using System**;**

using System**.**Collections**.**Generic**;**

using System**.**ComponentModel**;**

using System**.**Data**;**

using System**.**Drawing**;**

using System**.**IO**;**

using System**.**Linq**;**

using System**.**Runtime**.**Serialization**.**Formatters**.**Binary**;**

using System**.**Text**;**

using System**.**Threading**.**Tasks**;**

using System**.**Windows**.**Forms**;**

***// CR = Client Requirement***

***// PR = Program Requirement***

***// Jervin Alejandro***

***// Date 09 November 2021***

***// Version 1.0***

***// Vehicle Registration***

***// Program allows user to add, delete, edit, tag, reset registration plates. Additionally, the program can save the data to a text file***

***// Furthermore, the program can also open a text file that contains regstration plates***

***// Registration plates are displayed on the listbox***

namespace Lists

**{**

**public** partial **class** Lists **:** Form

**{**

**public** Lists**()**

**{**

InitializeComponent**();**

**}**

***// PR: The prototype must use a List<> data structure of data type “string”.***

List**<**string**>** RegoPlate **=** **new** List**<**string**>();**

***// Temp list to check if Registration Plates are already tagged***

List**<**string**>** isTagged **=** **new** List**<**string**>();**

#region FUNCTIONS

***// PR: Display all added/removed/edited items from the list***

***// PR: List is sorted aplhabetically using List sort method***

**private** **void** DisplayList**()**

**{**

listDisplay**.**Items**.**Clear**();**

RegoPlate**.**Sort**();**

foreach **(**var rego in RegoPlate**)**

**{**

listDisplay**.**Items**.**Add**(**rego**);**

**}**

**}**

***// Load demo\_##.txt file by default***

**private** **void** Lists\_Load**(**object sender**,** EventArgs e**)**

**{**

string fileName **=** "demo\_##.txt"**;**

using **(**Stream openFile **=** File**.**Open**(**fileName**,** FileMode**.**Open**))**

**{**

BinaryFormatter binFormat **=** **new** BinaryFormatter**();**

**while** **(**openFile**.**Position **<** openFile**.**Length**)**

**{**

RegoPlate**.**Add**((**string**)**binFormat**.**Deserialize**(**openFile**));**

**}**

**}**

DisplayList**();**

**}**

***// PR: Selected data is displayed in the TextBox on the right***

**private** **void** listDisplay\_SelectedIndexChanged**(**object sender**,** EventArgs e**)**

**{**

**if(**listDisplay**.**SelectedIndex **>** **-**1 **)**

**{**

singleDataTextBox**.**Text **=** listDisplay**.**SelectedItem**.**ToString**();**

**}**

**}**

***// Prevent user from pressing ENTER key***

**private** **void** textBoxInput\_KeyPress**(**object sender**,** KeyPressEventArgs e**)**

**{**

**if(**e**.**KeyChar **==** **(char)**Keys**.**Enter**)**

**{**

e**.**Handled **=** **true;**

**}**

**}**

***// Prevent user from entering any text or numbers***

**private** **void** singleDataTextBox\_KeyPress**(**object sender**,** KeyPressEventArgs e**)**

**{**

**if** **(!char.**IsDigit**(**e**.**KeyChar**)** **||** **(**e**.**KeyChar **==** **(char)**Keys**.**Enter**)** **||** **(char.**IsDigit**(**e**.**KeyChar**)))**

