**PROJECT 2**

Source Code

Billy Kong

**Form1.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 MySql.Data.MySqlClient;

namespace Project\_2

{

public partial class P2 : Form

{

private MySqlCommand cmd; //initialize cmd inputter for database

private MySqlDataReader read; //initialize reader for database

private MySqlConnection con; //initialize the connector for database

public P2() //constructor

{

InitializeComponent();

con = new MySqlConnection("SERVER=localhost;DATABASE=properties;UID=ricefieldboy;PASSWORD=Superhi2;"); //Creates a link to the SQL Database

}

private void P2\_Load(object sender, EventArgs e)

{

try//try statements

{

con.Open(); // opens the connection

}

catch //if fail, execute below

{

con.Close(); //close the connection

MessageBox.Show("ERROR IN DATABASE CONNECTION! CLOSING PROGRAM NOW ");//messagebox displaying error in connecting to the server

this.Close();//closes the application entireley

}

cmd = new MySqlCommand("SELECT address FROM listings", con); //Sytax, select all "address" from "listings" which is the database

read = cmd.ExecuteReader(); //executes the command

if (read.HasRows) //If there are rows to read in

{

while (read.Read()) //While there are more addresses to read

{

propertylist.Items.Add(read[0].ToString()); //add the address to the combobox, named "propertylist". [0] is the directory of the address

}

}

MessageBox.Show("Listings added!"); //Informs that listings have been added, will remove this in the future

read.Close(); //close the reader, this must be done or errors will incur

}

private void propertylist\_SelectedIndexChanged(object sender, EventArgs e)

{

cmd = new MySqlCommand("SELECT \* FROM listings where address='" + this.propertylist.SelectedItem.ToString() + "'", con); //Gather all data when address is equal to the address selected

read = cmd.ExecuteReader();

if (read.HasRows) //if there are rows to be read

{//Text of appropriate labels will be filled with the information of the address

read.Read(); //start reading

address.Text = read[0].ToString();

idnum.Text = read[1].ToString();

city.Text = read[2].ToString();

state.Text = read[3].ToString();

zip.Text = read[4].ToString();

description.Text = read[5].ToString();

status.Text = read[6].ToString();

buy.Text = read[7].ToString();

rent.Text = read[8].ToString();

lotsize.Text = read[9].ToString();

housesize.Text = read[10].ToString();

proptype.Text = read[11].ToString();

bedrooms.Text = read[12].ToString();

bathrooms.Text = read[13].ToString();

garage.Text = read[14].ToString();

street.Text = read[15].ToString();

basement.Text = read[17].ToString();

stories.Text = read[16].ToString();

age.Text = read[18].ToString();

//age.Text = read.GetString("age"); //different syntax, same result //for reference

read.Close(); //close read

}

}

private void purchase\_Click(object sender, EventArgs e) //the purchase button

{//Once pressed, a messagebox will prompt for a confirmation, options yes or no

DialogResult confirm = MessageBox.Show("Are you sure you want to purchase this property?", "CONFIRM YOUR PURCHASE", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (confirm == DialogResult.Yes && status.Text == "Active") //if yes has been selected

{

ccForm childForm4 = new ccForm(this.address.Text, this); //passes the address so the form can utilize it

childForm4.Show();//Opens up the credit card prompt form

}

else if (confirm == DialogResult.Yes && status.Text == "Sold")

MessageBox.Show("This property is unavailable for purchase!", "WARNING!");

if (confirm == DialogResult.No)

MessageBox.Show("Purchase canceled", "Cancel");

}

private void reset\_Click(object sender, EventArgs e) //reset button resets all statuses back to Active

{

cmd = new MySqlCommand("update listings set status='Active'", con);

read = cmd.ExecuteReader();

MessageBox.Show("All listings have been reset"); //prompts user that the listings have been reset

read.Close();

status.Text = "Active";

}

private void renewsel\_Click(object sender, EventArgs e) //reset selection button resets the status of selected property

{

cmd = new MySqlCommand("update listings set status='Active' where address='" + this.address.Text + "'", con);

read = cmd.ExecuteReader();

MessageBox.Show("Selected listing has been reset"); //prompts user of such

read.Close();

status.Text = "Active";

