using System;

using System.Collections.Generic;

using System.IO;

using System.Linq;

using System.Security.AccessControl;

using System.Windows.Forms;

using System.Xml.Linq;

namespace RegoReader\_v1.\_0

{

public partial class RegistrationReaderForm : Form

{

public RegistrationReaderForm()

{

InitializeComponent();

}

//

// global variables.

//

public bool enableDoubleClick = true;

public List<String> regoList = new List<String>();

public string fileDirectory = "D:\\";

//

// Update List Method. Clears list box then displays all the items // in regoList<>

//

private void UpdateList()

{

ListBoxRego.Items.Clear();

regoList.Sort();

foreach (String s in regoList)

{

ListBoxRego.Items.Add(s);

}

}

//

// Method clears list box selection and re-enables add button

// function

//

private void ClearSelect()

{

if (ListBoxRego.SelectedItem != null)

{

TextBoxEntry.Clear();

}

ListBoxRego.ClearSelected();

enableDoubleClick = true;

ButtonAdd.Enabled = true;

ButtonDelete.Enabled = true;

}

//

// Initialise Entry Method. Replaces spaces with "-" and

// capitalises letters.

//

private string InitialiseEntry(string s)

{

s = s.Replace(" ", "-");

s = s.ToUpper();

return s;

}

//

// Open button click causes an open file dialog to open.

// Filters for txt files

//

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

{

OpenFileDialog openFileDialog1 = new OpenFileDialog

{

InitialDirectory = fileDirectory,

Title = "Browse Text Files",

CheckFileExists = true,

CheckPathExists = true,

DefaultExt = "txt",

Filter = "txt files (\*.txt)|\*.txt",

FilterIndex = 1,

RestoreDirectory = true,

ReadOnlyChecked = true,

ShowReadOnly = true

};

if (openFileDialog1.ShowDialog() == DialogResult.OK)

{

string filename = openFileDialog1.FileName;

foreach (string s in File.ReadAllLines(filename).ToList())

{

if (!regoList.Contains(s))

{

regoList.Add(InitialiseEntry(s));

}

else

{

continue;

}

}

// Initialises default file directory to the one where file // opened from.

fileDirectory = System.IO.Path.GetDirectoryName(filename);

UpdateList();

TextBoxErrorMes.Text = "File Opened";

}

}

//

// Add Button:

// Will add entered text into listbox - auto capitalised.

// Check for valid entry - if not, will display an error message.

//

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

{

if (!String.IsNullOrWhiteSpace(TextBoxEntry.Text))

{

string entry = TextBoxEntry.Text;

entry = InitialiseEntry(entry);

if (!regoList.Contains(entry))

{

regoList.Add(InitialiseEntry(TextBoxEntry.Text));

UpdateList();

TextBoxEntry.Clear();

TextBoxErrorMes.Text = "Registration Plate Added";

}

}

else

{

TextBoxErrorMes.Text = "Invalid Entry. Plate already exists

or there was no data entered.";

}

}

//

//Delete item with double click

//

private void ListBoxRego\_DoubleClick(object sender, EventArgs e)

{

if(ListBoxRego.SelectedItem != null)

{

var confirm = MessageBox.Show("Are you sure you want to

delete this item?", "Delete

Confirmation",

MessageBoxButtons.YesNo);

if (confirm == DialogResult.Yes)

{

regoList.Remove(ListBoxRego.SelectedItem.ToString());

ClearSelect();

UpdateList();

TextBoxErrorMes.Text = "Registration plate deleted";

}

else

{

ClearSelect();

}

}

}

//

// Delete Item with button

//

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

{

if (ListBoxRego.SelectedItem != null)

{

var confirm = MessageBox.Show("Are you sure you want to

delete this item?", "Delete

Confirmation",

MessageBoxButtons.YesNo);

if (confirm == DialogResult.Yes)

{

regoList.Remove(ListBoxRego.SelectedItem.ToString());

ClearSelect();

UpdateList();

TextBoxErrorMes.Text = "Registration plate deleted";

}

else

{

ListBoxRego.ClearSelected();

}

}

}

//

// Reset button. Clears ListBox.