**{**

e**.**Handled **=** **true;**

**}**

**}**

***// Clear textbox and focus cursor in Text box***

**private** **void** ccTextBox**()**

**{**

***// Clear textbox***

***// Cursor focus on textbox***

textBoxInput**.**Clear**();**

textBoxInput**.**Select**();**

**}**

#endregion FUNCTIONS

#region BUTTONS\_1

***// CR: Add new rego plate***

***// PR: User can type data value into the textbox***

***// PR: Clear textbox and focus cursor in the textbox when data is added***

***// PR: Generate error message if textbox is empty***

**private** **void** addButton\_Click**(**object sender**,** EventArgs e**)**

**{**

***/\****

***if (!string.IsNullOrEmpty(textBoxInput.Text) && !RegoPlate.Contains(textBoxInput.Text))***

***{***

***RegoPlate.Add(textBoxInput.Text);***

***toolStripStatusLabel1.Text = "-";***

***ccTextBox();***

***DisplayList();***

***}***

***else if (RegoPlate.Contains(textBoxInput.Text))***

***{***

***MessageBox.Show("Add Fail");***

***toolStripStatusLabel1.Text = "Registration Plate already exist";***

***ccTextBox();***

***}***

***else***

***{***

***MessageBox.Show("Add Fail");***

***toolStripStatusLabel1.Text = "Text Box is empty";***

***ccTextBox();***

***}***

***\*/***

***// Temp variable***

string stmnt**;**

***// Terinary Operator = Condition ? Statement : Statement;***

***// Check if textbox is not empty and Registration Plate does not exist***

var checkCond **=** **!**string**.**IsNullOrEmpty**(**textBoxInput**.**Text**)** **&&** **!**RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** "1" **:** stmnt **=** "2"**;**

***// If Registration plate exist***

var checkCond2 **=** RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** "3" **:** **null;**

**switch** **(**stmnt**)**

**{**

**case** "1"**:**

RegoPlate**.**Add**(**textBoxInput**.**Text**);**

toolStripStatusLabel1**.**Text **=** "Add Success"**;**

ccTextBox**();**

DisplayList**();**

**break;**

**case** "2"**:**

MessageBox**.**Show**(**"Add Fail"**);**

toolStripStatusLabel1**.**Text **=** "Text box is empty"**;**

ccTextBox**();**

**break;**

**case** "3"**:**

MessageBox**.**Show**(**"Add Fail"**);**

toolStripStatusLabel1**.**Text **=** "Registration Plate already exist"**;**

ccTextBox**();**

**break;**

**}**

**}**

***// CR: Delete an existing rego plate***

***// PR: Two methods to remove registration plate from the list***

***// PR: Clear textbox and focus cursor in the textbox when data is removed***

***// PR: Remove by double clicking data item from the ListBox***

**private** **void** listDisplay\_MouseDoubleClick**(**object sender**,** MouseEventArgs e**)**

**{**

toolStripStatusLabel1**.**Text **=** "Remove Success"**;**

RegoPlate**.**Remove**(**singleDataTextBox**.**Text**);**

singleDataTextBox**.**Clear**();**

DisplayList**();**

**}**

***// PR: Remove by entering the rego plate information into the TextBox***

**private** **void** deleteButton\_Click**(**object sender**,** EventArgs e**)**

**{**

***/\****

***// Check if textbox is not empty and regoplate exist***

***if (!String.IsNullOrEmpty(textBoxInput.Text) && RegoPlate.Contains(textBoxInput.Text))***

***{***

***RegoPlate.Remove(textBoxInput.Text);***

***toolStripStatusLabel1.Text = "Remove Success";***

***ccTextBox();***

***DisplayList();***

***}***

***else***

***{***

***MessageBox.Show("Remove Fail");***

***toolStripStatusLabel1.Text = "Registration Plate does not exist";***

***ccTextBox();***

***}***

***\*/***

***// Temp variable***

string stmnt**;**

***// Terinary Operator Condition ? Statement : Statement;***

***// Check if textbox is not empty and registration plate exist***

var checkCond **=** **!**string**.**IsNullOrEmpty**(**textBoxInput**.**Text**)** **&&** RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** "1" **:** stmnt **=** "2"**;**

***// If registration does not exist***

var checkCond2 **=** **!**string**.**IsNullOrEmpty**(**textBoxInput**.**Text**)** **&&** **!**RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** "3" **:** **null;**

