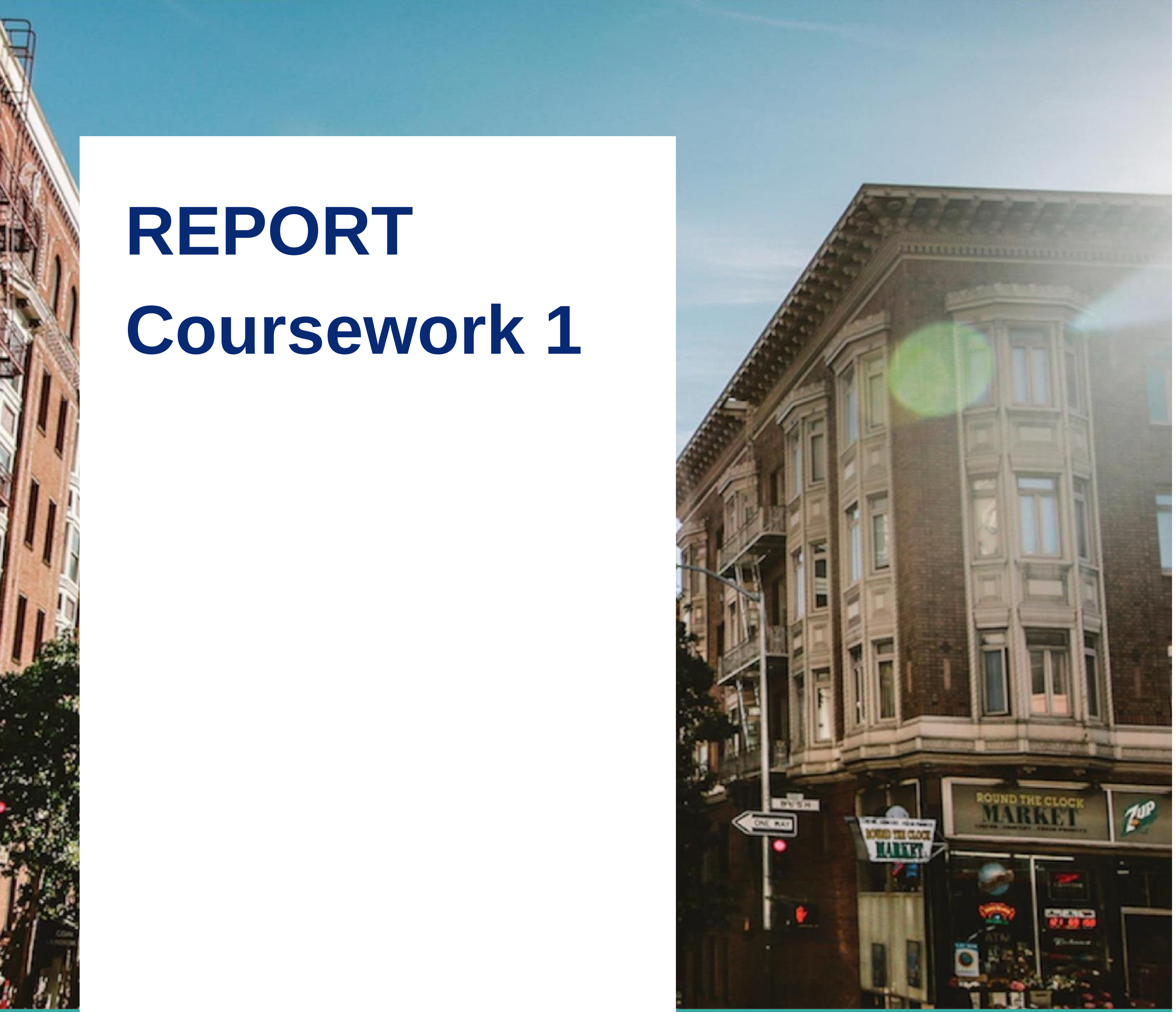
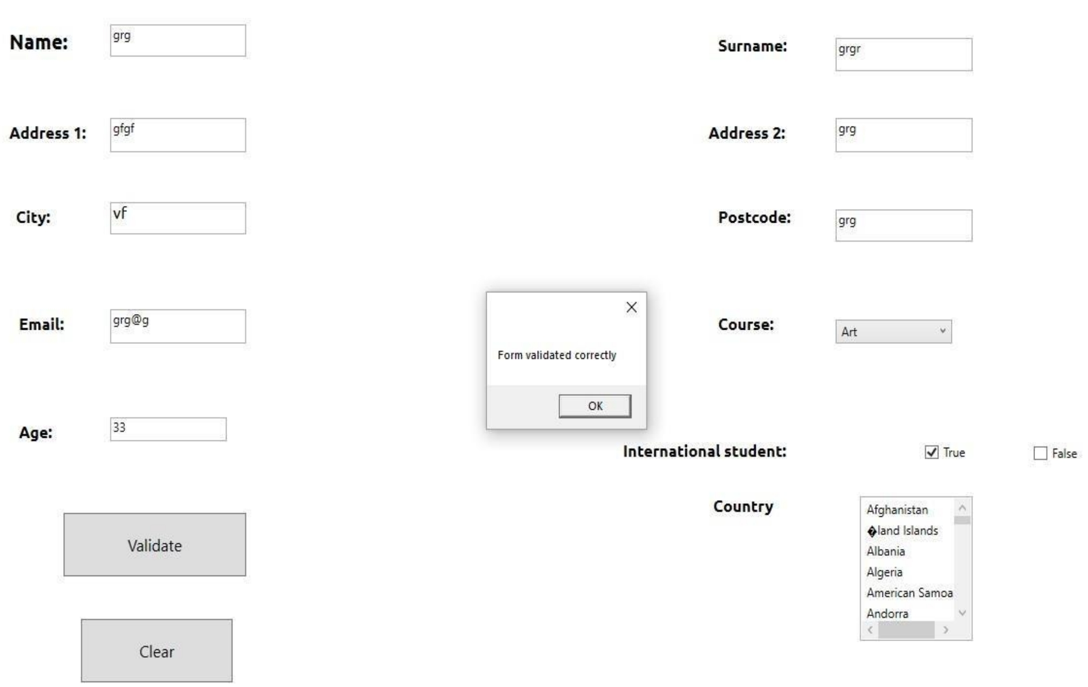


