 Active Systems PTY

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;

// Name: Poorav Sharma

// This GUI application will demonstrate the use of a List<string> to collect and process the vehicle registration plate information.

namespace Assessment\_3

{

public partial class Vehicle\_Registration\_Management : Form

{

public Vehicle\_Registration\_Management()

{

InitializeComponent();

}

List<string> PlateList = new List<string>() { };//Question Two: The prototype must use a List<> data structure of data type “string”.

string currentFileName = "demo\_00.txt";

int plateNumber = 0;

#region Buttons

//Question Three: OPEN: When the OPEN button is clicked the user can select different data from pre-saved text files.

private void button\_Open\_Click(object sender, EventArgs e)

{

string fileName = "demo\_01.txt";

OpenFileDialog OpenText = new OpenFileDialog();

DialogResult sr = OpenText.ShowDialog();

OpenText.Filter = "Text Files | \*.txt";

OpenText.DefaultExt = "txt";

if (sr == DialogResult.OK)

{

fileName = OpenText.FileName;

}

currentFileName = fileName;

try

{

PlateList.Clear();

using (StreamReader reader = new StreamReader(File.OpenRead(fileName)))

{

while (!reader.EndOfStream)

{

PlateList.Add(reader.ReadLine());

}

}

DisplayList();

}

catch (IOException)

{

MessageBox.Show("File NOT openned");

}

}

//Question Four: ENTER: To add a rego plate to the List<> the user will type the data value (rego plate info) into the TextBox and click the ENTER button.

private void button\_Enter\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrWhiteSpace(textBox.Text) && (ValidName(textBox.Text)))

{

PlateList.Add(textBox.Text);

DisplayList();

textBox.Clear();

plateNumber++;

StripStatus.Text = "Plate added to the list";

}

else if (string.IsNullOrWhiteSpace(textBox.Text))

{

StripStatus.Text = "Error: TextBox is empty";

}

else

{

StripStatus.Text = "Error: Plate is already present in list";

}

textBox.Focus();

}

//Question Five LEAVE: Method Two: the user will enter the rego plate information into the TextBox and click the DELETE button.

private void button\_Leave\_Click(object sender, EventArgs e)

{

if (listBox.SelectedIndex != -1)

{

listBox.SetSelected(listBox.SelectedIndex, true);

PlateList.RemoveAt(listBox.SelectedIndex);

plateNumber--;

textBox.Clear();

DisplayList();

StripStatus.Text = "Plate removed from list";

}

else

{

StripStatus.Text = "Error: Plate is not selected";

}

textBox.Focus();

}

//Question Five LEAVE: Method One: double click a data item from the ListBox and click the OK button in the popup dialog box.The data item will be removed from the List<>.

private void listBox\_MouseDoubleClick(object sender, MouseEventArgs e)

{

DialogResult dialog = MessageBox.Show("Do you want to delete the selected plate?", "", MessageBoxButtons.YesNo, MessageBoxIcon.Warning);

if (dialog == DialogResult.Yes)

{

PlateList.RemoveAt(listBox.SelectedIndex);

plateNumber--;

textBox.Clear();

DisplayList();

textBox.Focus();

StripStatus.Text = "Plate has been removed";

}

}

//Question Six EDIT: To edit a rego plate click(select) a data item from the ListBox so that is appears in the TextBox.

private void button\_Edit\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(textBox.Text))

{

if (listBox.SelectedItem != null)

{

string currentItem = listBox.SelectedItem.ToString();

int taskIndex = listBox.FindString(currentItem);

PlateList[taskIndex] = textBox.Text;

StripStatus.Text = "Plate edited";

}

}

else

{

StripStatus.Text = "Error: Select a plate from the list";

}

DisplayList();

textBox.Clear();

textBox.Focus();

}

// Question Eleven LINEAR SEARCH: Add a second search button that implements a linear search algorithm.

private void button\_LinearSearch\_Click(object sender, EventArgs e)

{

string target = textBox.Text;

bool found = false;

for (int i = 0; i < plateNumber; i++)

{

if (string.Compare(target, PlateList[i]) == 0)

{

found = true;

break;

}

}

if (found == true)

{

MessageBox.Show("The Plate " + target + " is found", "Sucess", MessageBoxButtons.OK, MessageBoxIcon.Information);

StripStatus.Text = "Sucess: Plate " + target + " is found";

}

else

{

MessageBox.Show("Plate not found", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

StripStatus.Text = "Error: Plate not found";

}

textBox.Clear();

textBox.Focus();

