารางที่ผ่านการกรองมาแล้ว

End Sub

End Class

* 1. **FRM\_PSU\_Data**

Imports System.Data.OleDb 'เพิ่มไลบารี่สำหรับฐานข้อมูล

Imports System.Data 'เพิ่มไลบารี่สำหรับจัดการฐานข้อมูล

Public Class FRM\_PSU\_Data

Dim da As OleDbDataAdapter 'สร้างตัวแปรสำหรับควบคุมคำสั่ง sql

Dim dt As DataTable 'สร้างตัวแปรสำหรับเก็บข้อมูลในตาราง

Dim ds As DataSet 'สร้างตัวแปรสำหรับเก็บเช็ตในตาราง

Private \_sql As String 'ประกาศตัวแปรสำหรับเก็บค่า sql จากภายนอก

Private Sub FRM\_PSU\_Data\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Search() 'เรียกใช้ฟังก์ชั่นเมื่อหน้าต่างถูกเปิดขึ้นมา

End Sub

Private Sub Search() 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

gridpsu.DataSource = \_QueryAsDataTable(\_sql) 'ให้วัตถุแสดงข้อมูลในตารางฐานข้อมูลโดยการเรียกใช้ฟังก์ชั่นพร้อมใส่พารามิเตอร์ที่ได้รับค่ามาแล้ว

End Sub

Public Sub Show\_DATA(ByVal sql As String) 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

\_sql = sql 'กำหนดให้ตัวแปรที่สร้างขึ้นมามีค่าเท่ากับ sql ที่ได้รับจากภายนอก

Show() 'แสดงข้อมูล

End Sub

Private Sub btcpu\_Click(sender As Object, e As EventArgs) Handles btcpu.Click

Search() 'เรียกใช้ฟังก์ชั่นสำหรับเรียกข้อมูลมาแสดงเมื่อกดปุ่ม

End Sub

Private Sub btadd\_Click(sender As Object, e As EventArgs) Handles btadd.Click

FRM\_PSU\_Insert.ShowDialog() 'แสดงหน้าต่างสำหรับเพิ่มข้อมูลใหม่เมื่อกดปุ่ม

End Sub

Private Sub btdelete\_Click(sender As Object, e As EventArgs) Handles btdelete.Click

If \_Confirm("ต้องการลบหรือไม่") Then 'แสดงข้อความยืนยันพร้อมกำหนดคำในพารามิเตอร์

Dim sql As String = "DELETE FROM [tbl\_psu] WHERE PSU\_Model = '" & txtModel.Text & "'" 'ประกาศตัวแปรสำหรับก็บคำสั่งลบข้อมูลในตารางฐานข้อมูล

\_Delete(sql) 'เรียกใช้ฟังก์ชั่นลบข้อมูลโดยกำหนดตัวแปรที่เก็บคำสั่งในพารามิเตอร์

\_Success("ลบข้อมูลเรียบร้อย") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความพร้อมกำหนดคำในพารามิเตอร์

Search() 'เรียกใช้ฟังก์ชั่นแสดงข้อมูล

End If

End Sub

Private Sub btSearch\_Click(sender As Object, e As EventArgs) Handles btSearch.Click

dt = New DataTable 'กำหนดคุณสมบัติให้ตัวแปร

ds = New DataSet 'กำหนดคุณสมบัติให้ตัวแปร

ds.Tables.Add(dt) 'เก็บข้อมูลตารางที่ได้รับไว้ในตัวแปร

da = New OleDbDataAdapter($"SELECT \* FROM tbl\_psu WHERE PSU\_Brand = '" & txtSearch.Text & "'", cn) 'ให้ตัวแปรเก็บคำสั่ง sql สำหรับค้นหาข้อมูล

da.Fill(dt) 'นำตัวแปรที่เก็บคำสั่งใปใช้กับตัวแปรที่เก็บตารางฐานข้อมูล

gridpsu.DataSource = dt.DefaultView 'ให้วัตถุแสดงข้อมูลที่อยู่ในตารางที่ผ่านการกรองมาแล้ว

