using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace DatabaseApplication

{

/// <summary>

/// Interaction logic for MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

//default constructor

public MainWindow()

{

InitializeComponent();

}//end default constructor

//fire this event immediateley after the grid is loaded

private void Grid\_Loaded(object sender, RoutedEventArgs e)

{

//call the function that refreshes the grid data

refreshGrid();

//set the active row to row 1

CustomerDataGrid.SelectedIndex = 0;

}//end Grid\_Loaded

//function to refresh the data in the grid

//this is not only called here, but from the Add and Edit Record forms

//and from the Delete Record method

public void refreshGrid()

{

//create an instance of a data connection to the database customer table

CustomerDataDataContext con = new CustomerDataDataContext();

//create a customer list

List<Customer> customers = (from c in con.Customers select c).ToList();

CustomerDataGrid.ItemsSource = customers;

}//end refreshGrid

//menu item click event for Add Record

private void mnuitemAddRecord\_Click(object sender, RoutedEventArgs e)

{

AddRecord addRecordWindow = new AddRecord();

addRecordWindow.ShowDialog();

}//end mnuitemAddRecord\_Click

//menu item click event for Edit Record

private void mnuitemEditRecord\_Click(object sender, RoutedEventArgs e)

{

//local variables

string rowID;

//figure out which row in the datagrid is selected

object item = CustomerDataGrid.SelectedItem;

//try block to prevent an error found during coding. The delete method throws an error if no row

//is selected in the datagrid.

try

{

rowID = (CustomerDataGrid.SelectedCells[0].Column.GetCellContent(item) as TextBlock).Text;

}

catch

{

//no rows are highlighted in the grid

MessageBox.Show("Please select a row in the datagrid to edit and try again.");

return;

}

//check to make sure we received a valid rowID

if (rowID != null)

{

//show the Edit Record window

EditRecord editRecordWindow = new EditRecord(rowID);

editRecordWindow.ShowDialog();

}

}//end mnuitemEditRecord\_Click

//menu item click event for Delete Record

private void mnuitmDeleteRecord\_Click(object sender, RoutedEventArgs e)

{

//local variables

string rowID;

//create an instance of a data connection to the database customer table

CustomerDataDataContext con = new CustomerDataDataContext();

//figure out which row in the datagrid is selected

object item = CustomerDataGrid.SelectedItem;

//try block to prevent an error found during coding. The delete method throws an error if no row

//is selected in the datagrid.

try

{

rowID = (CustomerDataGrid.SelectedCells[0].Column.GetCellContent(item) as TextBlock).Text;

}

catch

{

//no rows are highlighted in the grid

MessageBox.Show("Please select a row in the datagrid to delete and try again.");

return;

}

//check to make sure we received a valid rowID

if (rowID != null)

{

//main try/catch block

try

{

//get the target row

var cust =

(from c in con.Customers

where c.CustomerID == rowID

select c).First();

//delete this record from the database

con.Customers.DeleteOnSubmit(cust);

con.SubmitChanges();

//refresh the datagrid

refreshGrid();

//tell the user this was successful

MessageBox.Show("The item '" + rowID + "' was deleted successfully.");

}

catch (Exception ex)

{

//tell the user this was successful

MessageBox.Show("The item '" + rowID + "' failed to delete. " + ex.ToString());

}

}

}//end mnuitmDeleteRecord\_Click

//menu item click event for Exit

private void mnuitmExit\_Click(object sender, RoutedEventArgs e)

{

//shutdown

Application.Current.Shutdown();

}//end mnuitmExit\_Click

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace DatabaseApplication

{

/// <summary>

/// Interaction logic for AddRecord.xaml

/// </summary>

public partial class AddRecord : Window

{

//default constructor

public AddRecord()

{

InitializeComponent();

}//end default constructor

//code to handle the Cancel button click event

private void Cancel\_Click(object sender, RoutedEventArgs e)

{

//close the add window

DialogResult = false;

}//end Cancel\_Click

//code to handle the Add button click event

private void btnAdd\_Click(object sender, RoutedEventArgs e)

{

//add the record to the database

//the ID must be unique and specified or do not proceed with the add

if (txtCustomerID.Text.Trim() == "")

