NAME: Jervin Alejandro

STUDENT ID: 30009690

CAMPUS: Thornlie TAFE

DATE: 24/11/2021

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**;**

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

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

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

***// Jervin Alejandro***

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

***// Version 1.2***

***// 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 registration 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**>** tagged **=** **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"**;** ***// File to open***

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

**{**

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

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

**{**

RegoPlate**.**Add**((**string**)**binFormat**.**Deserialize**(**openFile**));** ***// Converters 1s and 0s back to original***

**}**

**}**

DisplayList**();**

toolStripStatusLabel1**.**Text **=** "Default text file successfully loaded"**;**

**}**

***// Auto save file when closed***

**private** **void** Lists\_FormClosing**(**object sender**,** FormClosingEventArgs e**)**

**{**

**}**

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

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

**{**

***// Terinary Operater = Condition ? Statement 1 : Statement 2;***

var cond **=** listDisplay**.**SelectedIndex **>** **-**1 **?** singleDataTextBox**.**Text **=** listDisplay**.**SelectedItem**.**ToString**()** **:** **null;**

**}**

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

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

**{**

***// Terinary Operater = Condition ? Statement 1 : Statement 2;***

bool cond **=** e**.**KeyChar **==** **(char)**Keys**.**Enter **?** e**.**Handled **=** **true** **:** e**.**Handled **=** **false;**

**}**

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

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

**{**

***// Terinary Operater = Condition ? Statement 1 : Statement 2;***

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

**?** e**.**Handled **=** **true** **:** e**.**Handled **=** **false;**

**}**

***// 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**)**

**{**

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

***// Condition1: Check if textbox is not empty and Registration Plate does not exist***

***// Condition2: If Registration plate exist***

**int** stmnt **=** **!**string**.**IsNullOrWhiteSpace**(**textBoxInput**.**Text**)** **&&** **!**RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** 1

**:** RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** 2 **:** stmnt **=** 3**;**

***// Prevent invalid registration plate***

***// N = Number***

***// L = Letter***

***// Standard WA rego plate format is NLLLNNN***

**if** **(!**string**.**IsNullOrWhiteSpace**(**textBoxInput**.**Text**)** **&&** **(!char.**IsDigit**(**textBoxInput**.**Text**[**0**])** **||** **!char.**IsLetter**(**textBoxInput**.**Text**[**1**])** **||** **!char.**IsLetter**(**textBoxInput**.**Text**[**2**])** **||** **!char.**IsLetter**(**textBoxInput**.**Text**[**3**])**

**||** **!char.**IsDigit**(**textBoxInput**.**Text**[**4**])** **||** **!char.**IsDigit**(**textBoxInput**.**Text**[**5**])** **||** **!char.**IsDigit**(**textBoxInput**.**Text**[**6**])))**

**{**

stmnt **=** 4**;**

**}**

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

**{**

**case** 1**:**

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

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

ccTextBox**();**

DisplayList**();**

**break;**

**case** 2**:**

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

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

ccTextBox**();**

**break;**

**case** 3**:**

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

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

ccTextBox**();**

**break;**

**case** 4**:**

toolStripStatusLabel1**.**Text **=** "Registration Plate is not valid"**;**

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

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**)**

**{**

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

**{**

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**)**

**{**

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

***// Condition1: Check if textbox is not empty and registration plate exist***

***// Condition2: If registration does not exist***

**int** stmnt **=** **!**string**.**IsNullOrWhiteSpace**(**textBoxInput**.**Text**)** **&&** RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** 1 **:**

**!**string**.**IsNullOrWhiteSpace**(**textBoxInput**.**Text**)** **&&** **!**RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** 2 **:** stmnt **=** 3**;**

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

**{**

**case** 1**:**

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

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

ccTextBox**();**

DisplayList**();**

**break;**

**case** 2**:**

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

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

ccTextBox**();**

**break;**

**case** 3**:**

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

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

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**)**

**{**

***// Temp variable***

string stmnt**;**

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

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

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

***// If input already exist in the list and if nothing is selected***

var checkCond2 **=** RegoPlate**.**Contains**(**textBoxInput**.**Text**)** **?** stmnt **=** "3" **:** **!**string**.**IsNullOrWhiteSpace**(**textBoxInput**.**Text**)** **&&** listDisplay**.**SelectedIndex **<** 0 **?** stmnt **=** "4"**:** **null;**

