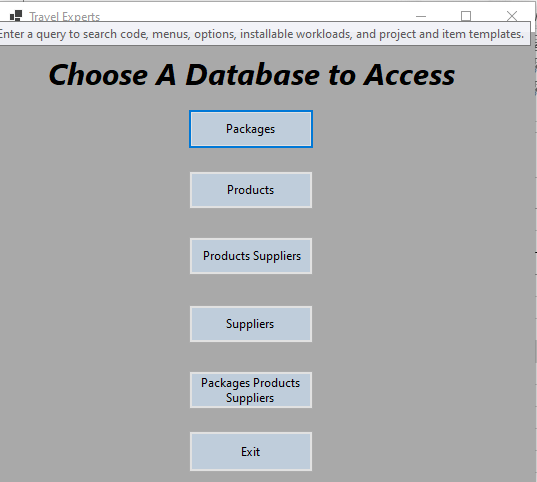
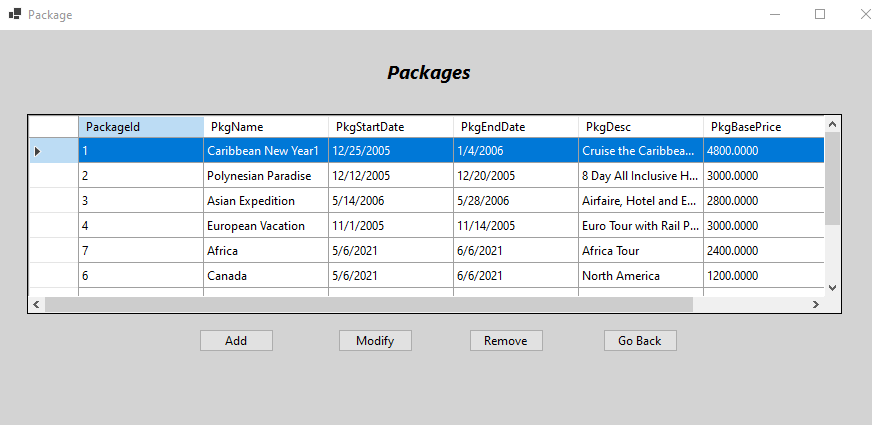
**Workshop4 Travel Experts Application**



Agents can add/edit travel packages



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 WorkShop4.Models;

namespace WorkShop4

{

public partial class PackageProductSupplierFrm : Form

{

TravelExpertsContext context;

PackagesProductsSupplier currentPPS;

public bool buttonstatusadd = false;

public bool viewPPS = false;

public PackageProductSupplierFrm()

{

InitializeComponent();

}

private void Display()

{

context = new TravelExpertsContext();

dataGridPPS.AutoGenerateColumns = false;

dataGridPPS.DataSource = context.PackagesProductsSuppliers.ToList();

}

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

{

Display();

}

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

{

context.PackagesProductsSuppliers.Remove(currentPPS);

context.SaveChanges();

Display();

}

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

{

this.Close();

}

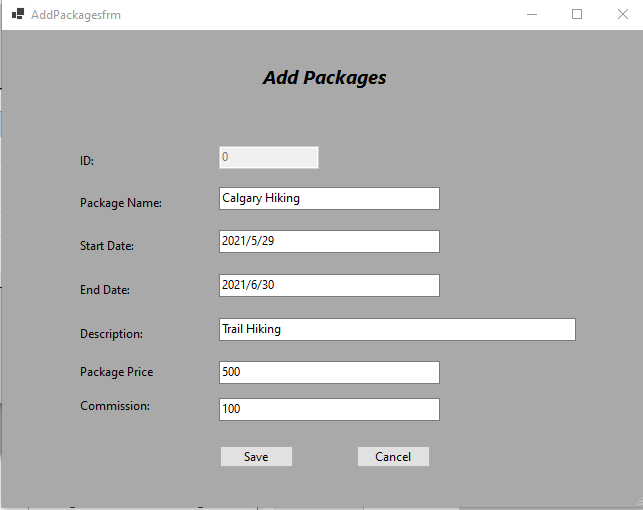
private void addBtn\_Click(object sender, EventArgs e)

