**Partial acceptance test**

This acceptance test should be run on the completed application in Debug mode to allow you to check the expected results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Test** | **Expected action** | **Actual action** | **Passed**  **Y/N** |
| 1 | Click on Clear all Entries. | Should prompt for confirmation. | As Expected | y |
| 2 | Confirm deletion. | Should clear all entries in collection. | As expected | y |
| 3 | Click on List All. | No entries should appear. | As expected | y |
| 4 | Add new entry: extension txt and name Notepad. | Should prompt ‘Entry added’ and display new entry ‘File extension .txt opens with Notepad’. | As expected | y |
| 5 | Add new entry: extension doc and name  Winword. | Should prompt ‘Entry added’ and display new entry ‘File extension .doc opens with Winword’. | As expected | y |
| 6 | Add new entry:  extension html and name  Firefox. | Should prompt ‘Entry added’ and display new entry ‘File extension .html opens with Firefox’. | As expected | y |
| 7 | Add new entry:extension txt and name  Wordpad. | Should display error message such as ‘entry with key = txt already exists’ and clear text box showing display. | As expected | y |
| 8 | Click on ‘List All’. | Should display three entries similar to | As expected | y |
| 9 | Enter extension txt and then click on find default program button. | Should display something similar to Extension .txt opens with Notepad. | As expected | y |
| 10 | Click on delete current entry. | Should display something similar to entry for txt removed. | As expected | y |
| 11 | Click on ‘List All’. | Should display two entries similar to | As expected | y |
| 12 | Enter extension xls and  then click on find default program button. | Should display something like ‘no entry with key xls exists’. | As Expected | y |
| 13 | Ensure xls still in find  default program text box and then click on delete current entry. | Should display something like ‘no entry with key xls to remove’. | As Expected | y |
| 14 | Exit program. | Program should exit with data stored in file datafile.dat | As expected | y |
| 15 | Start program and click on  List all. | Should display two entries similar to | As Expected | y |

Name of tester: Mark Riley

Date of test:

14/02/18 Test passed/failed: 15

List any errors found and how they were fixed.

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.IO;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Data\_Structures\_Assessment3\_Ex4

{

public partial class frmFileExtensions : Form

{

//Dictionary for storing entries of <KEY, VALUE>

private Dictionary<string, string> dict = new Dictionary<string, string>();

public frmFileExtensions()

{

InitializeComponent();

ReadDataFile();

}

/// <summary>

/// helper method for writing data to data file

/// </summary>

private void UpdateDataFile()

{

//should write collection to file datafile.dat

try

{

using (StreamWriter file = new StreamWriter("datafile.dat"))

foreach (var entry in dict)

file.WriteLine("{0} : {1}", entry.Key, entry.Value);

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

/// <summary>

/// helper method for reading data from data file

/// </summary>

private void ReadDataFile()

{

//should read in collection from file datafile.dat if it exists

try

{

using (StreamReader sr = new StreamReader("datafile.dat"))

{

while (!sr.EndOfStream) //Keep reading until we get to the end

{

string splitMe = sr.ReadLine();

string[] dataSplits = splitMe.Split(':'); //Split at the colons

dict.Add(dataSplits[0].Trim(), dataSplits[1].Trim());

}

}

string displayText = String.Empty;

foreach (var pair in dict)

{

displayText = displayText + pair.Key + " : " +

pair.Value + System.Environment.NewLine;

}

txtDisplay.Text = displayText;

}

catch (Exception ex) { MessageBox.Show(ex.Message); }

}

/// <summary>

/// Event handler method called when for load

/// should read entries from data file (if it exists)

/// or create a new collection object (the instance variable)

/// </summary>

/// <summary>

/// event handler method for clicking on Find Program button

/// </summary>

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

{

// should update textDisplay appropriately if found

string temp = string.Empty;

foreach(var item in dict)

{

if(item.Key == textBoxFindProgram.Text)

{

temp = item.Value;

}

if (temp == string.Empty)

{

txtDisplay.Text = "Extension not associated with a program";

}

else txtDisplay.Text = temp;

}

}

/// <summary>

/// Event handler method for listing all of the entries in the phoneBook

/// </summary>

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

{

//should iterate through collection object

//and append text to textDisplay text box

txtDisplay.Text = "";

string displayText = string.Empty;

foreach (var item in dict)

{

displayText = displayText + item.Value + System.Environment.NewLine;

}

txtDisplay.Text = displayText;

}

//Creates a new entry of Key, Value which is added/saved to the file

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

{

try

{

dict.Add(this.txtExtension.Text, this.txtProgram.Text);

string displayText = String.Empty;

foreach (var pair in dict)

{

displayText = displayText + pair.Key + " : " + pair.Value + System.Environment.NewLine;

}

txtDisplay.Text = displayText;

UpdateDataFile();

MessageBox.Show("File Added");

}

catch (Exception ex)

{ MessageBox.Show(ex.Message); }

}

//Event handler method for deleting an existing entry

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

{

if (txtDisplay.Text == string.Empty)

{

MessageBox.Show("There's nothing to delete");

}

else

{

try

{

dict.Remove(this.txtExtension.Text);

string displayText = String.Empty;

foreach (var pair in dict)

{

displayText = displayText + pair.Key + " : " +

pair.Value + System.Environment.NewLine;

}

txtDisplay.Text = displayText;

txtExtension.Text = "";

txtProgram.Text = "";

UpdateDataFile();

MessageBox.Show("File Deleted");

}

catch (Exception ex)

{ MessageBox.Show(ex.Message); }

}

}

// If confirmed will clear all values in collection

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

{

if (MessageBox.Show("Do you really want to delete all entries ?", "Confirm delete", MessageBoxButtons.YesNo) == DialogResult.Yes)

{

//clears all entries in collection

dict.Clear();

txtDisplay.Text = "";

txtExtension.Text = "";

txtProgram.Text = "";

UpdateDataFile();

MessageBox.Show("Cleared");

}

//Clears text fields

txtExtension.Text = "";

txtProgram.Text = "";

}

}

}