}

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

{

oForm childForm = new oForm(this.address.Text); //creates "childForm" of form2, childForm pass the string address of selected property

childForm.Show(); //Displays form2

}

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

{

nameForm childForm2 = new nameForm(this.address.Text);

childForm2.Show();

}

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

{

realtorform childForm3 = new realtorform(this.address.Text);

childForm3.Show();

}

private void filter\_SelectedIndexChanged(object sender, EventArgs e)

{

if (filter.SelectedItem.ToString() == "all")

{

propertylist.Items.Clear();

cmd = new MySqlCommand("SELECT address FROM listings", con); //Sytax, select all "address" from "listings" which is the database

read = cmd.ExecuteReader(); //executes the command

if (read.HasRows) //If there are rows to read in

{

while (read.Read()) //While there are more addresses to read

{

propertylist.Items.Add(read[0].ToString()); //add the address to the combobox, named "propertylist". [0] is the directory of the address

}

}

read.Close();

}

else

{

cmd = new MySqlCommand("select address from listings where state='" + filter.SelectedItem.ToString() + "'", con);

read = cmd.ExecuteReader();

if (read.HasRows)

{

propertylist.Items.Clear();

while (read.Read())

{

propertylist.Items.Add(read[0].ToString());

}

}

read.Close();

}

}

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

{

Form6 childform5 = new Form6();

childform5.Show();

}

}

}

**Form2.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 MySql.Data.MySqlClient;

namespace Project\_2

{

public partial class oForm : Form

{

private MySqlCommand cmd;

private MySqlDataReader read;

private MySqlConnection con;

public string test;

public oForm(string s) //important to pass string address text because it is needed to identify which offers are made on which listings

{//Same syntax to establish the connector for the database from form1

InitializeComponent();

con = new MySqlConnection("SERVER=localhost;DATABASE=properties;UID=ricefieldboy;PASSWORD=Superhi2;");

con.Open();

cmd = new MySqlCommand("SELECT \* FROM listings where address='" + s + "'", con); //selects all rows from listings where address is 's', the string address

read = cmd.ExecuteReader();

if (read.HasRows)

{//name and offers will be added

test = s;

read.Read();

name1.Text = read[19].ToString();

name2.Text = read[20].ToString();

name3.Text = read[21].ToString();

name4.Text = read[22].ToString();

name5.Text = read[23].ToString();

offer1.Text = read[24].ToString();

offer2.Text = read[25].ToString();

offer3.Text = read[26].ToString();

offer4.Text = read[27].ToString();

offer5.Text = read[28].ToString();

name6.Text = read[38].ToString();

offer6.Text = read[37].ToString();

read.Close();

}

}

private void submit\_Click(object sender, EventArgs e) //once submit has been clicked

{ //the labels will take the text of the textbox

name6.Text = name.Text;

offer6.Text = offer.Text;

cmd = new MySqlCommand("update listings set yourname='" + name6.Text + "' where address='" + test + "'", con);

read = cmd.ExecuteReader();

read.Close();

cmd = new MySqlCommand("update listings set youroffer='" + offer6.Text + "' where address='" + test + "'", con);

read = cmd.ExecuteReader();

read.Close();

}

}

}

**Form3.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 MySql.Data.MySqlClient;

namespace Project\_2

{

public partial class nameForm : Form

{

private MySqlCommand cmd;

private MySqlDataReader read;

private MySqlConnection con;

public nameForm(string s)

{

InitializeComponent();

con = new MySqlConnection("SERVER=localhost;DATABASE=properties;UID=ricefieldboy;PASSWORD=Superhi2;");

con.Open();

cmd = new MySqlCommand("select \* from listings where address='" + s + "'", con);

read = cmd.ExecuteReader();

if (read.HasRows)

{ //same setup from previous forms, sets strings to appropriate labels in the "OWNER DETAILS" form

read.Read();

address.Text = s;

ownername.Text = read[29].ToString();

phone.Text = read[30].ToString();

email.Text = read[31].ToString();

realtor.Text = read[32].ToString();

read.Close();

}

}

}

}

**Form4.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 MySql.Data.MySqlClient;

namespace Project\_2

{

public partial class realtorform : Form

{

private MySqlCommand cmd;

private MySqlDataReader read;

private MySqlConnection con;

public realtorform(string s)