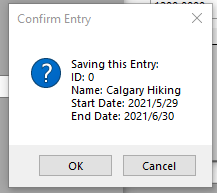
{

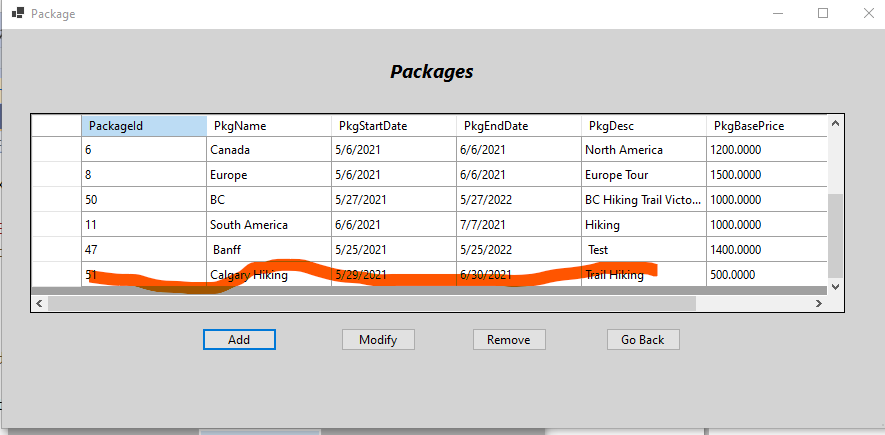
}

}

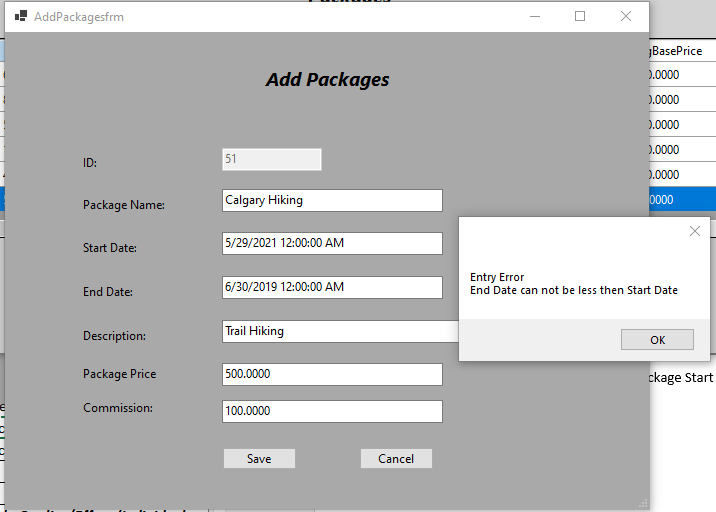
}



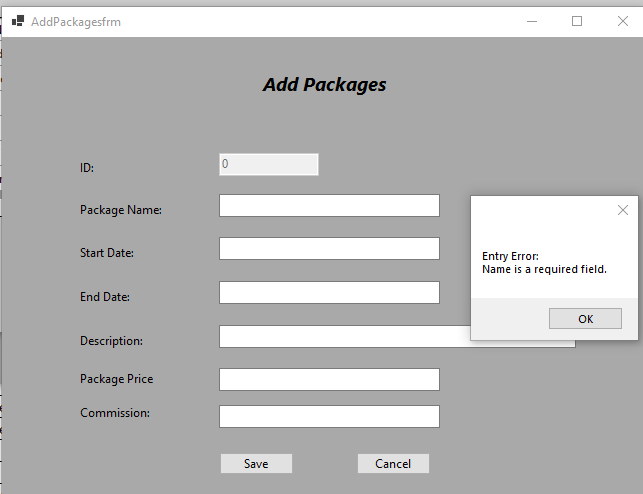




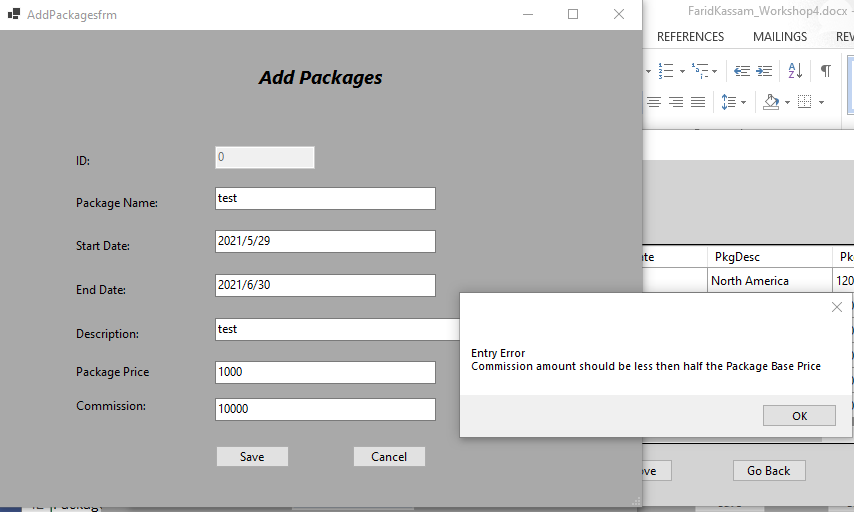
Package End Date must be later than Package Start Date



Package Name and Package Description cannot be null



Package End Date must be later than Package Start Date



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 WorkShop4.Models;

namespace WorkShop4

{

public partial class AddPackagesfrm : Form

{

TravelExpertsContext context = new TravelExpertsContext();

public Package currentPackage { get; set; }

private Package package = null;

public bool buttonstatadd { get; set; }

public AddPackagesfrm()

{

InitializeComponent();

}

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

{

//package = new Package();

if (buttonstatadd == true) //user clicked Add

{

IDtxt.Enabled = false;

IDtxt.Text = "0";

nameTxt.Text = "";

startDateTxt.Text = "";

endDateTxt.Text = "";

descriptionTxt.Text = "";

packagebasepriceTxt.Text = "";

commissionTxt.Text = "";

}

else // user clicked modify

{

IDtxt.Enabled = false;

IDtxt.Text = currentPackage.PackageId.ToString();

nameTxt.Text = currentPackage.PkgName;

startDateTxt.Text = currentPackage.PkgStartDate.ToString();

endDateTxt.Text = currentPackage.PkgEndDate.ToString();

descriptionTxt.Text = currentPackage.PkgDesc;