REPORT Coursework 1

OCTOBER 15

Authored by: Davide Pollicino

Documentation



Name: grg

Surname: grgr

Address 1: gfgf

Address 2: grg

City: vf

Postcode: grg

Email: grg@g

Course: Art

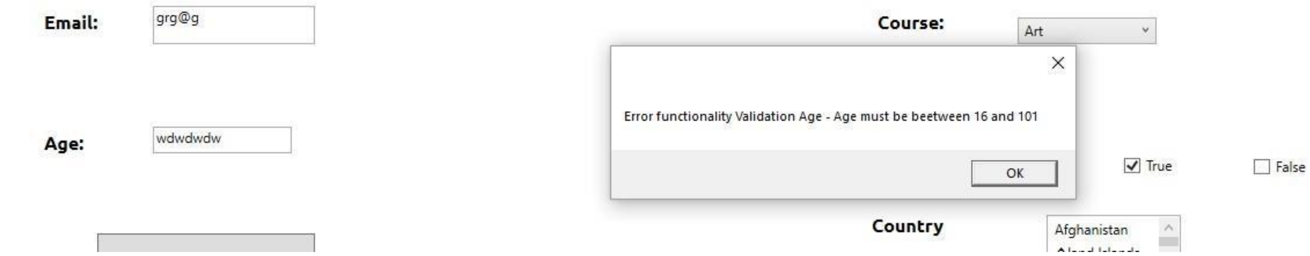
Age: 33

International student: ☑ True ☐ False

Country: Afghanistan, �land Islands, Albania, Algeria, American Samoa, Andorra

Form validated correctly

The age field cannot contains letters.



Email: grg@g

Course: Art

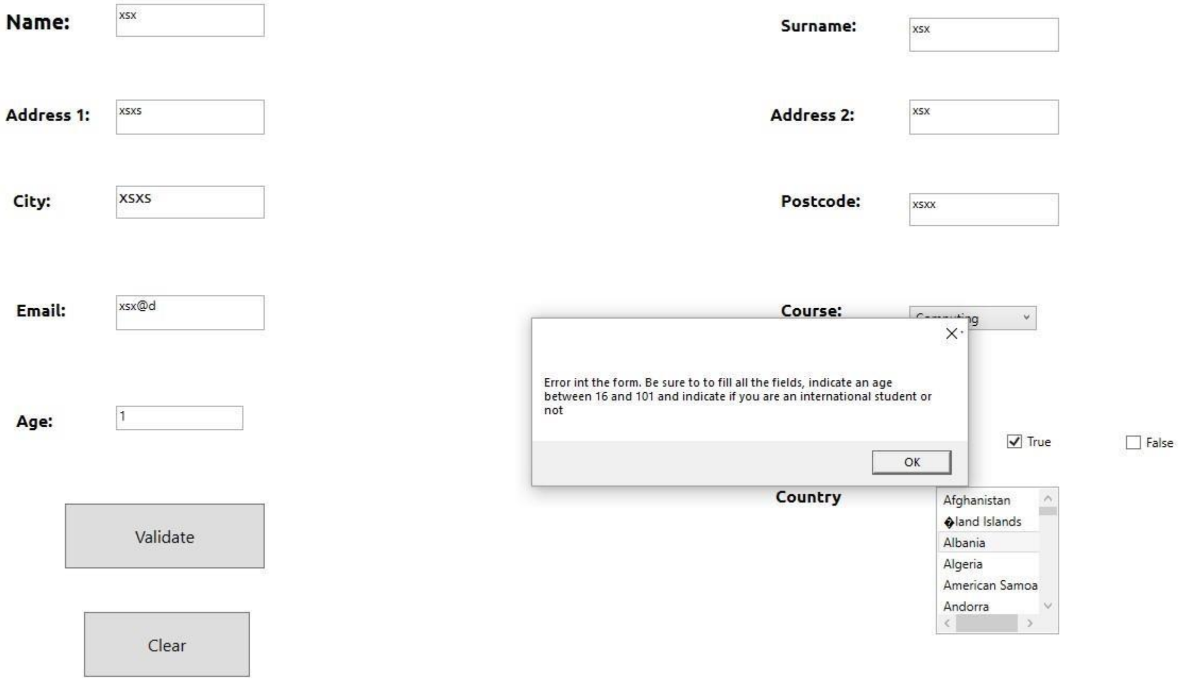
Age: wdwdwdw

☑ True ☐ False

Country: Afghanistan

Error functionality Validation Age - Age must be beetween 16 and 101

Age validation: that must be between 16 and 101:



Name: xsx

Surname: xsx

Address 1: xsxs

Address 2: xsx

City: XSXS

Postcode: xsxx

Email: xsx@d

Course: Computing

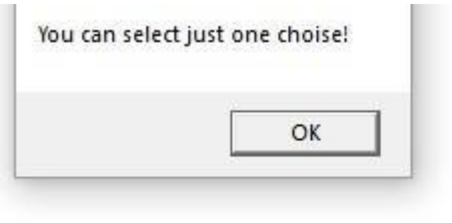
Age: 1

☑ True ☐ False

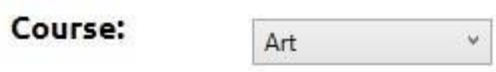
Country: Afghanistan, �land Islands, Albania, Algeria, American Samoa, Andorra

Error int the form. Be sure to to fill all the fields, indicate an age between 16 and 101 and indicate if you are an international student or not

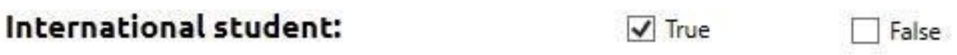
It is not possible to indicate at the same time “True” and “False” for the choice of the international student.



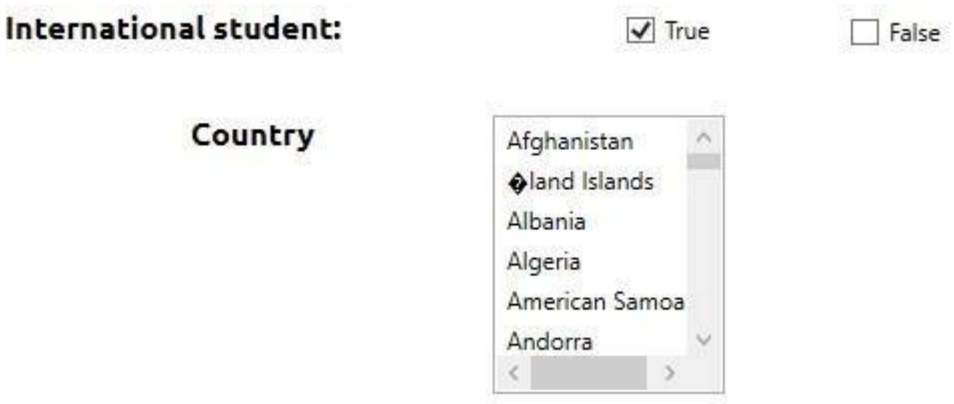
You can select just one choise!