{

MessageBox.Show("Missing unique Customer ID", "Entry error",

MessageBoxButton.OK, MessageBoxImage.Error);

return;

}

//add the record

Customer cust = new Customer();

cust.CustomerID = txtCustomerID.Text.Trim();

cust.CompanyName = txtCompanyName.Text.Trim();

cust.ContactName = txtContactName.Text.Trim();

cust.ContactTitle = txtContactTitle.Text.Trim();

cust.Country = txtCountry.Text.Trim();

cust.City = txtCity.Text.Trim();

cust.Region = txtRegionState.Text.Trim();

cust.PostalCode = txtPostalCode.Text.Trim();

cust.Address = txtAddress.Text.Trim();

cust.Phone = txtPhone.Text.Trim();

cust.Fax = txtFax.Text.Trim();

//main try block

try

{

//create an instance of a data connection to the database customer table

CustomerDataDataContext con = new CustomerDataDataContext();

//add new record to the database

con.Customers.InsertOnSubmit(cust);

con.SubmitChanges();

//tell the user this was added successfully

MessageBox.Show("Customer " + cust.CustomerID + " was added successfully.");

//refresh the data grid

Application app = Application.Current;

MainWindow main = (MainWindow)app.MainWindow;

main.refreshGrid();

//exit the dialog window with a success result

DialogResult = true;

}

catch (Exception ex)

{

//tell the user this was successful

MessageBox.Show("The item '" + cust.CustomerID + "' could not be added. " + ex.ToString());

//exit the dialog window without doing anything.

DialogResult = false;

}

}//end btnAdd\_Click

private void txtCustomerID\_Loaded(object sender, RoutedEventArgs e)

{

//put the cursor at the end of the customer ID textbox

txtCustomerID.Focus();

}//end txtCustomerID\_Loaded

}

}

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace DatabaseApplication

{

/// <summary>

/// Interaction logic for AddRecord.xaml

/// </summary>

public partial class EditRecord : Window

{

//class variables

protected string rowID; //stores the ID of the row being edited

//constructor to accept the Row ID from the grid on the main window

public EditRecord(string RowID)

{

InitializeComponent();

rowID = RowID;

}//end constructor with RowID

//code to handle the Cancel button click event

private void btnCancel\_Click(object sender, RoutedEventArgs e)

{

//close the edit window

DialogResult = false;

}//end Cancel\_Click

//code to handle the Save button click event

private void btnSave\_Click(object sender, RoutedEventArgs e)

{

//main try/catch block

try

{

//create an instance of a data connection to the database customer table

CustomerDataDataContext con = new CustomerDataDataContext();

// Query the database for the row to be updated.

var cust = con.Customers.SingleOrDefault(c => c.CustomerID == rowID);

//set all of the field values the record

cust.CustomerID = rowID;

cust.CompanyName = txtCompanyName.Text.Trim();

cust.ContactName = txtContactName.Text.Trim();

cust.ContactTitle = txtContactTitle.Text.Trim();

cust.Country = txtCountry.Text.Trim();

cust.City = txtCity.Text.Trim();

cust.Region = txtRegionState.Text.Trim();

cust.PostalCode = txtPostalCode.Text.Trim();

cust.Address = txtAddress.Text.Trim();

cust.Phone = txtPhone.Text.Trim();

cust.Fax = txtFax.Text.Trim();

//save record to the database

con.SubmitChanges();

//tell the user this was updated successfully

MessageBox.Show("Customer '" + rowID + "' was successfully updated.");

//refresh the data grid

Application app = Application.Current;

MainWindow main = (MainWindow)app.MainWindow;

main.refreshGrid();

//exit the dialog window with a success result

DialogResult = true;

}

catch (Exception ex)

{

//tell the user this was successful

MessageBox.Show("The item '" + rowID + "' could not be saved. " + ex.ToString());

//exit the dialog window without doing anything.

