﻿using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

using System.Data.OleDb;

using System.Drawing.Printing;

using PrintDataGrid;

namespace csvToGrid

{

public partial class search\_Received : Form

{

public search\_Received()

{

InitializeComponent();

this.dataGridView1.Columns[0].Visible = false;

}

//private void Form2\_Load(object sender, EventArgs e)

// {

// }

private void button1\_Click(object sender, EventArgs e)

{

OleDbConnection OdbConn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Search\KinneyDatabase.accdb");

DataTable dt = new DataTable();

dataGridView1.DataSource = dt;

}

private void button2\_Click(object sender, EventArgs e)

{

OleDbConnection OdbConn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Search\KinneyDatabase.accdb");

DataTable dt = new DataTable();

OleDbDataAdapter OdbA = new OleDbDataAdapter("Select \* from query\_All\_Received", OdbConn);

OdbA.Fill(dt);

dataGridView1.DataSource = dt;

}

private void button3\_Click(object sender, EventArgs e)

{

OleDbConnection OdbConn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Search\KinneyDatabase.accdb");

DataTable dt = new DataTable();

OleDbDataAdapter OdbA = new OleDbDataAdapter("SELECT TOP 9 \* from query\_All\_Received ORDER BY Date\_Received DESC", OdbConn);

OdbA.Fill(dt);

dataGridView1.DataSource = dt;

}

private void button4\_Click(object sender, EventArgs e)

{

Application.Exit();

}

private void button7\_Click(object sender, EventArgs e)

{

this.Close();

new Receive().Show();

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

try

{

OleDbConnection OdbConn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Search\KinneyDatabase.accdb");

DataTable dt = new DataTable();

dataGridView1.DataSource = dt;

}

catch (Exception c)

{

throw (c);

}

Console.WriteLine("Exception caught!");

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

try

{

OleDbConnection OdbConn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Search\KinneyDatabase.accdb");

DataTable dt = new DataTable();

dataGridView1.DataSource = dt;

}

catch (Exception c)

{

throw (c);

}

Console.WriteLine("Exception caught!");

}

private void button5\_Click(object sender, EventArgs e)

{

this.Hide();

new search\_Returned().Show();

}

private void button8\_Click(object sender, EventArgs e)

{

// Calling DataGridView Printing

PrintDGV.Print\_DataGridView(dataGridView1);

}

private void button6\_Click(object sender, EventArgs e)

{

this.Close();

new Return().Show();

}

private void pictureBox1\_Click(object sender, EventArgs e)

{

this.Hide();

new SplashScreen().Show();

}

private void button9\_Click(object sender, EventArgs e)

{

this.Close();

new search\_Unordered().Show();

}

private void button10\_Click(object sender, EventArgs e)

{

OleDbConnection OdbConn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Search\KinneyDatabase.accdb");

DataTable dt = new DataTable();

OleDbDataAdapter OdbA = new OleDbDataAdapter("Select \* from query\_Total\_All\_Received", OdbConn);

OdbA.Fill(dt);

dataGridView1.DataSource = dt;

}

private void button11\_Click(object sender, EventArgs e)

{

OleDbConnection OdbConn = new OleDbConnection(@"Provider=Microsoft.ACE.OLEDB.12.0;Data Source=C:\Search\KinneyDatabase.accdb");

DataTable dt = new DataTable();

OleDbDataAdapter OdbA = new OleDbDataAdapter("SELECT TOP 10 \* from query\_Total\_All\_Received ORDER BY Date\_Received DESC", OdbConn);

OdbA.Fill(dt);

dataGridView1.DataSource = dt;

}

private void search\_Received\_Load(object sender, EventArgs e)

{

// TODO: This line of code loads data into the 'kinneyDatabaseDataSet.script\_Orders' table. You can move, or remove it, as needed.

this.script\_OrdersTableAdapter.Fill(this.kinneyDatabaseDataSet.script\_Orders);

}

private void dataGridView1\_CellContentClick(object sender, DataGridViewCellEventArgs e)

{

}

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

{

}

private void bindingNavigator1\_RefreshItems(object sender, EventArgs e)

{

}

private void bindingNavigatorSeparator1\_Click(object sender, EventArgs e)

{

}

private void label2\_Click(object sender, EventArgs e)

{

}

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

{

}

private void label1\_Click(object sender, EventArgs e)

{

}

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

{

}

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

{

}

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

{

// TODO

try

{

string msg = "Do you want to permanently delete the selected records?";

string caption = "Delete Data";

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

MessageBoxIcon ico = MessageBoxIcon.Question;

DialogResult result;

result = MessageBox.Show(this, msg, caption, buttons, ico);

if (result == DialogResult.Yes)

{

// Delete

this.script\_OrdersBindingSource.RemoveCurrent();

// Saves the changes

this.script\_OrdersBindingSource.EndEdit();

this.script\_OrdersTableAdapter.Update(this.kinneyDatabaseDataSet.script\_Orders);

// Refresh data

this.script\_OrdersTableAdapter.Fill(this.kinneyDatabaseDataSet.script\_Orders);

MessageBox.Show("The record has been deleted.", "Delete Data", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

return;

}

}

catch (Exception ex)

{

MessageBox.Show("Delete data failed : " + ex.Message.ToString(), "Delete Data", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

}

private void dateTimePicker1\_ValueChanged(object sender, EventArgs e)