Course: Art



International student: ☑ True □ False

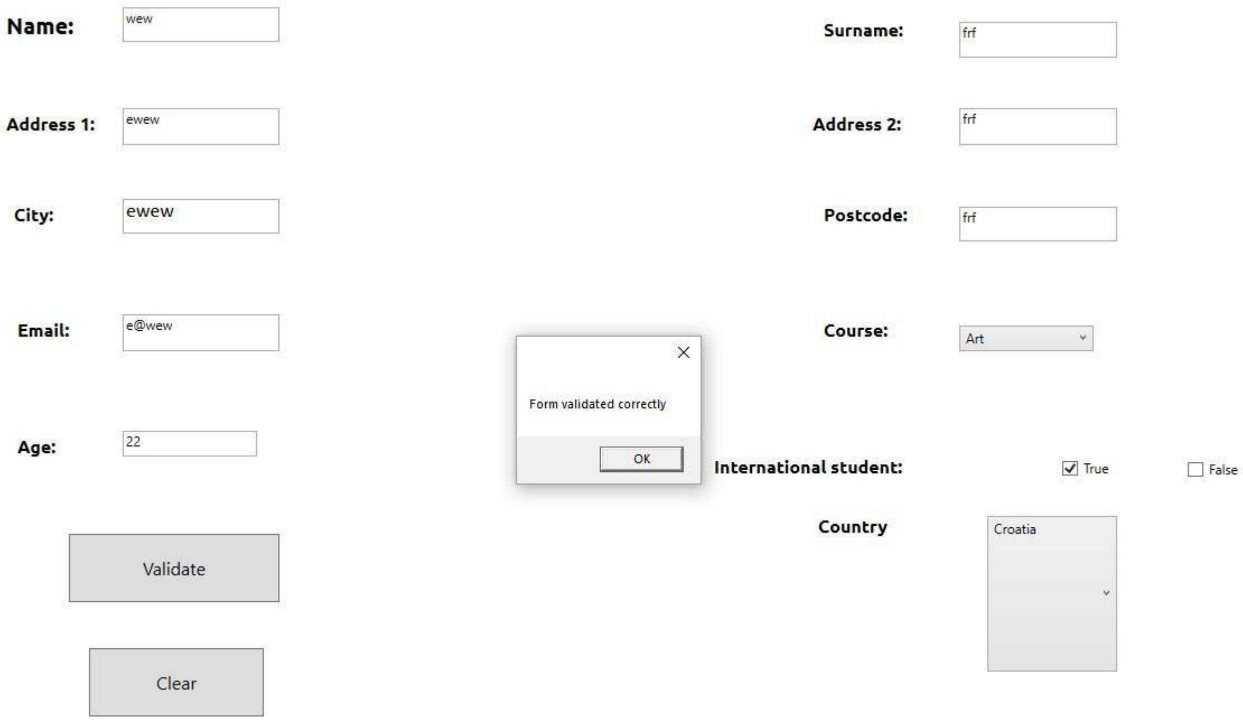


International student: ☑ True □ False

Country

Afghanistan
�land Islands
Albania
Algeria
American Samoa
Andorra

If the fields respect the validation policy indicated in the coursework statement, we will see a positive form validation form.



Name: wew

Surname: frf

Address 1: ewew

Address 2: frf

City: ewew

Postcode: frf

Email: e@wew

Course: Art

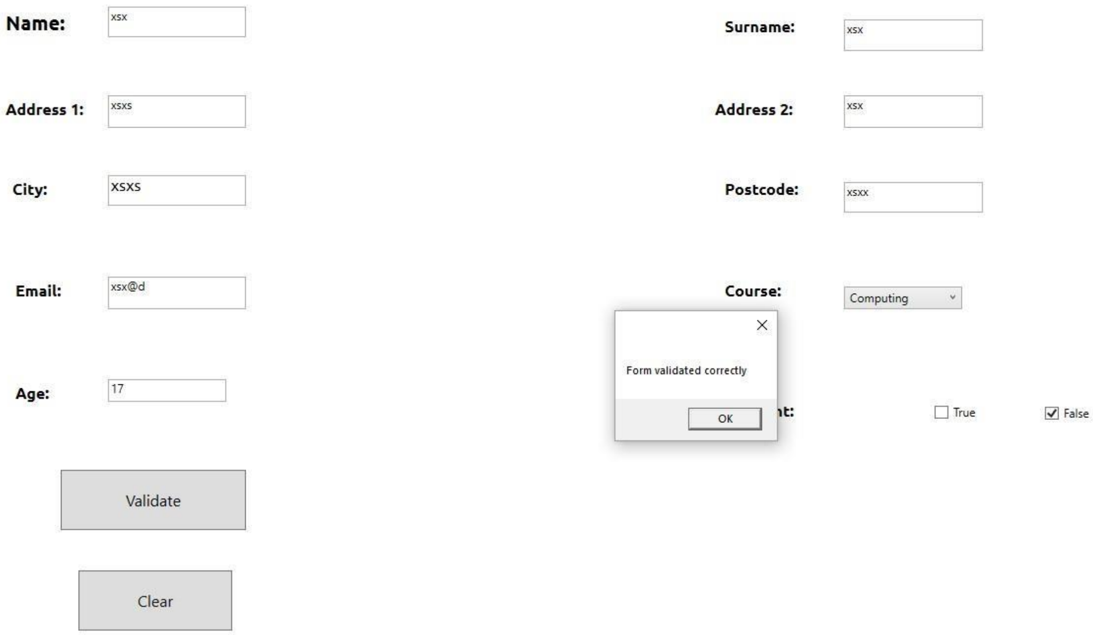
Form validated correctly

Age: 22

International student: ☑ True ☐ False

Country Croatia

Country selection hidden by default and still hidden if the the student is not an internation student