DialogResult = false;

}

}//end btnSave\_Click

//pull the data from the database and place the initial values in each control on this form

private void EditRecord\_Loaded(object sender, RoutedEventArgs e)

{

//create an instance of a data connection to the database customer table

CustomerDataDataContext con = new CustomerDataDataContext();

// Query the database for the row to be updated.

var cust = con.Customers.SingleOrDefault(c => c.CustomerID == rowID);

//the query should have returned 1 row

if(cust != null)

{

//set all of the field values the record

txtCustomerID.Text = cust.CustomerID;

txtCompanyName.Text = cust.CompanyName;

txtContactName.Text = cust.ContactName;

txtContactTitle.Text = cust.ContactTitle;

txtCountry.Text = cust.Country;

txtCity.Text = cust.City;

txtRegionState.Text = cust.Region;

txtPostalCode.Text = cust.PostalCode;

txtAddress.Text = cust.Address;

txtPhone.Text = cust.Phone;

txtFax.Text = cust.Fax;

//put the cursor at the end of the company name textbox

txtCompanyName.Focus();

txtCompanyName.SelectionStart = txtCompanyName.Text.Length;

}

}//end EditRecord\_Loaded

}

}

<Window x:Class="DatabaseApplication.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Homework 4: Courtney Bowden" Height="350" Width="525">

<Grid Loaded="Grid\_Loaded">

<Grid.ColumnDefinitions>

<ColumnDefinition Width="483\*"/>

<ColumnDefinition Width="34\*"/>

</Grid.ColumnDefinitions>

<Menu HorizontalAlignment="Stretch" VerticalAlignment="Top" Grid.ColumnSpan="2" HorizontalContentAlignment="Left" >

<MenuItem x:Name="mnuMenu" Header="Menu">

<MenuItem x:Name="mnuitmAddRecord" Header="Add Record" HorizontalAlignment="Left" Width="188" Margin="0,0,-43,0" Click="mnuitemAddRecord\_Click"/>

<MenuItem x:Name="mnuitmDeleteRecord" Header="Delete Record" HorizontalAlignment="Left" Width="181" Margin="0,0,-36,0" Click="mnuitmDeleteRecord\_Click"/>

<MenuItem x:Name="mnuitmExit" Header="Exit" HorizontalAlignment="Left" Width="145" Click="mnuitmExit\_Click"/>

</MenuItem>

<MenuItem x:Name="mnuEdit" Header="Edit">

<MenuItem x:Name="mnuitmEditRecord" Header="Edit Record" HorizontalAlignment="Left" Width="167" Margin="0,0,-27,0" Click="mnuitemEditRecord\_Click"/>

</MenuItem>

</Menu>

<DataGrid x:Name="CustomerDataGrid" HorizontalAlignment="Stretch" Margin="10,60,0,0" VerticalAlignment="Stretch"/>

<Label Content="Customers" HorizontalAlignment="Center" VerticalAlignment="Top" Margin="10,35,0,0"/>

</Grid>

</Window>

<Window x:Class="DatabaseApplication.AddRecord"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Add Record" Height="300" Width="635">

<Grid>

<Button x:Name="btnCancel" Content="Cancel" HorizontalAlignment="Left" Margin="439,235,0,0" VerticalAlignment="Top" Width="75" Click="Cancel\_Click" IsDefault="True"/>

<Button x:Name="btnAdd" Content="Add" HorizontalAlignment="Left" Margin="531,235,0,0" VerticalAlignment="Top" Width="75" Click="btnAdd\_Click"/>

<Label x:Name="lblCustomerID" Content="Customer ID:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="10,27,0,0"/>

<Label x:Name="lblCompanyName" Content="Company Name:" HorizontalAlignment="Left" Margin="10,58,0,0" VerticalAlignment="Top"/>

<Label x:Name="lblAddress" Content="Address:" HorizontalAlignment="Left" Margin="10,89,0,0" VerticalAlignment="Top"/>

<Label x:Name="lblCity" Content="City:" HorizontalAlignment="Left" Margin="10,120,0,0" VerticalAlignment="Top"/>

<Label x:Name="lblRegionState" Content="Region/State:" HorizontalAlignment="Left" Margin="10,151,0,0" VerticalAlignment="Top"/>

<Label x:Name="lblPostalCode" Content="Postal Code:" HorizontalAlignment="Left" Margin="10,182,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="txtCustomerID" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,31,0,0" MaxLength="5" TabIndex="0" Loaded="txtCustomerID\_Loaded"/>

<TextBox x:Name="txtCompanyName" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,61,0,0" MaxLength="40"/>

<TextBox x:Name="txtAddress" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,93,0,0" MaxLength="60"/>