End Sub

Private Sub gridpsu\_CellMouseUp(sender As Object, e As DataGridViewCellMouseEventArgs) Handles gridpsu.CellMouseUp

txtModel.Text = gridpsu.Rows.Item(e.RowIndex).Cells(1).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

txtBrand.Text = gridpsu.Rows.Item(e.RowIndex).Cells(3).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

End Sub

End Class

* 1. **FRM\_RAM\_Data**

Imports System.Data.OleDb 'เพิ่มไลบารี่สำหรับฐานข้อมูล

Imports System.Data 'เพิ่มไลบารี่สำหรับจัดการฐานข้อมูล

Public Class FRM\_RAM\_Data

Dim da As OleDbDataAdapter 'สร้างตัวแปรสำหรับควบคุมคำสั่ง sql

Dim dt As DataTable 'สร้างตัวแปรสำหรับเก็บข้อมูลในตาราง

Dim ds As DataSet 'สร้างตัวแปรสำหรับเซ็ตข้อมูลในตาราง

Private \_sql As String 'ประกาศตัวแปรสำหรับเก็บค่า sql จากภายนอก

Public Sub Show\_DATA(ByVal sql As String) 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

\_sql = sql 'กำหนดให้ตัวแปรที่สร้างขึ้นมามีค่าเท่ากับ sql ที่ได้รับจากภายนอก

Show() 'แสดงข้อมูล

End Sub

Private Sub Search() 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

gridram.DataSource = \_QueryAsDataTable(\_sql) 'ให้วัตถุแสดงข้อมูลในตารางฐานข้อมูลโดยการเรียกใช้ฟังก์ชั่นพร้อมใส่พารามิเตอร์ที่ได้รับค่ามาแล้ว

End Sub

Private Sub FRM\_RAM\_Data\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Search() 'เรียกใช้ฟังก์ชั่นเมื่อหน้าต่างถูกเปิดขึ้นมา

End Sub

Private Sub btcpu\_Click\_1(sender As Object, e As EventArgs) Handles btcpu.Click

Search() 'เรียกใช้ฟังก์ชั่นสำหรับเรียกข้อมูลมาแสดงเมื่อกดปุ่ม

End Sub

Private Sub btadd\_Click(sender As Object, e As EventArgs) Handles btadd.Click

FRM\_RAM\_Insert.ShowDialog() 'แสดงหน้าต่างสำหรับเพิ่มข้อมูลใหม่เมื่อกดปุ่ม

End Sub

Private Sub btdelete\_Click(sender As Object, e As EventArgs) Handles btdelete.Click

If \_Confirm("ต้องการลบหรือไม่") Then 'แสดงข้อความยืนยันพร้อมกำหนดคำในพารามิเตอร์

Dim sql As String = "DELETE FROM [tbl\_ram] WHERE RAM\_Model = '" & txtModel.Text & "'" 'ประกาศตัวแปรสำหรับก็บคำสั่งลบข้อมูลในตารางฐานข้อมูล

\_Delete(sql) 'เรียกใช้ฟังก์ชั่นลบข้อมูลโดยกำหนดตัวแปรที่เก็บคำสั่งในพารามิเตอร์

\_Success("ลบข้อมูลเรียบร้อย") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความพร้อมกำหนดคำในพารามิเตอร์