Name: xsx

Surname: xsx

Address 1: xsxs

Address 2: xsx

City: XSXS

Postcode: xsxx

Email: xsx@d

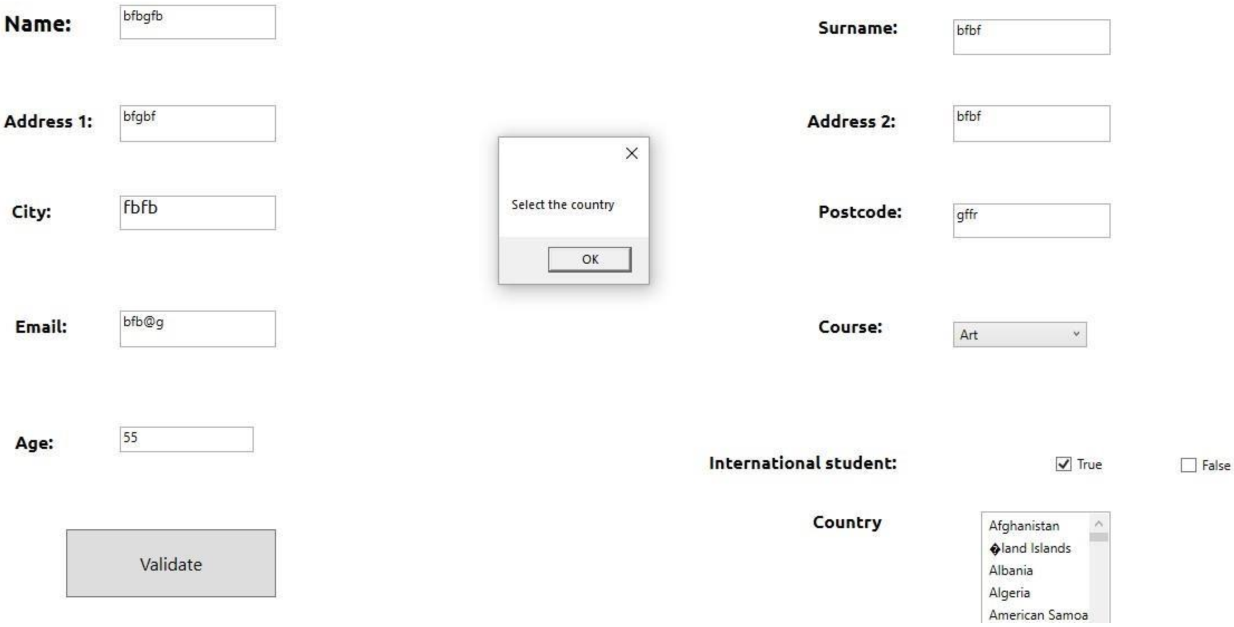
Course: Computing

Form validated correctly

Age: 17

nt: □ True ☑ False

If the student is an international student, the country must be selected



Name: bfbgfb

Surname: bfbf

Address 1: bfgbf

Address 2: bfbf

Select the country

City: fbfb

Postcode: gffr

Email: bfb@g

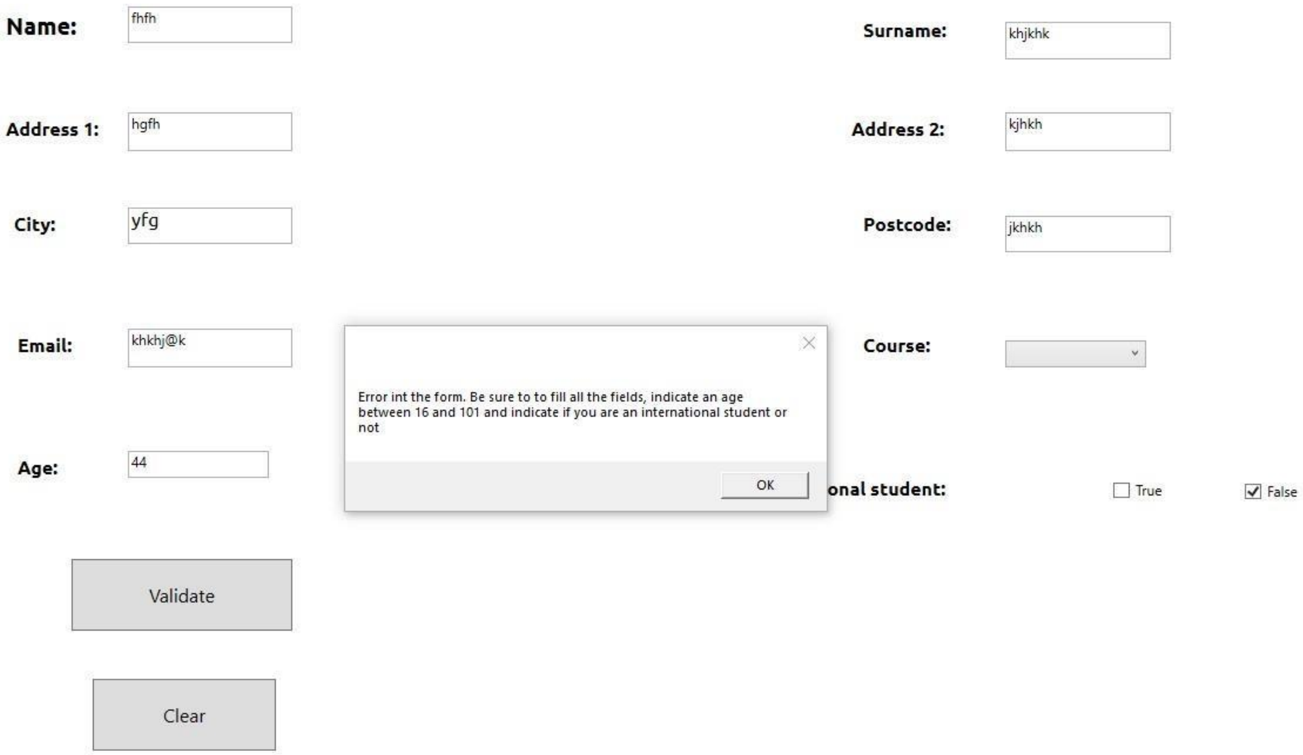
Course: Art

Age: 55

International student: ☑ True □ False

Country: Afghanistan
�land Islands
Albania
Algeria
American Samoa

Also the course must be selected:



Name: fhfh

Surname: khjkhk

Address 1: hgfh

Address 2: kjhkh

City: yfg

Postcode: jkhkh

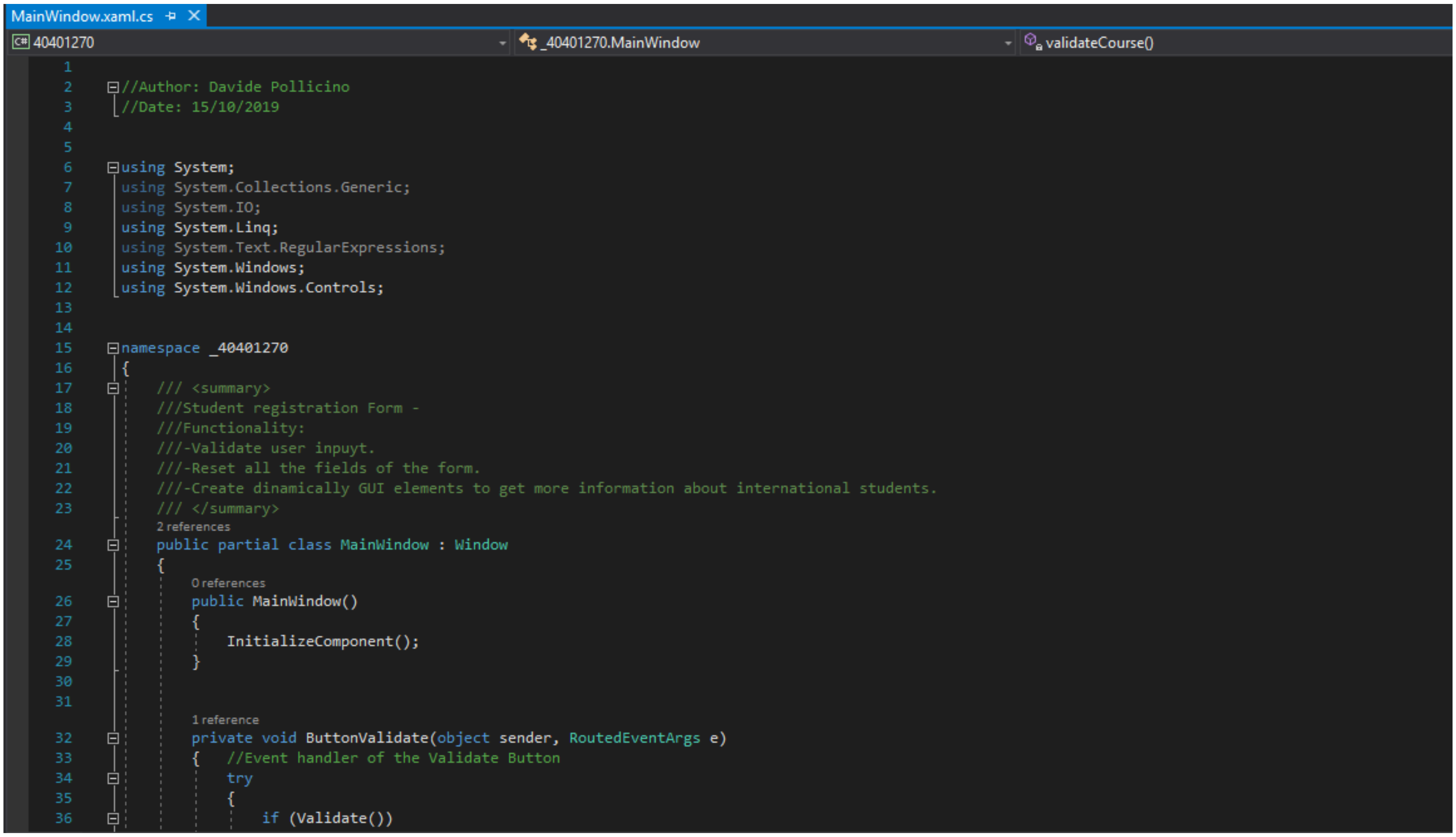
Email: khkhj@k

Course:

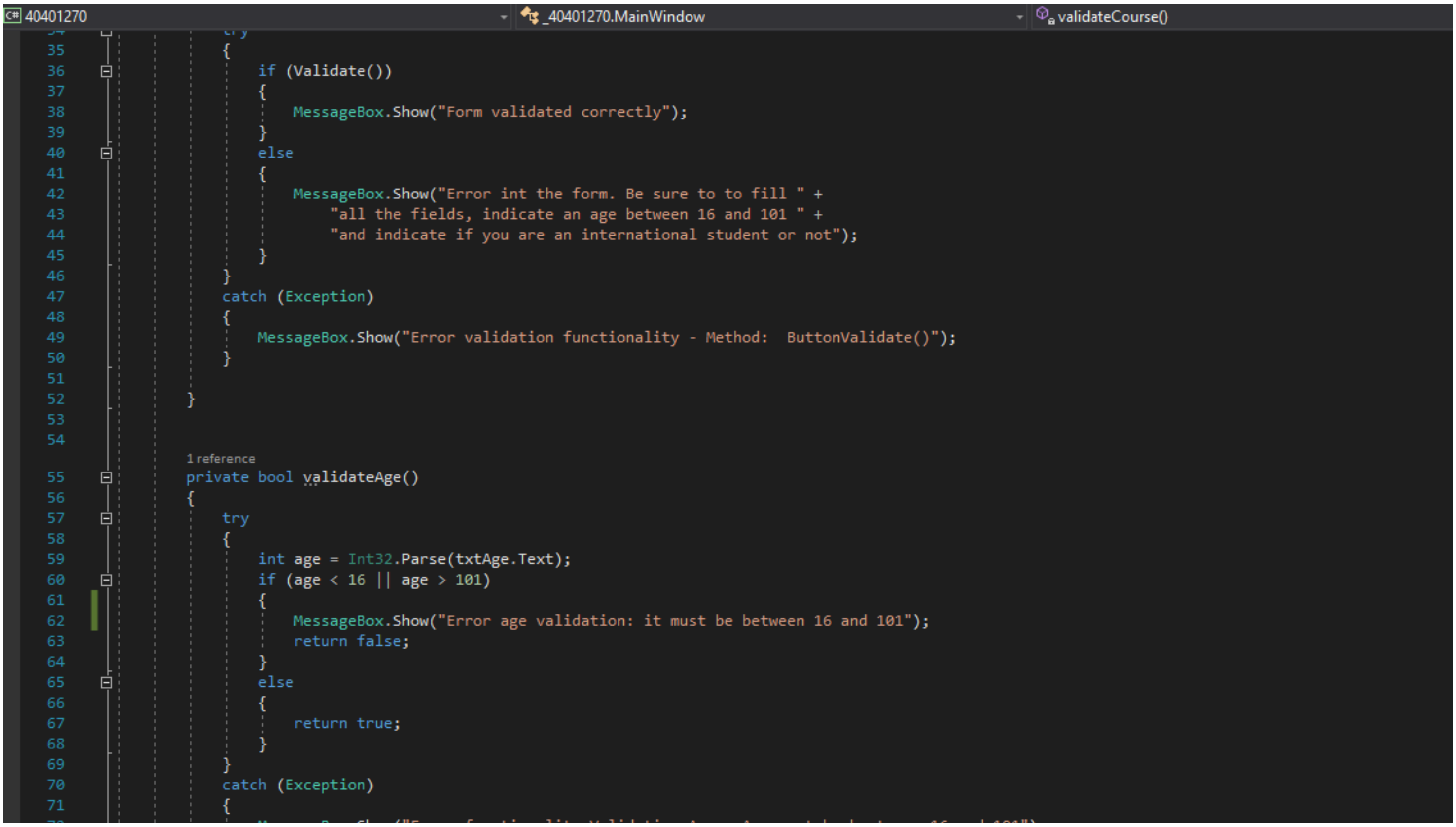
Age: 44

onal student: □ True ☑ False

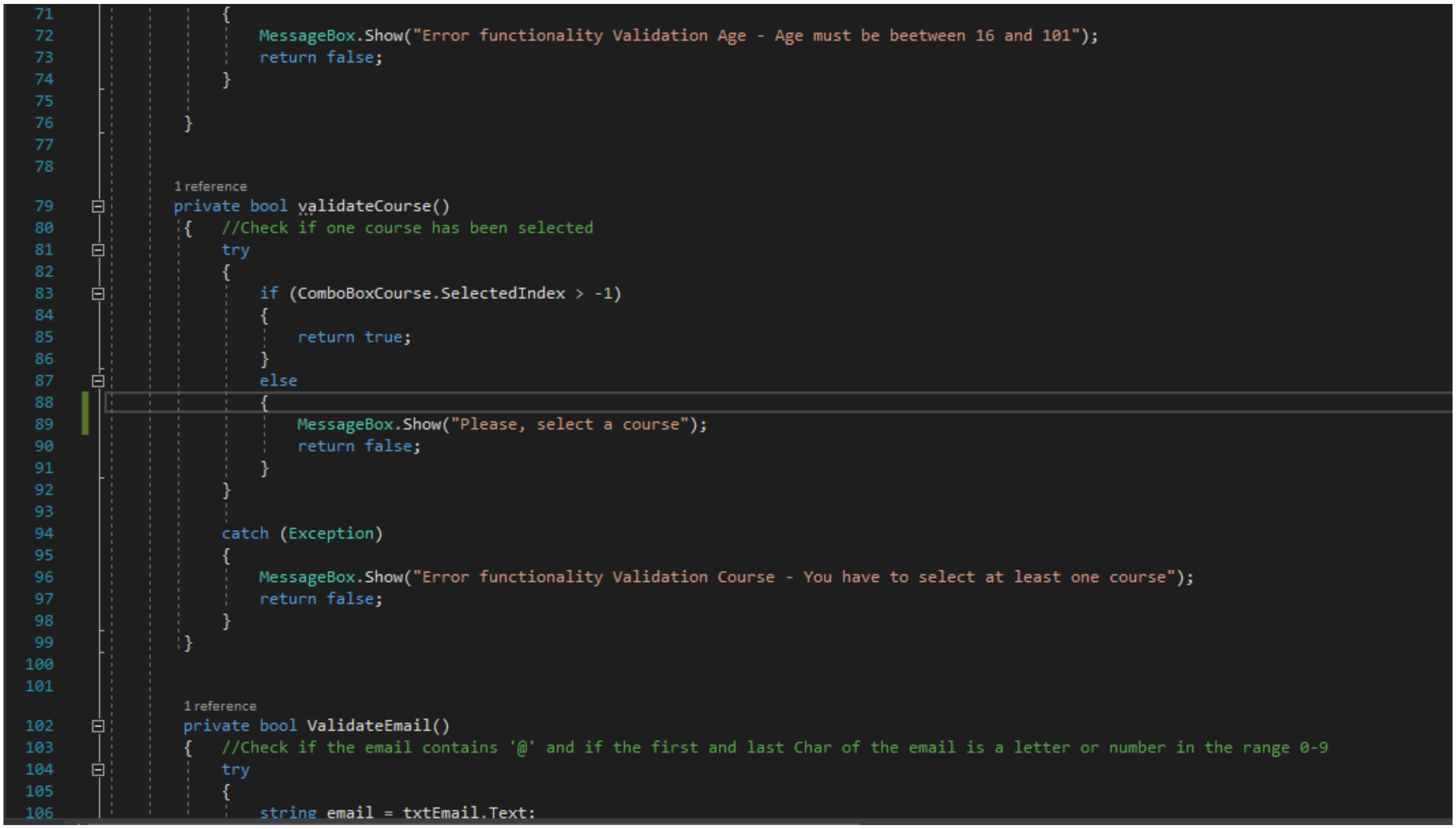
Error int the form. Be sure to to fill all the fields, indicate an age between 16 and 101 and indicate if you are an international student or not