**switch** **(**stmnt**)**

**{**

**case** "1"**:**

RegoPlate**.**Remove**(**textBoxInput**.**Text**);**

toolStripStatusLabel1**.**Text **=** "Remove Success"**;**

ccTextBox**();**

DisplayList**();**

**break;**

**case** "2"**:**

MessageBox**.**Show**(**"Remove Fail"**);**

toolStripStatusLabel1**.**Text **=** "Text box is empty"**;**

ccTextBox**();**

**break;**

**case** "3"**:**

MessageBox**.**Show**(**"Remove Fail"**);**

toolStripStatusLabel1**.**Text **=** "Registration Plate does not exist"**;**

ccTextBox**();**

**break;**

**}**

**}**

***// CR: Edit or update an existing rego plate***

***// PR: Select an item from the ListBox to edit***

***// PR: Clear textbox and focus cursor in the textbox when data is edited***

***// Change Registration plate in the list***

**private** **void** buttonEdit\_Click**(**object sender**,** EventArgs e**)**

**{**

***/\****

***// Check if textbox is not empty and selected an item***

***if (listDisplay.SelectedIndex > -1 && !String.IsNullOrEmpty(textBoxInput.Text))***

***{***

***RegoPlate[listDisplay.SelectedIndex] = textBoxInput.Text;***

***toolStripStatusLabel1.Text = "Edit Success";***

***ccTextBox();***

***DisplayList();***

***}***

***else if (String.IsNullOrEmpty(textBoxInput.Text))***

***{***

***MessageBox.Show("Edit Fail");***

***toolStripStatusLabel1.Text = "Text Box is empty";***

***ccTextBox();***

***}***

***else if (!RegoPlate.Contains(textBoxInput.Text))***

***{***

***MessageBox.Show("Edit Fail");***

***toolStripStatusLabel1.Text = "Registration Plate already exist";***

***ccTextBox();***

***}***

***\*/***

***// Temp variable***

string stmnt**;**

***// Terinary Operator Condition ? Statement : Statement;***

***// Check if text box is not empty and item is selected***

var checkCond **=** **!**string**.**IsNullOrEmpty**(**textBoxInput**.**Text**)** **&&** listDisplay**.**SelectedIndex **>** **-**1 **?** stmnt **=** "1" **:** stmnt **=** "2"**;**

***// If input already exist in the list***

var checkCond2 **=** RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** "3" **:** **null;**

**switch** **(**stmnt**)**

**{**

**case** "1"**:**

RegoPlate**[**listDisplay**.**SelectedIndex**]** **=** textBoxInput**.**Text**;**

toolStripStatusLabel1**.**Text **=** "Edit Success"**;**

ccTextBox**();**

DisplayList**();**

**break;**

**case** "2"**:**

MessageBox**.**Show**(**"Edit Fail"**);**

toolStripStatusLabel1**.**Text **=** "Text box is empty"**;**

ccTextBox**();**

**break;**

**case** "3"**:**

MessageBox**.**Show**(**"Edit Fail"**);**

toolStripStatusLabel1**.**Text **=** "Registration Plate already exist"**;**

ccTextBox**();**

**break;**

**}**

**}**

#endregion BUTTONS\_1

#region BUTTONS\_2

***// CR: Tag a specific rego plate for future investigation***

***// PR: Add "z" before the rego plate to mark a rego plate***

**private** **void** buttonTag\_Click**(**object sender**,** EventArgs e**)**

**{**

***/\****

***if (listDisplay.SelectedIndex > -1 && !isTagged.Contains(RegoPlate[listDisplay.SelectedIndex]))***

***{***

***RegoPlate[listDisplay.SelectedIndex] = "z " + singleDataTextBox.Text;***

***isTagged.Add(RegoPlate[listDisplay.SelectedIndex]);***

***toolStripStatusLabel1.Text = "Tag Success";***

***DisplayList();***

***}***

***else if (listDisplay.SelectedIndex > -1 && isTagged.Contains(RegoPlate[listDisplay.SelectedIndex]))***

***{***

***MessageBox.Show("Tag Fail");***

***toolStripStatusLabel1.Text = "Registration plate already Tagged";***

***ccTextBox();***

***}***

***else***

***{***

***MessageBox.Show("Tag Fail");***

***toolStripStatusLabel1.Text = "No Registration plate is selected";***

***ccTextBox();***

***}***

***\*/***

***// Temp variable***

string stmnt**;**

***// Terinary Operator = Condition ? Statement : Statement;***

***// Check if item is selected has not yet been tagged***

var checkCond **=** listDisplay**.**SelectedIndex **>** **-**1 **&&** **!**isTagged**.**Contains**(**RegoPlate**[**listDisplay**.**SelectedIndex**])** **?** stmnt **=** "1" **:** stmnt **=** "2"**;**