//

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

{

var confirm = MessageBox.Show("Are you sure you want to reset

the list?\n\nAll current

information will be lost.",

"Reset Confirmation",

MessageBoxButtons.YesNo);

if (confirm == DialogResult.Yes)

{

regoList.Clear();

UpdateList();

TextBoxErrorMes.Text = "List Reset";

}

else

{

ClearSelect();

}

}

//

// Clicking a rego number in the

//

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

{

if (ListBoxRego.SelectedItem != null)

// Checks to see if item is clicked or not.

{

string selection = (string)ListBoxRego.SelectedItem;

if (selection.StartsWith("z"))

{

selection = selection.Substring(1);

TextBoxEntry.Text = selection;

}

else if (ListBoxRego.SelectedItem != null)

{

TextBoxEntry.Text = selection;

}

else

{

ClearSelect();

}

ButtonAdd.Enabled = false;

}

}

//

// Edit button. Modifies current selected item with text from text // box when clicked.

//

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

{

if (ListBoxRego.SelectedItem != null)

{

// lambda function I don't know how works. But uses the

// selected item as the input

// for the FindIndex list function which is stored in index // variable. Then uses that

// index to replace the string at that index with the // TextBoxEntry text.

var index = regoList.FindIndex(c =>

c.Equals(ListBoxRego.SelectedItem.ToString()));

// From: <https://stackoverflow.com/questions/4914802/editing-an-item-in-a-listt#:~:text=How%20do%20I%20edit%20an,this%20index%20count%2B%2B%3B%20%7D%20list>.

if ((regoList[index]).StartsWith("z"))

{

regoList[index] =

"z" + InitialiseEntry(TextBoxEntry.Text);

}

else

{

regoList[index] = InitialiseEntry(TextBoxEntry.Text);

}

TextBoxEntry.Clear();

UpdateList();

ClearSelect();

TextBoxErrorMes.Text = "Plate modified";

}

else

{

TextBoxErrorMes.Text = "No selection made";

}

}

//

// Clicking outside of window elements clears list box selection.

//

private void splitContainer1\_Panel2\_Click(object sender, EventArgs e)

{

ClearSelect();

}

//

// Calls linear search to search rego plates currently stored.

//

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

{

regoList.Sort(); // Ensuring the list is sorted before search

if (!string.IsNullOrWhiteSpace(TextBoxEntry.Text))

{

LinearSearch(regoList,

InitialiseEntry(TextBoxEntry.Text));

ClearSelect();

TextBoxErrorMes.Text = "Linear search complete";

}

else

{

TextBoxErrorMes.Text = "Enter valid search entry";

}

}

//

// Linear Search Method.

//

private void LinearSearch(List<string> list, string key)

{

for (int i = 0; i < list.Count; i++)

{

if ((list[i] == key) || (list[i] == ("z"+key)))

{

MessageBox.Show("Registration plate exists.","Search

Result");

return;

}

}

MessageBox.Show("Registraition plate does not exist.", "Search Result");

return;

}

//

// Binary Search Button. Sorts list then uses built in binary

// search to find rego plate.

//

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

{

regoList.Sort(); // Ensuring the list is sorted before search.

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

{

if

(!(regoList.BinarySearch(InitialiseEntry(TextBoxEntry.Text)) < 0) ||

!(regoList.BinarySearch("z" + InitialiseEntry(TextBoxEntry.Text))<0))

{

MessageBox.Show("Registration plate exists.", "Search

Result");

}

else

{

MessageBox.Show("Registration plate does not exist.",

"Search Result");

}

TextBoxErrorMes.Text = "Binary search complete";

}

else

{

TextBoxErrorMes.Text = "Enter valid search entry";

}

}

//

// Save Button. Calls save file Method.

//

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

{

SaveFile();

}

//

// Save File Method. Creates save file dialogue, and automatically // increments a suggested name

//

private void SaveFile()

{

SaveFileDialog save = new SaveFileDialog();

save.InitialDirectory = fileDirectory;

save.Filter = "Text Files (\*.txt)|\*.txt";

save.DefaultExt = ".txt";

save.Title = "Select Save Location";

save.RestoreDirectory = true;

// Auto incrementation of file name. Will loop for as long as // the file name already exists in target folder.

int num = 0;

string filename;

string extension = ".txt";

string name;

// do-while from: https://stackoverflow.com/questions/10131667/automatically-increment-filename

do

{

name = "demo\_";

if (num < 9)

{

name = "demo\_0";

num++;

filename = String.Format("{0}{1}", name, num);

}

else

{

num++;

filename = String.Format("{0}{1}", name, num);

}

} while (File.Exists(fileDirectory+"\\"+name+num+extension));

save.FileName = filename;

// Sets new default directory to save location.

fileDirectory =

System.IO.Path.GetDirectoryName(save.FileName);

// Checks if file name is valid and "save" is pressed in // savedialog.