}

//Question Ten BINARY SEACH: To find a specific rego plate the user will type the information into the TextBox and click the BINARY SEARCH button.

private void button\_BinarySearch\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(textBox.Text))

{

if (PlateList.BinarySearch(textBox.Text) >= 0)

{

MessageBox.Show("The plate is Found", "Success", MessageBoxButtons.OK, MessageBoxIcon.Information);

StripStatus.Text = "Success: Plate is found";

}

else

{

MessageBox.Show("The plate is Not Found", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

StripStatus.Text = "Error: Plate not found";

}

}

else

{

MessageBox.Show("TextBox is empty", "Error", MessageBoxButtons.OK, MessageBoxIcon.Error);

StripStatus.Text = "Error: TextBox is empty";

}

textBox.Clear();

textBox.Focus();

}

//Question Thirteen TAG: When a rego plate is selected from the ListBox and “tagged” an additional character value “z” will be prefixed to the rego plate.The List<> will be re-sorted and displayed.

private void button\_Tag\_Click(object sender, EventArgs e)

{

if (!string.IsNullOrEmpty(textBox.Text))

{

PlateList[listBox.SelectedIndex] = "z" + PlateList[listBox.SelectedIndex];

StripStatus.Text = "Plate has been tagged";

}

else

{

StripStatus.Text = "Error: Plate is not selected";

}

DisplayList();

textBox.Clear();

}

//Question Seven: RESET: Add a RESET button to clear all the data items from the List<>.

private void button\_Reset\_Click(object sender, EventArgs e)

{

textBox.Clear();

PlateList.Clear();

DisplayList();

StripStatus.Text = "Plate list has been reset";

}

//Question Twelve SAVE: Add a SAVE button that can utilise the save method.

private void button\_Save\_Click(object sender, EventArgs e)

{

string fileName = "demo\_01.txt";

SaveFileDialog SaveText = new SaveFileDialog();

DialogResult sr = SaveText.ShowDialog();

SaveText.Filter = "Text Files | \*.txt";

SaveText.DefaultExt = "txt";

if (sr == DialogResult.OK)

{

fileName = SaveText.FileName;

}

if (sr == DialogResult.Cancel)

{

SaveText.FileName = fileName;

}

try

{

using (StreamWriter writer = new StreamWriter(fileName, false))

{

foreach (var plate in PlateList)

{

writer.WriteLine(plate);

}

}

}

catch (IOException)

{

MessageBox.Show("File NOT saved");

}

}

#endregion

#region Utility

private void DisplayList()

{

listBox.Items.Clear();

PlateList.Sort();

foreach (var plate in PlateList)

{

listBox.Items.Add(plate);

}

}

private bool ValidName(string checkThisName)

{

if (PlateList.Exists(duplicate => duplicate.Equals(checkThisName)))

return false;

else

return true;

}

//Question Eight SINGLE DATA DISPLAY: Create a single click method to do the following: when a data item is selected from the ListBox, the information is displayed in the TextBox .

private void listBox\_SelectedIndexChanged(object sender, EventArgs e)

{

if (listBox.SelectedIndex != -1)

{

string dataItem = listBox.SelectedItem.ToString();

int dataItemIndex = listBox.FindString(dataItem);

textBox.Text = PlateList[dataItemIndex].ToString();

textBox.Focus();

}

else

{

MessageBox.Show("Please select from the List Box", "ERROR", MessageBoxButtons.OK, MessageBoxIcon.Error);

}

PlateList.Sort();

}

#endregion

//Question Nine DISPLAY and SORT: All the rego plates should be displayed in the ListBox which is sorted alphabetically using the built-in List Sort method.

private void Vehicle\_Registration\_Management\_Load(object sender, EventArgs e)

{

DisplayList();

PlateList.Sort();

}

//Ouestion Twelve Close: Create a FORM closing method using the save method so all data from the List<> will be written back to a single text file called “demo\_##.txt” file which is auto incremented

private void Vehicle\_Registration\_Management\_FormClosed(object sender, FormClosedEventArgs e)

{

currentFileName = Path.GetFileNameWithoutExtension(currentFileName);

string str = currentFileName.Remove(0, 5);

int num;

try

{

num = int.Parse(str);

num++;

string fileClose;

if (num <= 9)

{

fileClose = "demo\_0" + num + ".txt";

}

else

{

fileClose = "demo\_" + num + ".txt";

}

using (StreamWriter writer = new StreamWriter(fileClose, false))

{

foreach (var plate in PlateList)

{

writer.WriteLine(plate);

}

}

}

catch (Exception)

{

return;

}

}

}

}