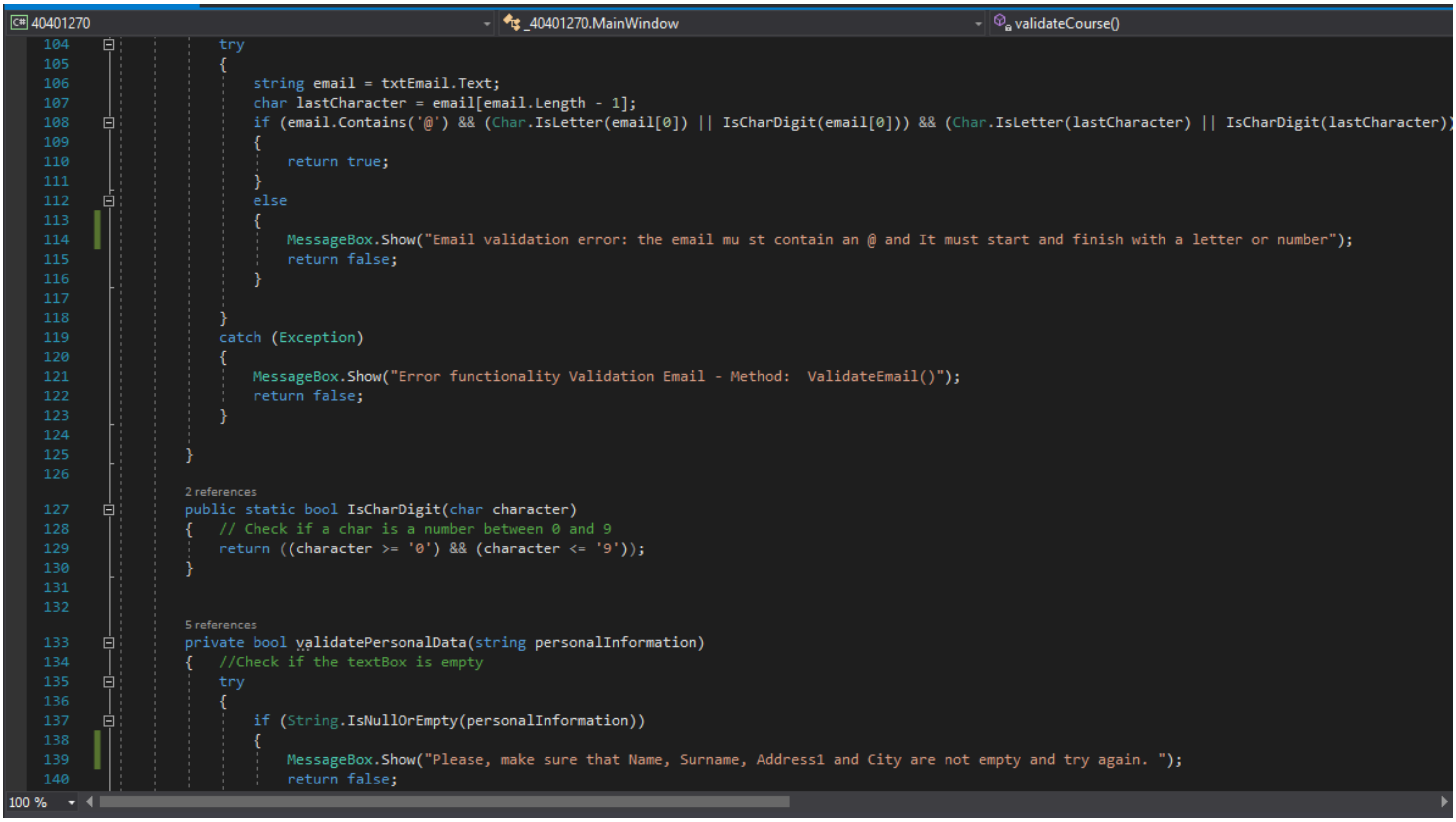
```

//Author: Davide Pollicino
//Date: 15/10/2019


using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Text.RegularExpressions;
using System.Windows;
using System.Windows.Controls;


namespace _40401270
{
    /// <summary>
    ///Student registration Form -
    ///Functionality:
    ///-Validate user inpuyt.
    ///-Reset all the fields of the form.
    ///-Create dinamically GUI elements to get more information about international students.
    /// </summary>
    2 references
    public partial class MainWindow : Window
    {
        0 references
        public MainWindow()
        {
            InitializeComponent();
        }


        1 reference
        private void ButtonValidate(object sender, RoutedEventArgs e)
        {   //Event handler of the Validate Button
            try
            {
                if (Validate())
```



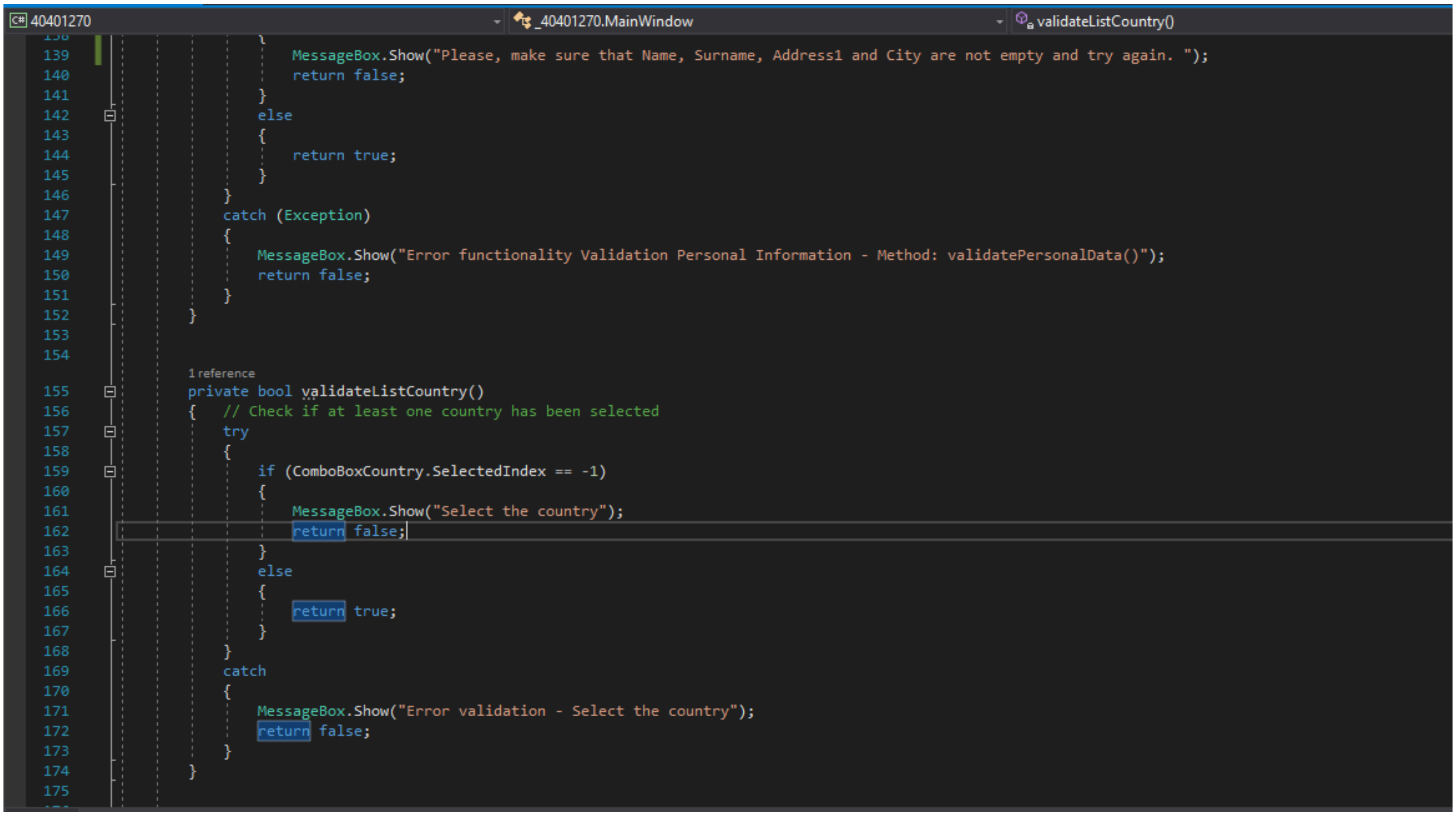
```
        try
        {
            if (Validate())
            {
                MessageBox.Show("Form validated correctly");
            }
            else
            {
                MessageBox.Show("Error int the form. Be sure to to fill " +
                    "all the fields, indicate an age between 16 and 101 " +
                    "and indicate if you are an international student or not");
            }
        }
        catch (Exception)
        {
            MessageBox.Show("Error validation functionality - Method:  ButtonValidate()");
        }

    }


private bool validateAge()
{
    try
    {
        int age = Int32.Parse(txtAge.Text);
        if (age < 16 || age > 101)
        {
            MessageBox.Show("Error age validation: it must be between 16 and 101");
            return false;
        }
        else
        {
            return true;
        }
    }
    catch (Exception)
    {
```