***// If no item is selected***

var checkCond2 **=** listDisplay**.**SelectedIndex **<** 0 **?** stmnt **=** "3" **:** **null;**

**switch** **(**stmnt**)**

**{**

**case** "1"**:**

RegoPlate**[**listDisplay**.**SelectedIndex**]** **=** "Z " **+** singleDataTextBox**.**Text**;**

isTagged**.**Add**(**RegoPlate**[**listDisplay**.**SelectedIndex**]);**

toolStripStatusLabel1**.**Text **=** "Tag Success"**;**

DisplayList**();**

**break;**

**case** "2"**:**

MessageBox**.**Show**(**"Tag Fail"**);**

toolStripStatusLabel1**.**Text **=** "Registration plate already tagged"**;**

ccTextBox**();**

**break;**

**case** "3"**:**

MessageBox**.**Show**(**"Tag Fail"**);**

toolStripStatusLabel1**.**Text **=** "No Registration Plate is selected"**;**

ccTextBox**();**

**break;**

**}**

**}**

***// CR: Binary Search for a specific rego plate***

***// PR: Locate a particular registration plate using a binary search algorithm***

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

**{**

***/\****

***DisplayList()***

***if (RegoPlate.BinarySearch(textBoxInput.Text) >= 0) // >= 0 is a comparator***

***{***

***listDisplay.SelectedItem = textBoxInput.Text;***

***MessageBox.Show("Found");***

***textBoxInput.Clear();***

***}***

***else***

***{***

***MessageBox.Show("Search Fail");***

***ccTextBox();***

***}***

***\*/***

***// Temp variable***

string stmnt**;**

***// Terinary Operator = Condition ? Statement : Statement;***

***// Check If the registration plate exists. If there is no registration plate, the textbox is empty.***

var checkCond **=** RegoPlate**.**BinarySearch**(**textBoxInput**.**Text**)** **>=** 0 **?** stmnt **=** "1" **:** stmnt **=** "2"**;**

***// If the textbox is not empty, there is no registration plate.***

var checkCond2 **=** **!**string**.**IsNullOrEmpty**(**textBoxInput**.**Text**)** **?** stmnt **=** "3" **:** **null;**

**switch** **(**stmnt**)**

**{**

**case** "1"**:**

listDisplay**.**SelectedItem **=** textBoxInput**.**Text**;**

toolStripStatusLabel1**.**Text **=** "Search Success"**;**

ccTextBox**();**

**break;**

**case** "2"**:**

MessageBox**.**Show**(**"Search Fail"**);**

toolStripStatusLabel1**.**Text **=** "Text box is empty"**;**

ccTextBox**();**

**break;**

**case** "3"**:**

MessageBox**.**Show**(**"Search Fail"**);**

toolStripStatusLabel1**.**Text **=** "Registration Plate does not exist"**;**

ccTextBox**();**

**break;**

**}**

**}**

***// CR: Linear Search for a specific rego plate***

***// PR: Locate a particular registration plate using a linear search algorithm***

**private** **void** buttonLinSearch\_Click**(**object sender**,** EventArgs e**)**

**{**

***/\****

***if(RegoPlate.Contains(textBoxInput.Text))***

***{***

***listDisplay.SelectedItem = textBoxInput.Text;***

***MessageBox.Show("Found");***

***textBoxInput.Clear();***

***}***

***else***

***{***

***MessageBox.Show("Not Found");***

***ccTextBox();***

***}***

***\*/***

***// Temp variable***

string stmnt**;**

***// Terinary Operator = Condition ? Statement : Statement;***

***// Check If the registration plate exists. If there is no registration plate, the textbox is empty.***

var checkCond **=** RegoPlate**.**Contains**(**textBoxInput**.**Text**)?** stmnt **=** "1" **:** stmnt **=** "2"**;**

