C#期末專題報告

NCKU-甲崩

目錄

系級:資訊碩一

學號:P76081108

姓名:蘇奕瑋

* 摘要
* 專案功能介紹
* 程式碼分析
* 未來展望

摘要

* 研究動機:

之前實驗室的同學有在閒暇之餘練習寫類似的軟體，不過是用Python寫LINEBOT機器人的版本。所以起心動念想要把這個想法移植到C# Winform的版本試試看

* 程式概略:

本報告採用:

1. C# WinForm視窗程式呈現
2. 資料庫處理部分採用LINQ和Entity Framework之實體資料模型

專案功能介紹

* 專案程式版面呈現:



B

A

* 功能介紹:

A.



pnl\_operation

btn\_show\_on\_map

btn\_AllRestaurant

btn\_Update

btn\_Remove

btn\_Add

Txt\_Restaurant\_Name

Txt\_RestaurantID

Txt\_Address

Txt\_Lng

Txt\_Distance

Txt\_Lat

Txt\_Category

* pnl\_operation:

將以上控制項包成一個群組,可以一起拖曳

* btn\_Add:

1. 先將欲新增的餐廳資料(距離的欄位可以不用,會有函式自動計算之)鍵入在左邊Textbox中
2. 按下”新增餐廳”的按鍵,按下去會將左邊Textbox的資料新增成一筆餐廳的資料加入資料庫中

* btn\_Remove:

1. 先在Txt\_RestaurantID鍵入欲移除的餐廳之編號
2. 按下”移除餐廳”的按鈕,會根據左邊Txt\_RestaurantID移除對應的餐廳資料

* btn\_Update:

1. 先在Txt\_RestaurantID鍵入欲更新的餐廳之編號
2. 按下”更新餐廳”的按鈕,會根據左邊Textbox的資料更新到對應之餐廳的資料庫欄位

* btn\_AllRestaurant:

1. 按下”顯示所有餐廳”的按鈕,會在另一個名為AllRestaurant.cs的WinForm顯示出目前資料庫中所有餐廳的資料

* btn\_show\_on\_map:

1. 先在Txt\_RestaurantID鍵入欲更新的餐廳之編號
2. 按下”在地圖中顯示”的按鈕,會在右下方名為web\_map的webbrowser中連結到google map,並且根據經緯度show出ID對應之餐廳的位置

B.

rdb\_select\_random

rdb\_showall

btn\_select

Txt\_condition\_distance

groupBox\_selection

web\_map

DGV\_selected\_restaurant



* groupBox\_selection:

將兩個radiobutton結合成一個group

* rdb\_showall:

1. 點選此radiobutton
2. 在Txt\_condition\_distance輸入條件:範圍為?公尺內
3. 按下btn\_select的”挑選餐廳按鈕”
4. DGV\_selected\_restaurant的datagirdview中會顯示所有符合條件的餐廳列表的資料庫資料,並且依照距離由近到遠排列
5. Web\_map會顯示出第一筆資料的Google map位置

* rdb\_showall:

1. 點選此radiobutton
2. 在Txt\_condition\_distance輸入條件:範圍為?公尺內
3. 按下btn\_select的”挑選餐廳按鈕”
4. DGV\_selected\_restaurant的datagirdview中會從所有符合條件的餐廳中隨機挑取一筆資料,並顯示在datagridview中
5. Web\_map會顯示出該筆資料的Google map位置

程式碼分析

* Menu.cs:

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;

using System.Data.SqlClient;

namespace NCKU\_Eat

{

public partial class Menu : Form

{

//地球半徑，單位:m

private const double EARTH\_RADIUS = 6378137;

//資工系館的緯度/經度

PointF PositionCSIE = new PointF(22.996992f,120.221510f);

//建立RetaurantEntities物件,此物件可以管理Restaurant.mdf資料庫

RestaurantEntities resDB = new RestaurantEntities();

public Menu()

{

InitializeComponent();

}

//計算兩點距離位置，單位：m

//公式為google提供誤差<0.2m

//(lat1,lng1)=第一點之緯經度

//(lat2,lng2)=第二點之緯經度

public static float GetDistance(float lat1, float lng1, float lat2,float lng2)

{

float radLat1 = Rad(lat1);

float radLng1 = Rad(lng1);

float radLat2 = Rad(lat2);

float radLng2 = Rad(lng2);

float a = radLat1 - radLat2;

float b = radLng1 - radLng2;

float result = (float)(2 \* Math.Asin(Math.Sqrt(Math.Pow(Math.Sin(a / 2), 2) + Math.Cos(radLat1) \* Math.Cos(radLat2) \* Math.Pow(Math.Sin(b / 2), 2))) \* EARTH\_RADIUS);

return result;

}

//經緯度轉弧度

private static float Rad(float d)

{

return (float)(d \* Math.PI / 180d);

}

//”顯示所有餐廳”按鍵被按下的事件事件處理

private void btn\_AllRestaurant\_Click(object sender, EventArgs e)

{

//

AllRestaurant allRestaurant = new AllRestaurant();

allRestaurant.Show();

}

//”新增餐廳”按鍵被按下的事件處理

private void btn\_Add\_Click(object sender, EventArgs e)

{

try

{

//利用Entity Framework來簡化對資料庫的操作

餐廳 newres = new 餐廳();

newres.Id = int.Parse(Txt\_RestaurantID.Text);

newres.店名 = Txt\_Restaurant\_Name.Text;

newres.地址 = Txt\_Address.Text;