Search() 'เรียกใช้ฟังก์ชั่นแสดงข้อมูล

End If

End Sub

Private Sub btSearch\_Click(sender As Object, e As EventArgs) Handles btSearch.Click

dt = New DataTable 'กำหนดคุณสมบัติให้ตัวแปร

ds = New DataSet 'กำหนดคุณสมบัติให้ตัวแปร

ds.Tables.Add(dt) 'เก็บข้อมูลตารางที่ได้รับไว้ในตัวแปร

da = New OleDbDataAdapter($"SELECT \* FROM tbl\_cpu WHERE CPU\_Brand = '" & txtSearch.Text & "'", cn) 'ให้ตัวแปรเก็บคำสั่ง sql สำหรับค้นหาข้อมูล

da.Fill(dt) 'นำตัวแปรที่เก็บคำสั่งใปใช้กับตัวแปรที่เก็บตารางฐานข้อมูล

gridram.DataSource = dt.DefaultView 'ให้วัตถุแสดงข้อมูลที่อยู่ในตารางที่ผ่านการกรองมาแล้ว

End Sub

Private Sub gridram\_CellMouseUp(sender As Object, e As DataGridViewCellMouseEventArgs) Handles gridram.CellMouseUp

txtModel.Text = gridram.Rows.Item(e.RowIndex).Cells(1).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

txtBrand.Text = gridram.Rows.Item(e.RowIndex).Cells(2).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

End Sub

End Class

* 1. **FRM\_STO\_Data**

Imports System.Data.OleDb

Imports System.Data

Public Class FRM\_STO\_Data

Dim da As OleDbDataAdapter 'สร้างตัวแปรสำหรับควบคุมคำสั่ง sql

Dim dt As DataTable 'สร้างตัวแปรสำหรับเก็บข้อมูลในตาราง

Dim ds As DataSet 'สร้างตัวแปรสำหรับเก็บเช็ตในตาราง

Private \_sql As String 'ประกาศตัวแปรสำหรับเก็บค่า sql จากภายนอก

Private Sub Search() 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

gridsto.DataSource = \_QueryAsDataTable(\_sql) 'ให้วัตถุแสดงข้อมูลในตารางฐานข้อมูลโดยการเรียกใช้ฟังก์ชั่นพร้อมใส่พารามิเตอร์ที่ได้รับค่ามาแล้ว

End Sub

Public Sub Show\_DATA(ByVal sql As String) 'สร้างฟังก์ชั่นสำหรับเรียกข้อมูล

\_sql = sql 'กำหนดให้ตัวแปรที่สร้างขึ้นมามีค่าเท่ากับ sql ที่ได้รับจากภายนอก

Show() 'แสดงข้อมูล

End Sub

Private Sub FRM\_STO\_Data\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

Search() 'เรียกใช้ฟังก์ชั่นเมื่อหน้าต่างถูกเปิดขึ้นมา

End Sub

Private Sub btcpu\_Click(sender As Object, e As EventArgs) Handles btcpu.Click

Search() 'เรียกใช้ฟังก์ชั่นสำหรับเรียกข้อมูลมาแสดงเมื่อกดปุ่ม

End Sub

Private Sub btadd\_Click(sender As Object, e As EventArgs) Handles btadd.Click

FRM\_STO\_Insert.ShowDialog() 'แสดงหน้าต่างสำหรับเพิ่มข้อมูลใหม่เมื่อกดปุ่ม

End Sub

Private Sub btdelete\_Click(sender As Object, e As EventArgs) Handles btdelete.Click

If \_Confirm("ต้องการลบหรือไม่") Then 'แสดงข้อความยืนยันพร้อมกำหนดคำในพารามิเตอร์

Dim sql As String = "DELETE FROM [tbl\_storage] WHERE STO\_Model = '" & txtModel.Text & "'" 'ประกาศตัวแปรสำหรับก็บคำสั่งลบข้อมูลในตารางฐานข้อมูล

\_Delete(sql) 'เรียกใช้ฟังก์ชั่นลบข้อมูลโดยกำหนดตัวแปรที่เก็บคำสั่งในพารามิเตอร์

\_Success("ลบข้อมูลเรียบร้อย") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความพร้อมกำหนดคำในพารามิเตอร์