***// If the textbox is not empty, there is no registration plate.***

var checkCond2 **=** **!**string**.**IsNullOrEmpty**(**textBoxInput**.**Text**)** **?** stmnt **=** "3" **:** **null;**

**switch** **(**stmnt**)**

**{**

**case** "1"**:**

listDisplay**.**SelectedItem **=** textBoxInput**.**Text**;**

toolStripStatusLabel1**.**Text **=** "Search Success"**;**

ccTextBox**();**

**break;**

**case** "2"**:**

MessageBox**.**Show**(**"Search Fail"**);**

toolStripStatusLabel1**.**Text **=** "Text box is empty"**;**

ccTextBox**();**

**break;**

**case** "3"**:**

MessageBox**.**Show**(**"Search Fail"**);**

toolStripStatusLabel1**.**Text **=** "Registration Plate does not exist"**;**

ccTextBox**();**

**break;**

**}**

**}**

***// CR: Save data to text file***

***// Open saved text file***

#endregion Buttons\_2

#region BUTTONS\_3

**private** **void** openButton\_Click**(**object sender**,** EventArgs e**)**

**{**

string fileName **=** "demo\_nn.txt"**;**

OpenFileDialog OpenText **=** **new** OpenFileDialog**();**

DialogResult sr **=** OpenText**.**ShowDialog**();**

**if** **(**sr **==** DialogResult**.**OK**)**

**{**

fileName **=** OpenText**.**FileName**;**

**try**

**{**

RegoPlate**.**Clear**();**

using **(**Stream stream **=** File**.**Open**(**fileName**,** FileMode**.**Open**))**

**{**

BinaryFormatter binaryFormatter **=** **new** BinaryFormatter**();**

**while** **(**stream**.**Position **<** stream**.**Length**)**

**{**

RegoPlate**.**Add**((**string**)**binaryFormatter**.**Deserialize**(**stream**));**

**}**

**}**

DisplayList**();**

**}**

**catch** **(**IOException**)**

**{**

MessageBox**.**Show**(**"Cannot Open File"**);**

**}**

**}**

**}**

***// CR: Open Text File and Load data***

***// PR: User can select different data from pre-saved files***

**private** **void** saveButton\_Click**(**object sender**,** EventArgs e**)**

**{**

string fileName **=** "demo\_##.txt"**;**

SaveFileDialog SaveText **=** **new** SaveFileDialog**();**

DialogResult sr **=** SaveText**.**ShowDialog**();**

**if** **(**sr **==** DialogResult**.**Cancel**)**

**{**

SaveText**.**FileName **=** fileName**;**

**}**

**if** **(**sr **==** DialogResult**.**OK**)**

**{**

fileName **=** SaveText**.**FileName**;**

**}**

**try**

**{**

using **(**Stream stream **=** File**.**Open**(**fileName**,** FileMode**.**Create**))**

**{**

BinaryFormatter binaryFormatter **=** **new** BinaryFormatter**();**

foreach **(**var item in RegoPlate**)**

**{**

binaryFormatter**.**Serialize**(**stream**,** item**);**

**}**

**}**

**}**

**catch** **(**IOException**)**

**{**

MessageBox**.**Show**(**"Cannot Save File"**);**

**}**

**}**

***// CR: Reset button to remove all rego plate data from the List<>***

***// PR: RESET button to erase all items from the List<>***

***// PR: Clear ListBox, Textbox and RegoPlate<> when button is pressed***

**private** **void** buttonReset\_Click**(**object sender**,** EventArgs e**)**

**{**

RegoPlate**.**Clear**();**

textBoxInput**.**Clear**();**

singleDataTextBox**.**Clear**();**

DisplayList**();**

**}**

#endregion BUTTONS\_3

**}**

**}**