//利用Round取小數點後6位加入資料庫

newres.緯度 = System.Math.Round(float.Parse(Txt\_Lat.Text),6);

newres.經度 =System.Math.Round(float.Parse(Txt\_Lng.Text),6);

newres.距離 = System.Math.Round(GetDistance((float)newres.緯度, (float)newres.經度, PositionCSIE.X, PositionCSIE.Y),2);

newres.餐廳類別 = Txt\_Category.Text;

resDB.餐廳.Add(newres);

//保存異動

resDB.SaveChanges();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btn\_Remove\_Click(object sender, EventArgs e)

{

DialogResult result;

result = MessageBox.Show("確定要刪除此餐廳?", "警告", MessageBoxButtons.YesNo);

if (result == DialogResult.Yes)

{

try

{

//利用LINQ鏈式寫法取得ID對應的餐廳

var removed = resDB.餐廳.Where(r => r.Id.ToString() == Txt\_RestaurantID.Text).FirstOrDefault();

//找不到餐廳則跳走

if (removed == null)

return;

//移除找到的餐廳

resDB.餐廳.Remove(removed);

resDB.SaveChanges();

MessageBox.Show("餐廳:" + removed.店名 + "已被刪除", "已刪除餐廳", MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

private void btn\_Update\_Click(object sender, EventArgs e)

{

try

{

//利用LINQ鏈式寫法取得ID對應的餐廳

var updated = resDB.餐廳.Where(r => r.Id.ToString() == Txt\_RestaurantID.Text).FirstOrDefault();

if (updated == null)

return;

updated.店名 = Txt\_Restaurant\_Name.Text;

updated.地址 = Txt\_Address.Text;

updated.緯度 = System.Math.Round(float.Parse(Txt\_Lat.Text), 6);

updated.經度 = System.Math.Round(float.Parse(Txt\_Lng.Text), 6);

updated.距離 = System.Math.Round(GetDistance((float)updated.緯度, (float)updated.經度, PositionCSIE.X, PositionCSIE.Y), 2);

updated.餐廳類別 = Txt\_Category.Text;

resDB.SaveChanges();

MessageBox.Show("餐廳:" + updated.店名 + "已被更新", "已更新餐廳", MessageBoxButtons.OKCancel, MessageBoxIcon.Asterisk);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void btn\_select\_Click(object sender, EventArgs e)

{

//防止Txt\_condition\_distance.Text為空時會出錯

if (Txt\_condition\_distance.Text == "")

return;

//讀入限制的範圍

float distance\_condition=float.Parse(Txt\_condition\_distance.Text);

//如果選擇顯示全部餐廳

if (rdb\_showall.Checked==true)

{

//取得所有距離內的餐廳,依照距離排列

var selected\_result = from r in resDB.餐廳

where r.距離<=distance\_condition

orderby r.距離

select r;

DGV\_selected\_restaurant.DataSource = selected\_result.ToList();

MessageBox.Show("只會顯示第一個餐廳之位置","提示",MessageBoxButtons.OKCancel,MessageBoxIcon.Asterisk);

//Take(1)代表只取selected\_result的前1筆資料

var showed = (from r in selected\_result select r).Take(1);

//將第一筆餐廳的經緯度丟去callwebbrowser開啟Google map

foreach (var s in showed)

{

callwebbrowser(s.緯度.ToString(), s.經度.ToString());

}

}

//如果要隨機抽餐廳

if (rdb\_select\_random.Checked == true)

{

//隨機&取前n筆資料的方法:https://www.cnblogs.com/yuandotnet/archive/2012/06/07/2539938.html

var selected\_result = (from r in resDB.餐廳

where r.距離 <= distance\_condition

orderby Guid.NewGuid()

select r).Take(1);

DGV\_selected\_restaurant.DataSource = selected\_result.ToList();

foreach (var s in selected\_result)

{

callwebbrowser(s.緯度.ToString(),s.經度.ToString());

}

}

}

private void btn\_show\_on\_map\_Click(object sender, EventArgs e)

{

try

{

var toshowed = resDB.餐廳.Where(r => r.Id.ToString() == Txt\_RestaurantID.Text).FirstOrDefault();

if (toshowed == null)

return;

Txt\_Restaurant\_Name.Text=toshowed.店名;

Txt\_Address.Text = toshowed.地址;

Txt\_Lng.Text = toshowed.經度.ToString();

Txt\_Lat.Text = toshowed.緯度.ToString();

Txt\_Distance.Text=toshowed.距離.ToString();

Txt\_Category.Text = toshowed.餐廳類別;

callwebbrowser( Txt\_Lat.Text,Txt\_Lng.Text);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void callwebbrowser(string lag, string lng)

{

//依照下面公式可以在google map顯示該經緯度的位置

string url = "http://maps.google.com/maps?q=" + lag + "%2c" + lng;

web\_map.Url = new Uri(url);

}

}

}

* AllRestaurant.cs:

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;

using System.Data.SqlClient;

namespace NCKU\_Eat

{

public partial class AllRestaurant : Form

{

RestaurantEntities re = new RestaurantEntities();

public AllRestaurant()

{

InitializeComponent();

}

private void AllRestaurant\_Load(object sender, EventArgs e)

{

DGV\_AllRestaurant.DataSource = re.餐廳.ToList();

}

}

}

未來展望



當呼叫callwebbrowser的副程式開啟網站的時候會有網頁指令碼錯誤的問題,經過幾番嘗試之後仍然無法解決,未來有機會再想辦法解決

之後有機會可以改成ASP.NET MVC版本,對於google map api會有原廠api support.