packagebasepriceTxt.Text = currentPackage.PkgBasePrice.ToString();

commissionTxt.Text = currentPackage.PkgAgencyCommission.ToString();

}

}

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

{

package = new Package();

package.PackageId = Convert.ToInt32(IDtxt.Text);

package.PkgName = nameTxt.Text;

package.PkgStartDate = Convert.ToDateTime(startDateTxt.Text);

package.PkgEndDate = Convert.ToDateTime(endDateTxt.Text);

package.PkgDesc = descriptionTxt.Text;

package.PkgBasePrice = Convert.ToDecimal(packagebasepriceTxt.Text);

package.PkgAgencyCommission = Convert.ToDecimal(commissionTxt.Text);

if (buttonstatadd == true) //add a package

context.Packages.Add(package);

else //modify a package

context.Packages.Update(package);

DialogResult save =

MessageBox.Show(

$"Saving this Entry:\nID: {IDtxt.Text}\nName: {nameTxt.Text}\nStart Date: {startDateTxt.Text}\n" +

$"End Date: {endDateTxt.Text}",

"Confirm Entry",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button2);

if (save == DialogResult.OK)

{

context.SaveChanges();

this.Close();

}

}

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

{

this.Close();

}

private void commissionTxt\_Validating(object sender, CancelEventArgs e)

{

if (packagebasepriceTxt.Text == "")

{

MessageBox.Show("Entry Error\nYou must enter the Package Price, before the commission Price.");

packagebasepriceTxt.Focus();