Search() 'เรียกใช้ฟังก์ชั่นแสดงข้อมูล

End If

End Sub

Private Sub btSearch\_Click(sender As Object, e As EventArgs) Handles btSearch.Click

dt = New DataTable 'กำหนดคุณสมบัติให้ตัวแปร

ds = New DataSet 'กำหนดคุณสมบัติให้ตัวแปร

ds.Tables.Add(dt) 'เก็บข้อมูลตารางที่ได้รับไว้ในตัวแปร

da = New OleDbDataAdapter($"SELECT \* FROM tbl\_storage WHERE STO\_Brand = '" & txtSearch.Text & "'", cn) 'ให้ตัวแปรเก็บคำสั่ง sql สำหรับค้นหาข้อมูล

da.Fill(dt) 'นำตัวแปรที่เก็บคำสั่งใปใช้กับตัวแปรที่เก็บตารางฐานข้อมูล

gridsto.DataSource = dt.DefaultView 'ให้วัตถุแสดงข้อมูลที่อยู่ในตารางที่ผ่านการกรองมาแล้ว

End Sub

Private Sub gridsto\_CellMouseUp(sender As Object, e As DataGridViewCellMouseEventArgs) Handles gridsto.CellMouseUp

txtModel.Text = gridsto.Rows.Item(e.RowIndex).Cells(1).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

txtBrand.Text = gridsto.Rows.Item(e.RowIndex).Cells(2).Value.ToString 'กำหนดให้กล่องข้อความแสดงค่าในตารางฐานข้อมูลเมื่อคลิกในตาราง

End Sub

End Class

* 1. **FRM\_CASE\_Insert**

Public Class FRM\_CASE\_Insert

Private Sub btsave\_Click(sender As Object, e As EventArgs) Handles btsave.Click

Dim sql As String = $"INSERT INTO [tbl\_case]([CASE\_Model],[CASE\_Brand],[CASE\_Type],[CASE\_Mainboard],[CASE\_Material],[CASE\_PSU]) VALUES('{txtCASEMODEL.Text}','{txtCASEBrand.Text}','{txtCASEType.Text}','{txtCASEMB.Text}','{txtCASEMat.Text}','{txtCASEPSU.Text}')" 'ประกาศตัวแปรสำหรับเก็บคำสั่งเพิ่มข้อมูล

If \_InsertAdd(sql) Then 'เรียกใช้ฟังก์ชั่นเพิ่มข้อมูลโดยใส่ตัวแปรที่เก็บคำสั่งเพิ่มข้อมูลในพารามิเตอร์

\_Success("Saved") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกสำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

Else

\_Error("Error") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกไม่สำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

End If

End Sub

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

Close() 'ปิดหน้าต่าง

End Sub

Private Sub FRM\_CASE\_Insert\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

End Class

* 1. **FRM\_CPU\_Insert**

Public Class FRM\_CPU\_Insert

Private Sub btsave\_Click(sender As Object, e As EventArgs) Handles btsave.Click

Dim sql As String = $"INSERT INTO [tbl\_cpu]([CPU\_Model],[CPU\_Brand],[CPU\_Socket],[CPU\_Core],[CPU\_Thread],[CPU\_Frequency],[CPU\_Turbo]) VALUES('{txtCPUMODEL.Text}','{txtCPUBrand.Text}','{txtCPUSocket.Text}','{txtCPUCore.Text}','{txtCPUThread.Text}','{txtCPUFrequency.Text}','{txtCPUTurbo.Text}')" 'ประกาศตัวแปรสำหรับเก็บคำสั่งเพิ่มข้อมูล

If \_InsertAdd(sql) Then 'เรียกใช้ฟังก์ชั่นเพิ่มข้อมูลโดยใส่ตัวแปรที่เก็บคำสั่งเพิ่มข้อมูลในพารามิเตอร์

\_Success("Saved") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกสำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

Else

\_Error("Error") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกไม่สำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

End If

End Sub

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

Close() 'ปิดหน้าต่าง

End Sub

Private Sub FRM\_CPU\_Insert\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