DialogResult result = save.ShowDialog();

if ((save.FileName != "") && (result == DialogResult.OK))

{

// StreamWriter will create the auto incremented named file // in the target folder for it to write to.

System.IO.StreamWriter sw = new

StreamWriter("E:\\DemoRego\\" + filename+extension);

// Will loop for each list element, writes each into a txt // file.

foreach (var item in regoList)

{

sw.WriteLine(item.ToString());

}

TextBoxErrorMes.Text = String.Format("File saved as: {0}{1} {2}", name, num,

extension);

sw.Close();

// Write to file From: https://stackoverflow.com/questions/20595279/c-sharp-save-all-items-in-a-listbox-to-text-file

}

}

//

// Exit Button. Confirms exit of program and Calls the save file

// method before closing.

//

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

{

var result = MessageBox.Show("Are you sure you want to exit?

\n(Program will propt a save

before closing)", "Exit Confirmation", MessageBoxButtons.YesNo);

if (result == DialogResult.Yes)

{

SaveFile();

Close();

}

}

//

// Tag Button. Will add a "z" to the front of a rego plate and

// remove it if it starts with a z

//

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

{

if (ListBoxRego.SelectedIndex != -1)

{

var index = regoList.FindIndex(c =>

c.Equals(ListBoxRego.SelectedItem.ToString())); // Finds the index of selected item.

string selection = ListBoxRego.SelectedItem.ToString();

// Stores value of selection.

if (!selection.StartsWith("z"))

{

regoList[index] = "z" + selection;

ClearSelect();

UpdateList();

TextBoxErrorMes.Text = "Registration entry: " +

selection + " tagged";

}

// From:https://www.tutorialspoint.com/How-to-find-the-first-character-of-a-string-in-Chash#:~:text=How%20to%20find%20the%20first%20character%20of%20a%20string%20in%20C%23%3F&text=To%20get%20the%20first%20character%2C%20use%20the%20substring()%20method.&text=string%20str%20%3D%20%22Welcome%20to%20the,in%20the%20substring()%20method.

else

{

TextBoxErrorMes.Text = "Registration entry: " +

selection + " untagged";

// Else statement begins with this statement so it keeps the "z" when printing message.

regoList[index] = selection.Substring(1);

ClearSelect();

UpdateList();

}

//From:https://stackoverflow.com/questions/3222125/fastest-way-to-remove-first-char-in-a-string

}

}

//

// All Tooltip hovers. Give brief descripton of each element

// function.

// Code based off:https://www.youtube.com/watch?v=izv5S38ctBs

//

private void TextBoxEntry\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Enter Registraion Plate Information Here", TextBoxEntry);

}

private void ButtonAdd\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Adds text in entry field to list", ButtonAdd);

}

private void ButtonEdit\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Replaces selection with text in entry field", ButtonEdit);

}

private void ButtonDelete\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Deletes selection", ButtonDelete);

}

private void ButtonOpen\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Browse text files and import contents", ButtonOpen);

}

private void ButtonSave\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Saves content of list to text file", ButtonSave);

}

private void ButtonExit\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Save and Exit Program", ButtonExit);

}

private void ListBoxRego\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("List of Registration Plates\n\nDouble-Click Item to Delete", ListBoxRego);

}

private void ButtonTag\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Add a \"z\" to the front of selected plate number\n\nPress \"Tag\" again to remove.", ButtonTag);

}

private void ButtonReset\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Clears the list\n\nList will NOT save", ButtonReset);

}

private void ButtonLinSearch\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Check if plate exists in list\n\nsearches using linear search", ButtonLinSearch);

}

private void ButtonBinSearch\_MouseHover(object sender, EventArgs e)

{

toolTip1.Show("Check if plate exists in list\n\nsearches using binary search", ButtonBinSearch);

}

}

}