}

else if (commissionTxt.Text.Length <= 0)

{

MessageBox.Show("Entry Error\nYou must enter a commission rate, Enter 0 if Unknown.");

commissionTxt.Focus();

}

else if (!Decimal.TryParse(commissionTxt.Text, out \_))

{

MessageBox.Show($"Entry Error\nMust be a valid number");

commissionTxt.Focus();

}

else if (Convert.ToInt32(commissionTxt.Text) > Decimal.Parse(packagebasepriceTxt.Text)/2)

{

MessageBox.Show("Entry Error\nCommission amount should be less then half the Package Base Price");

commissionTxt.Focus();

}

}

private void nameTxt\_Validating(object sender, CancelEventArgs e)

{

//if (nameTxt.Text.Length <= 1)

//{

// MessageBox.Show("Entry Error\nPackage name can not be empty ");

// nameTxt.Focus();

//}

if (nameTxt.Text == "")

{

MessageBox.Show($"Entry Error:\nName is a required field.");

nameTxt.Focus();

}

if (nameTxt.Text.Length > 50)

{

MessageBox.Show("The maximum amount of characters for a Product Name is 50");

nameTxt.Focus();

nameTxt.SelectAll();

}

}

private void descriptionTxt\_Validating(object sender, CancelEventArgs e)

{

if (descriptionTxt.Text.Length <= 1 & descriptionTxt.Text.Length > 40)

{

MessageBox.Show("Entry Error\nDescription name can not be empty and must be less than 50 characters.");

descriptionTxt.Focus();

}

}

private void nameTxt\_Validated(object sender, EventArgs e)

{

if (nameTxt.Text.Length > 25 && nameTxt.Text.Length <= 0)

{

MessageBox.Show("Entry Error\nName must have content and not be less than 25");

nameTxt.Focus();

}

}

private void endDateTxt\_Validating\_1(object sender, CancelEventArgs e)

{

if(endDateTxt.Text.Length <= 0)

{

MessageBox.Show("Entry Error\nEnd Date can not be empty");

endDateTxt.Focus();

}

else if (!DateTime.TryParse(endDateTxt.Text, out \_))

{

MessageBox.Show($"Entry Error\nMust be a valid date\nPreferred Format: YYYY/MM/DD");

endDateTxt.Focus();

}

else if (DateTime.Parse(endDateTxt.Text) < DateTime.Parse(startDateTxt.Text))

{

MessageBox.Show("Entry Error\nEnd Date can not be less then Start Date");

endDateTxt.Focus();

}

}

private void startDateTxt\_Validating(object sender, CancelEventArgs e)

{

if (startDateTxt.Text.Length <= 0)

{

MessageBox.Show("Entry Error\nStart Date can not be empty");

startDateTxt.Focus();

}

else if (!DateTime.TryParse(startDateTxt.Text, out \_))

{

MessageBox.Show($"Entry Error\nMust be a valid date\nPreferred Format: YYYY/MM/DD");

startDateTxt.Focus();

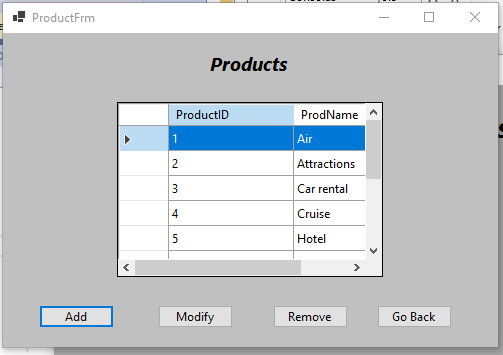
}

}

}

}

Agents can add/edit products



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 WorkShop4.Models;

namespace WorkShop4

{

public partial class ProductFrm : Form

{

TravelExpertsContext context;

Product currentProduct;

public bool buttonclickadd = false;

public ProductFrm()

{

InitializeComponent();

}

private void Display()

{

context = new TravelExpertsContext();

dataGridproduct.AutoGenerateColumns = false;

dataGridproduct.DataSource = context.Products.ToList();