{

}

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

{

}

private void label3\_Click(object sender, EventArgs e)

{

}

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

{

}

private void bindingNavigatorSeparator2\_Click(object sender, EventArgs e)

{

}

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

{

this.script\_OrdersBindingSource.AddNew();

}

private void groupBox1\_Enter(object sender, EventArgs e)

{

}

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

{

Application.Exit();

}

private void dataGridView1\_CellContentClick\_1(object sender, DataGridViewCellEventArgs e)

{

}

private void cust\_NameLabel\_Click(object sender, EventArgs e)

{

}

private void cust\_NameTextBox\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

private void fill\_DateDateTimePicker\_ValueChanged(object sender, EventArgs e)

{

}

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

{

// TODO

try

{

string msg = "Do you want to permanently delete the selected records?";

string caption = "Delete Data";

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

MessageBoxIcon ico = MessageBoxIcon.Question;

DialogResult result;

result = MessageBox.Show(this, msg, caption, buttons, ico);

if (result == DialogResult.Yes)

{

// Delete

this.script\_OrdersBindingSource.RemoveCurrent();

// Saves the changes

this.script\_OrdersBindingSource.EndEdit();

this.script\_OrdersTableAdapter.Update(this.kinneyDatabaseDataSet.script\_Orders);

// Refresh data

this.script\_OrdersTableAdapter.Fill(this.kinneyDatabaseDataSet.script\_Orders);

MessageBox.Show("The record has been deleted.", "Delete Data", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

return;

}

}

catch (Exception ex)

{

MessageBox.Show("Delete data failed : " + ex.Message.ToString(), "Delete Data", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

// TODO

try

{

string msg = "Do you want to save | update the selected record?";

string caption = "Save | Update";

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

MessageBoxIcon ico = MessageBoxIcon.Question;

DialogResult result;

result = MessageBox.Show(this, msg, caption, buttons, ico);

if (result == DialogResult.Yes)

{

// Save | Update

this.script\_OrdersBindingSource.EndEdit();

this.script\_OrdersTableAdapter.Update(this.kinneyDatabaseDataSet.script\_Orders);

// Refresh data

this.script\_OrdersTableAdapter.Fill(this.kinneyDatabaseDataSet.script\_Orders);

MessageBox.Show("The record has been saved successfully.", "Save Data", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

return;

}

}

catch (Exception ex)

{

MessageBox.Show("Save | Update data failed : " + ex.Message.ToString(), "Save Data", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

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

{

this.script\_OrdersBindingSource.MoveNext();

}

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

{

this.script\_OrdersBindingSource.AddNew();

}

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

{

this.script\_OrdersBindingSource.MovePrevious();

}

private void search\_Received\_FormClosing(object sender, FormClosingEventArgs e)

{

string msg = "Do you want to exit the application?";

string caption = "Exit Application";

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

MessageBoxIcon ico = MessageBoxIcon.Question;

DialogResult result;

result = MessageBox.Show(this, msg, caption, buttons, ico);

if (result == DialogResult.Yes)

{

// Closes the form.

// MessageBox.Show("Goodbye, fellow user.", "",

// MessageBoxButtons.OK, MessageBoxIcon.Information);

e.Cancel = false;

}

else

{

e.Cancel = true;

}

}

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

{

// TODO

try

{

string msg = "Do you want to save | update the selected record?";

string caption = "Save | Update";

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

MessageBoxIcon ico = MessageBoxIcon.Question;

DialogResult result;

result = MessageBox.Show(this, msg, caption, buttons, ico);

if (result == DialogResult.Yes)

{

// Save | Update

this.script\_OrdersBindingSource.EndEdit();

this.script\_OrdersTableAdapter.Update(this.kinneyDatabaseDataSet.script\_Orders);

// Refresh data

this.script\_OrdersTableAdapter.Fill(this.kinneyDatabaseDataSet.script\_Orders);

MessageBox.Show("The record has been saved successfully.", "Save Data", MessageBoxButtons.OK, MessageBoxIcon.Information);

}

else

{

return;

}

}

catch (Exception ex)

{

MessageBox.Show("Save | Update data failed : " + ex.Message.ToString(), "Save Data", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

private void toolStripButtonRefresh\_Click\_1(object sender, EventArgs e)

{

search\_Received\_Load(sender, e);

}

private void scriptTextBox\_TextChanged(object sender, EventArgs e)

{

}

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

{

}

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

{

try

{

this.photoPictureBox.Image = null;

}

catch

{

}

}

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

{

try

{

OpenFileDialog ofd = new OpenFileDialog();

string myPictures = System.Environment.GetFolderPath(Environment.SpecialFolder.MyPictures);

ofd.Filter = "Jpg, Jpeg Images|\*.jpg;\*.jpeg|PNG Images|\*.png|BMP Images|\*.bmp|" + "All files (\*.\*)|\*.\*";

ofd.FileName = "Image file name";

ofd.Title = "Choose an image...";

ofd.AddExtension = true;

ofd.FilterIndex = 0;

ofd.Multiselect = false;

ofd.ValidateNames = true;

ofd.InitialDirectory = myPictures;

ofd.RestoreDirectory = true;

if (ofd.ShowDialog() == DialogResult.OK)

{

this.photoPictureBox.Image = Image.FromFile(ofd.FileName);

}

else

{

return;

}

}

catch

{

MessageBox.Show("Error : Something went wrong.", "Choose Image", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

}

}

}