**if** **(!**string**.**IsNullOrWhiteSpace**(**textBoxInput**.**Text**)** **&&** **(!char.**IsDigit**(**textBoxInput**.**Text**[**0**])** **||** **!char.**IsLetter**(**textBoxInput**.**Text**[**1**])** **||** **!char.**IsLetter**(**textBoxInput**.**Text**[**2**])** **||** **!char.**IsLetter**(**textBoxInput**.**Text**[**3**])**

**||** **!char.**IsDigit**(**textBoxInput**.**Text**[**4**])** **||** **!char.**IsDigit**(**textBoxInput**.**Text**[**5**])** **||** **!char.**IsDigit**(**textBoxInput**.**Text**[**6**])))**

**{**

stmnt **=** "5"**;**

**}**

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

**{**

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

RegoPlate**[**listDisplay**.**SelectedIndex**]** **=** textBoxInput**.**Text**.**ToUpper**();**

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

ccTextBox**();**

DisplayList**();**

**break;**

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

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

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

ccTextBox**();**

**break;**

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

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

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

ccTextBox**();**

**break;**

**case** "4"**:**

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

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

ccTextBox**();**

**break;**

**case** "5"**:**

toolStripStatusLabel1**.**Text **=** "Registration Plate is not valid"**;**

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

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**)**

**{**

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

***// Condition1: Check if item is selected has not yet been tagged***

***// Condition2: If no item is selected***

**int** stmnt **=** listDisplay**.**SelectedIndex **>** **-**1 **&&** **!**tagged**.**Contains**(**RegoPlate**[**listDisplay**.**SelectedIndex**])** **?** stmnt **=** 1

**:** listDisplay**.**SelectedIndex **<** 0 **?** stmnt **=** 2**:** stmnt **=** 3**;**

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

**{**

**case** 1**:**

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

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

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

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

DisplayList**();**

**break;**

**case** 2**:**

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

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

ccTextBox**();**

**break;**

***//case 3:***

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

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

***// ccTextBox();***

***// break;***

***// Untag a registration plate***

**case** 3**:**

tagged**.**Remove**(**RegoPlate**[**listDisplay**.**SelectedIndex**]);**

singleDataTextBox**.**Text **=** singleDataTextBox**.**Text**.**Remove**(**0**,** 2**);**

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

toolStripStatusLabel1**.**Text **=** "Registration plate untagged"**;**

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

DisplayList**();**

**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**)**

**{**

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

***// Condition1: Check If the registration plate exists. If there is no registration plate, the textbox is empty.***

***// Condition2: If the textbox is not empty, there is no registration plate.***

**int** condition **=** RegoPlate**.**BinarySearch**(**textBoxInput**.**Text**)** **>=** 0 **?** condition **=** 1

**:** **!**string**.**IsNullOrWhiteSpace**(**textBoxInput**.**Text**)** **?** condition **=** 2 **:** condition **=** 3**;**

**switch** **(**condition**)**

**{**

**case** 1**:**

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

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

ccTextBox**();**

**break;**

**case** 2**:**

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

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

ccTextBox**();**

**break;**

**case** 3**:**

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

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

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**)**

**{**

bool found **=** **false;**

***// Loop till input text is found***

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

**{**

**if** **(**textBoxInput**.**Text **==** RegoPlate**[**i**])**

**{**

found **=** **true;**

**break;**

**}**

**}**

***// If input text is found***

**if** **(**found **==** **true)**

**{**

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

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

ccTextBox**();**

***// If input is not empty it must be registration must not exist***

**}**

**else** **if(!**string**.**IsNullOrWhiteSpace**(**textBoxInput**.**Text**))**

**{**

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

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

ccTextBox**();**

***// If none of the if statements are true then text box must be empty***

**}**

**else**

**{**

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

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

ccTextBox**();**

**}**

**}**

***// 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\_##.txt"**;** ***// File to open***

OpenFileDialog OpenText **=** **new** OpenFileDialog**();** ***// Dialogbox instance to open a file***