//Display();

}

private void dataGridproduct\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

private void dataGridproduct\_SelectionChanged(object sender, EventArgs e)

{

if (dataGridproduct.SelectedRows.Count > 0)

{

currentProduct = (Product)dataGridproduct.SelectedRows[0].DataBoundItem;

}

}

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

{

context = new TravelExpertsContext();

dataGridproduct.AutoGenerateColumns = false;

dataGridproduct.DataSource = context.Products.ToList();

Display();

}

private Product product = null;

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

{

buttonclickadd = true;

AddProductfrm addProduct = new AddProductfrm();

addProduct.buttonstatusadd = buttonclickadd;

addProduct.ShowDialog();

Display();

}

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

{

DialogResult button =

MessageBox.Show(

"Are you sure that you want to remove the selected record?",

"Remove Item",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button2);

if (button == DialogResult.OK)

{

context.Products.Remove(currentProduct);

context.SaveChanges();

Display();

}

}

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

{

this.Close();

}

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

{

AddProductfrm addProduct = new AddProductfrm();

// ProductFrm addProduct = new ProductFrm();

//addProduct.context = context;

addProduct.currentProduct = currentProduct;

addProduct.buttonstatusadd = false; //for modify button

addProduct.ShowDialog();

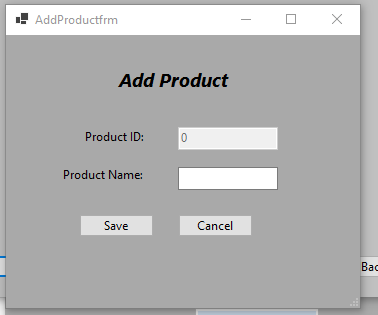
// context.SaveChanges();

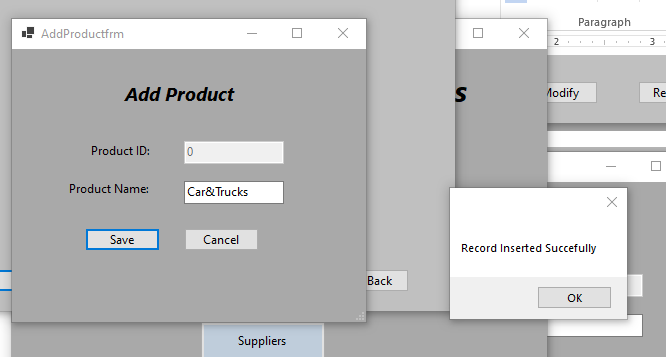
Display();

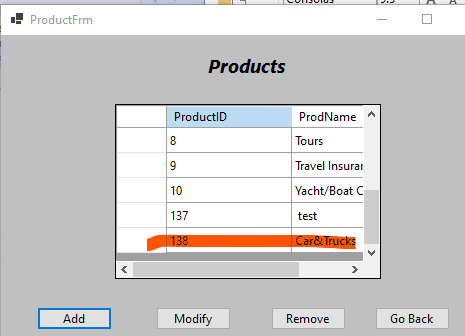
}

}

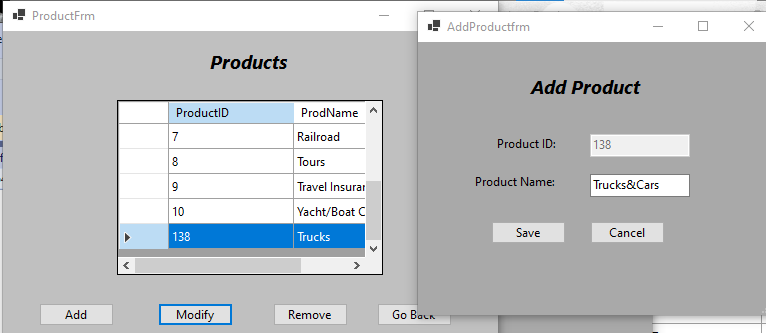
}

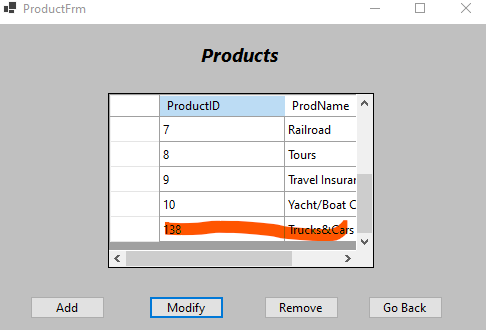






Agents can add/edit suppliers





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 WorkShop4.Models;

