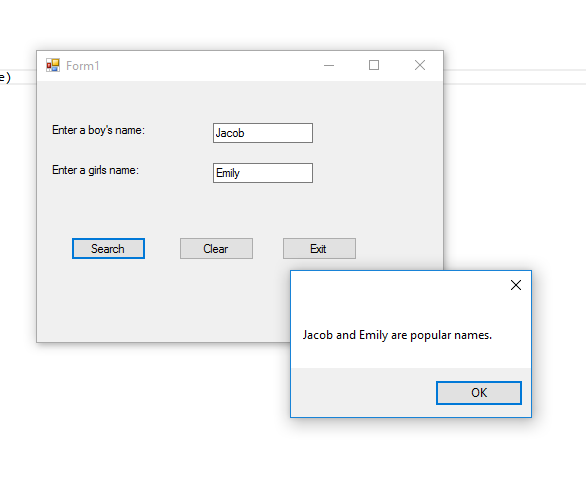
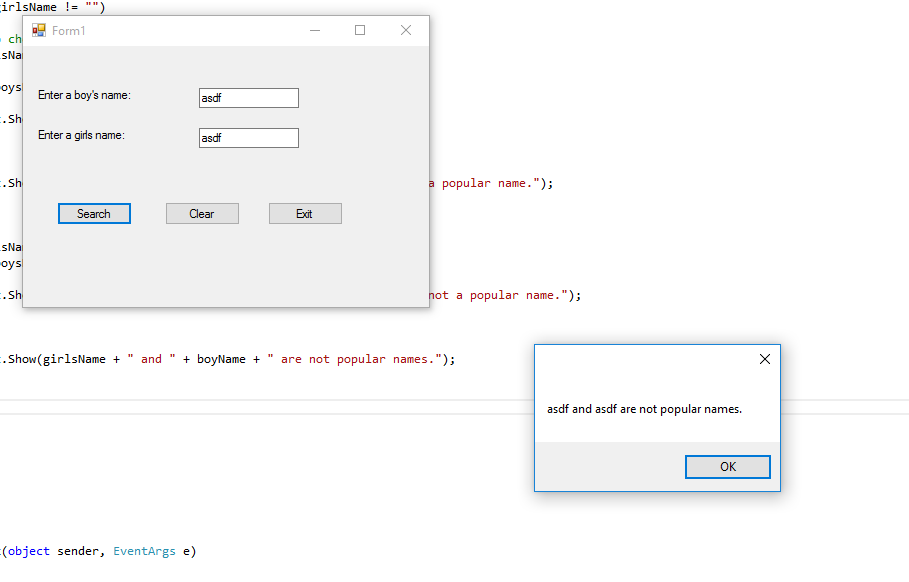
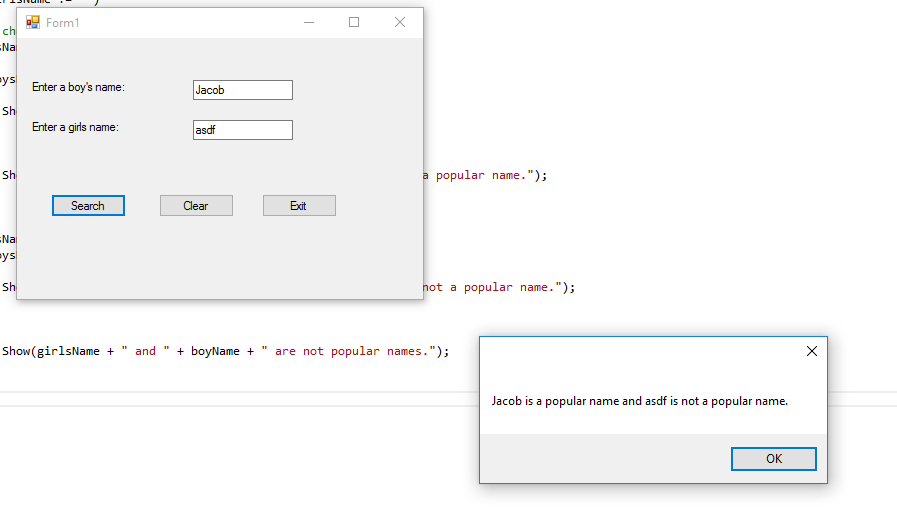
I worked alone