{

InitializeComponent();

con = new MySqlConnection("SERVER=localhost;DATABASE=properties;UID=ricefieldboy;PASSWORD=Superhi2;");

con.Open();

cmd = new MySqlCommand("select \* from listings where address='" + s + "'", con);

read = cmd.ExecuteReader();

if (read.HasRows)

{//same setup code, give appropriate text to labels in the "REALTOR DETAILS" form

read.Read();

address.Text = s;

rname.Text = read[32].ToString();

rphone.Text = read[33].ToString();

remail.Text = read[34].ToString();

rwebsite.Text = read[35].ToString();

rcomments.Text = read[36].ToString();

read.Close();

}

}

}

}

**Form5.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 MySql.Data.MySqlClient;

//This Form contains inheritance/polymorphism requirement

namespace Project\_2

{

public partial class ccForm : Form

{

private MySqlCommand cmd;

private MySqlDataReader read;

private MySqlConnection con;

public string test;

P2 objs;

public ccForm(string s, P2 obj)

{ //same code as previously, note that 's' passed to this constructor has been given to test

InitializeComponent();

con = new MySqlConnection("SERVER=localhost;DATABASE=properties;UID=ricefieldboy;PASSWORD=Superhi2;");

con.Open();

objs = obj;

test = s; //public string test recieves the value of 's', an address

//Adds click events to textboxes - name, cc, csc, and expdate

name.Click += new EventHandler(name\_Click);

cc.Click += new EventHandler(cc\_Click);

csc.Click += new EventHandler(csc\_Click);

expdate.Click += new EventHandler(expdate\_Click);

}

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

{

expdate.Clear(); //This allows the user to clear the textbox once it has been clicked

}

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

{

csc.Clear(); //same code for the following

}

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

{

cc.Clear();

}

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

{

name.Clear();

}

public void classEdit(string type) //parameter is string type, the Credit Card Provider

{//sets label to the credit card type, passed from the inheritance class below ("type" being mastercard, visa etc...)

label1.Text = type;

}

private void mastercard\_CheckedChanged(object sender, EventArgs e) //if mastercard radiobutton is checked

{

CreditCard[] credit = new CreditCard[1]; //create pointer object of CreditCard

credit[0] = new MasterCard(this); //pointer points to MasterCard (a derived class) and passes argument (this), which is this form

credit[0].PrintCard(); //calls the class function, PrintCard

}

//Repeats the same process for each Radio Button checked

private void visa\_CheckedChanged(object sender, EventArgs e)

{

CreditCard[] credit = new CreditCard[1];

credit[0] = new Visa(this);

credit[0].PrintCard();

}

private void americanexpress\_CheckedChanged(object sender, EventArgs e)

{

CreditCard[] credit = new CreditCard[1];

credit[0] = new AmericanExpress(this);

credit[0].PrintCard();

}

private void discovercard\_CheckedChanged(object sender, EventArgs e)

{

CreditCard[] credit = new CreditCard[1];

credit[0] = new DiscoverCard(this);

credit[0].PrintCard();

}

public void submit2\_Click(object sender, EventArgs e)

{//When you click submit in Form5

CreditCard[] credit = new CreditCard[1]; //object pointer is created

credit[0] = new CreditCard(this); //points object of CreditCard

credit[0].name = this.name.Text; //sets data members to the text boxes values

credit[0].ccNum = this.cc.Text;

credit[0].sNum = this.csc.Text;

credit[0].expDate = this.expdate.Text;

//MessageBox will output information stored in CreditCard and not this form to show that values have been passed to the class

DialogResult confirm = MessageBox.Show("Name: " + credit[0].name + "\nCredit Card Number: " + credit[0].ccNum + "\nCVV: " + credit[0].sNum + "\nExpiration Date: " + credit[0].expDate + "\n\n\nConfirm the above information correct?", "Confirm your Credit Card information", MessageBoxButtons.YesNo, MessageBoxIcon.Information);

if (confirm == DialogResult.Yes) //Information confirmation, yes no

{ //if yes, executes command that sets status to sold

cmd = new MySqlCommand("update listings set status='Sold' where address='" + test + "'", con);

read = cmd.ExecuteReader();

MessageBox.Show("Congratulations! You have purchased this property"); //prompts user that they have purchased the property

read.Close(); //close the read

cmd = new MySqlCommand("SELECT address FROM listings", con); //Sytax, select all "address" from "listings" which is the database

read = cmd.ExecuteReader(); //executes the command

objs.status.Text = "Sold";