namespace WorkShop4

{

public partial class AddProductfrm : Form

{

TravelExpertsContext context = new TravelExpertsContext();

public Product currentProduct { get; set; }

public bool buttonstatusadd { get; set; }

private Product product = null;

public AddProductfrm()

{

InitializeComponent();

}

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

{

if (buttonstatusadd == true) //User Clicked Add

{

IDtxt.Enabled = false;

IDtxt.Text = "0";

productnameTxt.Text = " ";

}

else // user clicked Modify

{

IDtxt.Enabled = false;

IDtxt.Text = currentProduct.ProductId.ToString();

productnameTxt.Text = currentProduct.ProdName;

}

}

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

{

product = new Product

{

ProductId = Convert.ToInt32(IDtxt.Text),

ProdName = productnameTxt.Text

};

if (buttonstatusadd == true) // user clicked Add button

context.Products.Add(product);

else

context.Products.Update(product);

context.SaveChanges();

MessageBox.Show("Record Inserted Succefully");

this.Close();

}

private void IDtxt\_Validated(object sender, EventArgs e)

{

}

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

{

this.Close();

}

private void productnameTxt\_Validating(object sender, CancelEventArgs e)

{

if (productnameTxt.Text.Length == 0 )

{

MessageBox.Show("Product name can not be empty");

productnameTxt.Focus();

}

//else

//{

// productnameTxt.Text = "";

// MessageBox.Show("Product name can not be empty");

//}

}

private void productnameTxt\_Validated(object sender, EventArgs e)

{

if (productnameTxt.Text.Length > 15 )

{

MessageBox.Show("This field must be less than 15");

productnameTxt.Focus();

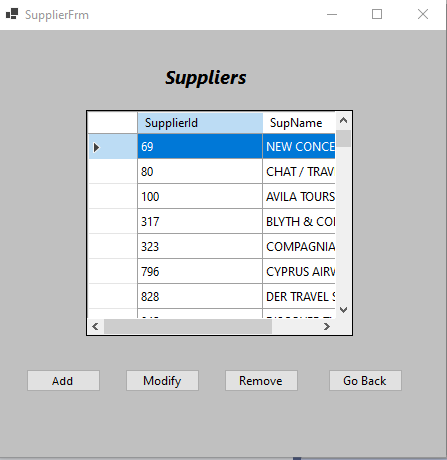
}

}

}

}

Agents can add/edit suppliers



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 WorkShop4.Models;

namespace WorkShop4

{

public partial class SupplierFrm : Form

{

TravelExpertsContext context;

Supplier currentSupplier;

public bool clickAddButton = false;

public SupplierFrm()

{

InitializeComponent();

}

public void Display()

{

context = new TravelExpertsContext();

dataGridViewSupplier.AutoGenerateColumns = false;

dataGridViewSupplier.DataSource = context.Suppliers.ToList();

}

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

{

Display();

}

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

{

clickAddButton = true;

AddSuppliersFrm SupplierData = new AddSuppliersFrm();

SupplierData.AddButton = clickAddButton;

SupplierData.ShowDialog();

Display();

}

private void dataGridViewSupplier\_SelectionChanged(object sender, EventArgs e)

{

if(dataGridViewSupplier.SelectedRows.Count > 0)

{

currentSupplier = (Supplier)dataGridViewSupplier.SelectedRows[0].DataBoundItem;

}

}

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

{

DialogResult button =

MessageBox.Show(

"Are you sure that you want to remove the selected record?",

"Remove Item",

MessageBoxButtons.OKCancel,

MessageBoxIcon.Question,

MessageBoxDefaultButton.Button2);

if (button == DialogResult.OK)