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.IO;

namespace NameSearch

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

//Add boysnames to BoysNameList

public void BoysInList(List<string> boysNameList)

{

try

{

//open boys llist

StreamReader inputFile=File.OpenText("BoyNames.txt");

//read names into list

while(!inputFile.EndOfStream)

{

boysNameList.Add(inputFile.ReadLine());

}

inputFile.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

//method to read girlsnames into a list

public void GirlsInList(List<string> girlsNameList)

{

try

{

//open boys llist

StreamReader inputFile2 = File.OpenText("GirlNames.txt");

//read names into list

while (!inputFile2.EndOfStream)

{

girlsNameList.Add(inputFile2.ReadLine());

}

inputFile2.Close();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

//check to see if boys name is in list

private bool BoyCheck(List<string> BoysNameList, string boyName )

{

//search for boy name

int position = BoysNameList.IndexOf(boyName);

//was boyname in list

if (position != -1)

{

return true;

}

else return false;

}

//check to see if girls name is in list

private bool GirlCheck(List<string> GirlsNameList, string girlName)

{

//search for girls name

int position = GirlsNameList.IndexOf(girlName);

//was girl in list

if (position != -1)

{

return true;

}

else return false;

}

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

{

//declaring local variables

string boyName; //holds boy's name

string girlsName; //holds girls name

boyName = boyTextBox.Text;

girlsName = girlTextBox.Text;

//create lists to hold names

List<string> boysNameList = new List<string>();

List<string> girlsNameList = new List<string>();

//read the boys name into the list

BoysInList(boysNameList);

//read girls name into the list

GirlsInList(girlsNameList);

//determine messages to display

//if no name is entered

if (boyName == "" && girlsName == "")

{

MessageBox.Show("Please enter at least one name");

}

//Checks if boy name textbox is not populate and girls is

if (boyName != "" && girlsName == "")

{

//calls boycheck method and determines if boy is in the list

if (BoyCheck(boysNameList, boyName) == true)

{

MessageBox.Show(boyName + " is a popular name.");

}

else

{

MessageBox.Show(boyName + " is not a popular name.");

}

}

//Checks if boy name textbox is not populate and girls is

if (boyName == "" && girlsName != "")

{

//calls GirlCheck method and determines if the girls name is in the list

if (GirlCheck(girlsNameList, girlsName) == true)

{

MessageBox.Show(girlsName + " is a popular name.");

}

else

{

MessageBox.Show(girlsName + " is not a popular name.");

}

}

//Determines if there is both a boy and a girls name populated

if (boyName != "" && girlsName != "")

{

//calls methods to check if names are in the list

if (GirlCheck(girlsNameList, girlsName) == true)

{

if (BoyCheck(boysNameList, boyName) == true)

{

MessageBox.Show(boyName + " and " + girlsName + " are popular names.");

}

else

{

MessageBox.Show(girlsName + "is a popular name, " + boyName + " is not a popular name.");

}

}

if (GirlCheck(girlsNameList, girlsName) == false)

if (BoyCheck(boysNameList, boyName) == true)

{

MessageBox.Show(boyName + " is a popular name and " + girlsName + " is not a popular name.");

}

else

{

MessageBox.Show(girlsName + " and " + boyName + " are not popular names.");

}

}

}

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

{

//Clear Text Boxes and give focus

boyTextBox.Clear();

girlTextBox.Clear();

boyTextBox.Focus();

}

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

{

//closes the form

this.Close();

}

}

}