<TextBox x:Name="txtCity" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,124,0,0" MaxLength="15"/>

<TextBox x:Name="txtRegionState" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,155,0,0" MaxLength="15"/>

<TextBox x:Name="txtPostalCode" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,183,0,0" MaxLength="10"/>

<Label x:Name="lblContactName" Content="Contact Name:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,28,0,0"/>

<Label x:Name="lblContactTitle" Content="Contact Title:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,61,0,0"/>

<Label x:Name="lblPhone" Content="Phone:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,97,0,0"/>

<Label x:Name="lblFax" Content="Fax:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,129,0,0"/>

<Label x:Name="lblCountry" Content="Country:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,164,0,0"/>

<TextBox x:Name="txtContactName" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,32,0,0" MaxLength="30"/>

<TextBox x:Name="txtContactTitle" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,64,0,0" MaxLength="30"/>

<TextBox x:Name="txtPhone" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,97,0,0" MaxLength="24"/>

<TextBox x:Name="txtFax" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,129,0,0" MaxLength="24"/>

<TextBox x:Name="txtCountry" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,157,0,0" MaxLength="15"/>

</Grid>

</Window>

<Window x:Class="DatabaseApplication.EditRecord"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="Edit Record" Height="300" Width="635">

<Grid Loaded="EditRecord\_Loaded">

<Button x:Name="btnCancel" Content="Cancel" HorizontalAlignment="Left" Margin="439,235,0,0" VerticalAlignment="Top" Width="75" Click="btnCancel\_Click" IsDefault="True"/>

<Button x:Name="btnSave" Content="Save" HorizontalAlignment="Left" Margin="531,235,0,0" VerticalAlignment="Top" Width="75" Click="btnSave\_Click"/>

<Label x:Name="lblCustomerID" Content="Customer ID:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="10,27,0,0"/>

<Label x:Name="lblCompanyName" Content="Company Name:" HorizontalAlignment="Left" Margin="10,58,0,0" VerticalAlignment="Top"/>

<Label x:Name="lblAddress" Content="Address:" HorizontalAlignment="Left" Margin="10,89,0,0" VerticalAlignment="Top"/>

<Label x:Name="lblCity" Content="City:" HorizontalAlignment="Left" Margin="10,120,0,0" VerticalAlignment="Top"/>

<Label x:Name="lblRegionState" Content="Region/State:" HorizontalAlignment="Left" Margin="10,151,0,0" VerticalAlignment="Top"/>

<Label x:Name="lblPostalCode" Content="Postal Code:" HorizontalAlignment="Left" Margin="10,182,0,0" VerticalAlignment="Top"/>

<TextBox x:Name="txtCustomerID" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,31,0,0" MaxLength="5" IsEnabled="False"/>

<TextBox x:Name="txtCompanyName" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,61,0,0" MaxLength="40"/>

<TextBox x:Name="txtAddress" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,93,0,0" MaxLength="60"/>

<TextBox x:Name="txtCity" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,124,0,0" MaxLength="15"/>

<TextBox x:Name="txtRegionState" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,155,0,0" MaxLength="15"/>

<TextBox x:Name="txtPostalCode" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="113,183,0,0" MinHeight="10"/>

<Label x:Name="lblContactName" Content="Contact Name:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,28,0,0"/>

<Label x:Name="lblContactTitle" Content="Contact Title:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,61,0,0"/>

<Label x:Name="lblPhone" Content="Phone:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,97,0,0"/>

<Label x:Name="lblFax" Content="Fax:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,129,0,0"/>

<Label x:Name="lblCountry" Content="Country:" HorizontalAlignment="Left" VerticalAlignment="Top" Margin="364,164,0,0"/>

<TextBox x:Name="txtContactName" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,32,0,0" MaxLength="30"/>

<TextBox x:Name="txtContactTitle" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,64,0,0" MaxLength="30"/>

<TextBox x:Name="txtPhone" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,97,0,0" MaxLength="24"/>

<TextBox x:Name="txtFax" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,129,0,0" MaxLength="24"/>

<TextBox x:Name="txtCountry" HorizontalAlignment="Left" Height="23" TextWrapping="Wrap" Text="" VerticalAlignment="Top" Width="120" Margin="486,157,0,0" MaxLength="15"/>

</Grid>

</Window>