{

context.Suppliers.Remove(currentSupplier);

context.SaveChanges();

Display();

}

}

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

{

this.Close();

}

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

{

AddSuppliersFrm updateSupplier = new AddSuppliersFrm();

//updateSupplier.context = context;

updateSupplier.currentSupplier = currentSupplier;

updateSupplier.AddButton = false;

updateSupplier.ShowDialog();

Display();

//context.SaveChanges();

//Display();

}

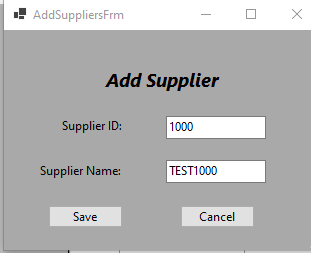
private void dataGridViewSupplier\_CellContentClick(object sender, DataGridViewCellEventArgs e)

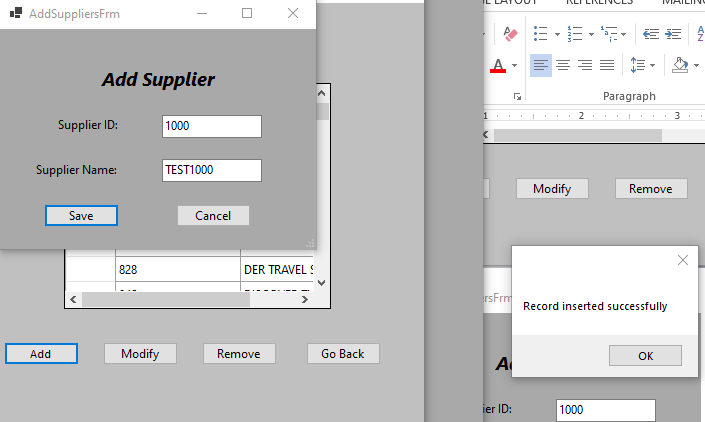
{

}

}

}





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 WorkShop4.Models;

namespace WorkShop4

{

public partial class AddSuppliersFrm : Form

{

public TravelExpertsContext context = new TravelExpertsContext();

public Supplier currentSupplier { get; set; }

private Supplier supplier = null;

public bool AddButton { get; set; }

public AddSuppliersFrm()

{

InitializeComponent();

}

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

{

supplier = new Supplier

{

SupplierId = Convert.ToInt32(SupplierIdTxt.Text),

SupName = supNameTxt.Text

};

if (AddButton == true)

context.Suppliers.Add(supplier);

else

context.Suppliers.Update(supplier);

context.SaveChanges();

MessageBox.Show("Record inserted successfully");

this.Close();

//supplier = new Supplier();

//context.Suppliers.Update(supplier);

//MessageBox.Show("Supplier details updated");

//context.SaveChanges();

//Display();

//this.Close();

//supplier = new Supplier();

//supplier.SupplierId = Convert.ToInt32(SupplierIdTxt.Text);

//supplier.SupName = supNameTxt.Text;

//context.Suppliers.Add(supplier);

//MessageBox.Show("Supplier added");

//context.SaveChanges();

//this.Close();

}

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

{

this.Close();

}

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

{

if (AddButton == true) // user clicked add

{

//SupplierIdTxt.Enabled = false; - this is when id is autogenerated

//SupplierIdTxt.Text = "0"; - this is when id is autogenerated

SupplierIdTxt.Text = "";

supNameTxt.Text = "";

}

else //user clicked Modify

{

SupplierIdTxt.Enabled = false;

SupplierIdTxt.Text = currentSupplier.SupplierId.ToString();

supNameTxt.Text = currentSupplier.SupName;

}

}

private void supNameTxt\_Validating(object sender, CancelEventArgs e)

{

if (supNameTxt.Text.Length <= 1)

{

MessageBox.Show("Supplier name cannot be empty");

supNameTxt.Focus();

}

}

}

}