OpenText**.**Filter **=** "Text|\*.txt"**;** ***// Open txt files only***

DialogResult dlg **=** OpenText**.**ShowDialog**();** ***// set dialog result***

***// terinary operator = condition ? statement 1: statement 2:***

**int** cond **=** dlg **==** DialogResult**.**OK **?** cond **=** 1 **:** dlg **==** DialogResult**.**Cancel **?** cond **=** 2 **:** cond **=** 0**;**

**switch** **(**cond**)**

**{**

**case** 1**:**

**try**

**{**

fileName **=** OpenText**.**FileName**;** ***// Get selected text file name and store to filename***

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

using **(**Stream stream **=** File**.**Open**(**fileName**,** FileMode**.**Open**))** ***// Stream allows reading and writing bytes***

**{**

BinaryFormatter binaryFormatter **=** **new** BinaryFormatter**();** ***// Instance of binaryformatter to serialize or deserialize***

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

**{**

RegoPlate**.**Add**((**string**)**binaryFormatter**.**Deserialize**(**stream**));** ***// Converters 1s and 0s back to original***

**}**

**}**

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

DisplayList**();**

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

**}**

***// Error trapping***

**catch** **(**Exception**)**

**{**

toolStripStatusLabel1**.**Text **=** "Text File did not come from the program"**;**

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

**}**

**break;**

**case** 2**:**

toolStripStatusLabel1**.**Text **=** "Load Cancelled"**;**

**break;**

**}**

**}**

***// 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"**;** ***// File name to save***

SaveFileDialog SaveText **=** **new** SaveFileDialog**();** ***// Dialogbox instance to save a file***

SaveText**.**Filter **=** "Text|\*.txt"**;** ***// Save txt files only***

SaveText**.**FileName **=** "demo\_##.txt"**;** ***// Default file name***

DialogResult dlg **=** SaveText**.**ShowDialog**();** ***// set dialog result***

***// terinary operator = condition ? statement 1: statement 2:***

**int** cond **=** dlg **==** DialogResult**.**Cancel **?** cond **=** 1 **:** dlg **==** DialogResult**.**OK **?** cond **=** 2**:** cond **=** 0**;**

**switch** **(**cond**)**

**{**

**case** 1**:**

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

toolStripStatusLabel1**.**Text **=** "Save Cancelled"**;**

**break;**

**case** 2**:**

fileName **=** SaveText**.**FileName**;** ***// Set text file and store to fileName***

**try**

**{**

***// Get every line of text from the list and save to filename***

using **(**Stream stream **=** File**.**Open**(**fileName**,** FileMode**.**Create**))** ***// Stream allows reading and writing bytes***

**{**

BinaryFormatter binaryF **=** **new** BinaryFormatter**();** ***// Instance of binaryformatter to serialize or deserialize***

foreach **(**var text in RegoPlate**)**

**{**

binaryF**.**Serialize**(**stream**,** text**);** ***// Converters stream to 1s and 0s***

**}**

**}**

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

**}**

**catch** **(**Exception**)**

**{**

MessageBox**.**Show**(**"Cannot Save File"**);** ***// Error trapping***

**}**

**break;**

**}**

**}**

***// 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**)**

**{**

***// Show message when resetting***

DialogResult dlg **=** MessageBox**.**Show**(**"Are you sure you want to RESET your data?"**,** "Confirm Reset"**,** MessageBoxButtons**.**OKCancel**);**

**switch** **(**dlg**)**

**{**

***// Clear Everything***

**case** DialogResult**.**OK**:**

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

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

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

DisplayList**();**

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

**break;**

**case** DialogResult**.**Cancel**:**

toolStripStatusLabel1**.**Text **=** "Reset Cancelled"**;**

**break;**

**}**

**}**

#endregion

**}**

**}**