End Sub

End Class

* 1. **FRM\_MB\_Insert**

Public Class FRM\_MB\_Insert

Private Sub btsave\_Click(sender As Object, e As EventArgs) Handles btsave.Click

Dim sql As String = $"INSERT INTO [tbl\_mainboard]([MB\_Model],[MB\_Brand],[MB\_Socket],[MB\_Chipset],[MB\_SlotRam],[MB\_RamType],[MB\_SATA2],[MB\_SATA3],[MB\_M2SSD],[MB\_USB2],[MB\_USB3],[MB\_FormFactor]) VALUES('{txtMBMODEL.Text}','{txtMBBrand.Text}','{txtMBSocket.Text}','{txtMBChip.Text}','{txtMBSlot.Text}','{txtMBRType.Text}','{txtSATA2.Text}','{txtSATA3.Text}','{txtM2.Text}','{txtUSB2.Text}','{txtUSB3.Text}','{txtMBSize.Text}')" 'ประกาศตัวแปรสำหรับเก็บคำสั่งเพิ่มข้อมูล

If \_InsertAdd(sql) Then 'เรียกใช้ฟังก์ชั่นเพิ่มข้อมูลโดยใส่ตัวแปรที่เก็บคำสั่งเพิ่มข้อมูลในพารามิเตอร์

\_Success("Saved") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกสำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

Else

\_Error("Error") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกไม่สำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

End If

End Sub

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

Close() 'ปิดหน้าต่าง

End Sub

End Class

* 1. **FRM\_GPU\_Insert**

Public Class FRM\_GPU\_Insert

Private Sub btsave\_Click(sender As Object, e As EventArgs) Handles btsave.Click

Dim sql As String = $"INSERT INTO [tbl\_gpu]([GPU\_Model],[GPU\_Brand],[GPU\_GSpeed],[GPU\_RAM],[GPU\_RamType],[GPU\_VGA],[GPU\_DVI],[GPU\_HDMI],[GPU\_Display]) VALUES('{txtGPUMODEL.Text}','{txtGPUBrand.Text}','{txtGPUSpeed.Text}','{txtGPURam.Text}','{txtGPURType.Text}','{txtGPUVGA.Text}','{txtGPUDVI.Text}','{txtGPUHDMI.Text}','{txtGPUDisplay.Text}')" 'ประกาศตัวแปรสำหรับเก็บคำสั่งเพิ่มข้อมูล

If \_InsertAdd(sql) Then 'เรียกใช้ฟังก์ชั่นเพิ่มข้อมูลโดยใส่ตัวแปรที่เก็บคำสั่งเพิ่มข้อมูลในพารามิเตอร์

\_Success("Saved") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกสำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

Else

\_Error("Error") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกไม่สำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

End If

End Sub

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

Close() 'ปิดหน้าต่าง

End Sub

End Class

* 1. **FRM\_RAM\_Insert**

Public Class FRM\_RAM\_Insert

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

Close() 'ปิดหน้าต่าง

End Sub

Private Sub btsave\_Click(sender As Object, e As EventArgs) Handles btsave.Click

Dim sql As String = $"INSERT INTO [tbl\_ram]([RAM\_Model],[RAM\_Brand],[RAM\_PartNo],[RAM\_Type],[RAM\_Speed],[RAM\_TCapacity],[RAM\_CapacityPerDIMM]) VALUES('{txtRAMMODEL.Text}','{txtRAMBrand.Text}','{txtRAMPartNo.Text}','{txtRAMType.Text}','{txtRAMSpeed.Text}','{txtRAMTC.Text}','{txtRAMCPD.Text}')" 'ประกาศตัวแปรสำหรับเก็บคำสั่งเพิ่มข้อมูล

If \_InsertAdd(sql) Then 'เรียกใช้ฟังก์ชั่นเพิ่มข้อมูลโดยใส่ตัวแปรที่เก็บคำสั่งเพิ่มข้อมูลในพารามิเตอร์