}

}

}

//inheritance/polymorphism

public class CreditCard //base class

{

public ccForm inst; //public ccForm object, ccForm is Form5

public string name, ccNum, sNum, expDate; //public strings of name, credit card number, security number, and expiration date

public CreditCard() { } // default constructor

public CreditCard(ccForm obj) {inst = obj;} //constructor initializer that takes object of Form5

public virtual void PrintCard() { } //Virtual functions to allow for overriding of PrintCard() function

public virtual void setName(string s)

{//Set name sets name to 's', which is the value in the name textbox

name = s;

}

public virtual void setExp(string s)

{//sets the other data members to textbox values

expDate = s;

}

public virtual void setCC(string s)

{

ccNum = s;

}

public virtual void setSecurity(string s)

{

sNum = s;

}

}

public class MasterCard : CreditCard //derived classes follow

{

public MasterCard(ccForm obj) :base(obj) { } //obj of ccForm is pased from the base class

public override void PrintCard() //overriders PrintCard to print its own card provider in the label when its specific radio button has been checked

{

inst.classEdit("MasterCard");//

}

}

public class Visa : CreditCard

{//same code for the other card providers

public Visa(ccForm obj) :base(obj) { }

public override void PrintCard()

{

inst.classEdit("Visa");

}

}

public class AmericanExpress : CreditCard

{

public AmericanExpress(ccForm obj) :base(obj) { }

public override void PrintCard()

{

inst.classEdit("American Express");

}

}

public class DiscoverCard : CreditCard

{

public DiscoverCard(ccForm obj) :base(obj) { }

public override void PrintCard()

{

inst.classEdit("Discover");

}

}

}

**Form 6.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 MySql.Data.MySqlClient;

namespace Project\_2

{

public partial class Form6 : Form

{

private MySqlCommand cmd;

private MySqlDataReader read;

private MySqlConnection con;

public Form6()

{

InitializeComponent();

con = new MySqlConnection("SERVER=localhost;DATABASE=properties;UID=ricefieldboy;PASSWORD=Superhi2;");

con.Open();

}

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

{

Random random = new Random();

int randomNumber = random.Next (100000, 109999);

string add = one.Text;

string cit = two.Text;

string state = three.Text;

string zip = four.Text;

string desc = five.Text;

string buy = six.Text;

string rent = seven.Text;

string lsize = eight.Text;

string hsize = nine.Text;

string prop = ten.Text;

string bed = eleven.Text;

string bath = twelve.Text;

string garage = thirteen.Text;

string street = fourteen.Text;

string stories = fifteen.Text.ToString();

string basement = sixteen.Text;

string year = seventeen.Text;

string owner = oName.Text;

string ownerPhone = oPhone.Text;

string ownerEmail = oEmail.Text;

string realtor = rName.Text;

string realtorPhone = rPhone.Text;

string realtorEmail = rEmail.Text;

string realtorWebsite = rWebsite.Text;

string realtorComments = rComments.Text;

cmd = new MySqlCommand("insert into listings(address,city,state,zip,description,buyprice,rentprice,lotsize,housesize,propertytype,bedrooms,bathrooms,garage,streetparking,stories,basement,yearbuilt,idNum,owner,phone,email,realtor,rphone,remail,rwebsite,rcomments) values('" + add + "','" + cit + "','"

+ state + "','" + zip + "','" + desc + "','" + buy + "','" + rent + "','" + lsize + "','" + hsize + "','" + prop + "','" + bed + "','" + bath + "','" + garage + "','"

+ street + "','" + stories + "','" + basement + "','" + year + "','" + randomNumber.ToString() + "','" + owner + "','" + ownerPhone + "','" + ownerEmail + "','"

+ realtor + "','" + realtorPhone + "','" + realtorEmail + "','" + realtorWebsite + "','" + realtorComments + "')", con);

int nuRowsAffected = cmd.ExecuteNonQuery(); //execute qeury to database

if (nuRowsAffected < 1)

{

MessageBox.Show("Registration failed", "ERROR");

}

else

{

MessageBox.Show("Property has been succesfully listed,", "Registration");

}

}

}

}

**FORM UML DIAGRAM (rotated)**



**CLASS UML DIAGRAM**