\_Success("Saved") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกสำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

Else

\_Error("Error") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกไม่สำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

End If

End Sub

End Class

* 1. **FRM\_PSU\_Insert**

Public Class FRM\_PSU\_Insert

Private Sub btsave\_Click(sender As Object, e As EventArgs) Handles btsave.Click

Dim sql As String = $"INSERT INTO [tbl\_psu]([PSU\_Model],[PSU\_Series],[PSU\_Brand],[PSU\_Power],[PSU\_FanSize],[PSU\_MB],[PSU\_CPU],[PSU\_PCI],[PSU\_Sata],[PSU\_Molex],[PSU\_ISO],[PSU\_Input]) VALUES('{txtPSUMODEL.Text}','{txtPSUSeries.Text}','{txtPSUBrand.Text}','{txtPSUWatt.Text}','{txtPSUFan.Text}','{txtPSUMB.Text}','{txtPSUCPU.Text}','{txtPSUPCI.Text}','{txtPSUSATA.Text}','{txtPSUMolex.Text}','{txtPSUISO.Text}','{txtPSUInput.Text}')" 'ประกาศตัวแปรสำหรับเก็บคำสั่งเพิ่มข้อมูล

If \_InsertAdd(sql) Then 'เรียกใช้ฟังก์ชั่นเพิ่มข้อมูลโดยใส่ตัวแปรที่เก็บคำสั่งเพิ่มข้อมูลในพารามิเตอร์

\_Success("Saved") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกสำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

Else

\_Error("Error") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกไม่สำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

End If

End Sub

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

Close() 'ปิดหน้าต่าง

End Sub

End Class

* 1. **FRM\_STO\_Insert**

Public Class FRM\_STO\_Insert

Private Sub btsave\_Click(sender As Object, e As EventArgs) Handles btsave.Click

Dim sql As String = $"INSERT INTO [tbl\_storage]([STO\_Model],[STO\_Brand],[STO\_Size]) VALUES('{txtSTOMODEL.Text}','{txtSTOBrand.Text}','{txtSTOSize.Text}')" 'ประกาศตัวแปรสำหรับเก็บคำสั่งเพิ่มข้อมูล

If \_InsertAdd(sql) Then 'เรียกใช้ฟังก์ชั่นเพิ่มข้อมูลโดยใส่ตัวแปรที่เก็บคำสั่งเพิ่มข้อมูลในพารามิเตอร์

\_Success("Saved") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกสำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

Else

\_Error("Error") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกไม่สำเร็จพร้อมกำหนดข้อความลงในพารามิเตอร์

End If

End Sub

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

Close() 'ปิดหน้าต่าง

End Sub

End Class

* 1. **FRM\_ADDUSER**

Public Class FRM\_ADDUSER

Private Sub btexit\_Click(sender As Object, e As EventArgs) Handles btexit.Click

Close() 'คำสั่งปิดหน้าต่าง

End Sub

Private Sub btsave\_Click(sender As Object, e As EventArgs) Handles btsave.Click

If txtUsername.Text = "" Then 'ตรวจสอบว่า textbox username ว่างหรือไม่

MessageBox.Show("กรุณาใส่รหัสอุปกรณ์") 'แสดงกล่องข้อความเตือน

Else

Dim sql As String = $"INSERT INTO [tbl\_admin]([username],[password],[fullname]) VALUES('{txtUsername.Text}','{txtPassword.Text}','{txtName.Text}')" 'คำสั่ง sql สำหรับเพิ่มข้อมูลลงตาราง

If \_InsertAdd(sql) Then 'เรียกใช้ฟังก์ชั่นกำหนดค่าพารามิเตอร์จาก sql

\_Success("Saved") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกสำเร็จ

Else

\_Error("Error") 'เรียกใช้ฟังก์ชั่นแสดงกล่องข้อความเมื่อบันทึกไม่สำเร็จ

End If

End If

End Sub

End Class