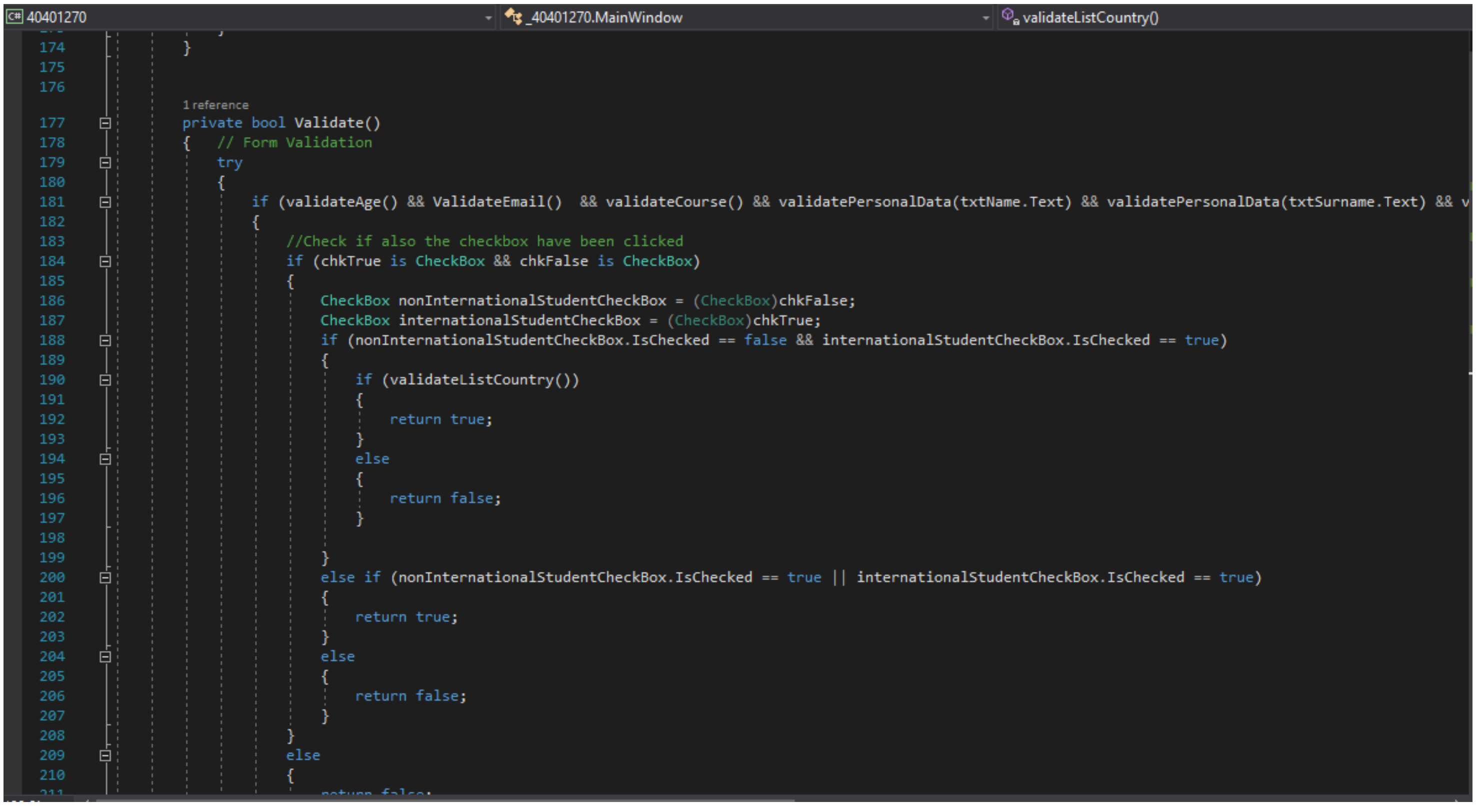
```
            {
                MessageBox.Show("Error functionality Validation Age - Age must be beetween 16 and 101");
                return false;
            }

        }


        1 reference
        private bool validateCourse()
        {   //Check if one course has been selected
            try
            {
                if (ComboBoxCourse.SelectedIndex > -1)
                {
                    return true;
                }
                else
                {
                    MessageBox.Show("Please, select a course");
                    return false;
                }
            }

            catch (Exception)
            {
                MessageBox.Show("Error functionality Validation Course - You have to select at least one course");
                return false;
            }
        }


        1 reference
        private bool ValidateEmail()
        {   //Check if the email contains '@' and if the first and last Char of the email is a letter or number in the range 0-9
            try
            {
                string email = txtEmail.Text;
```



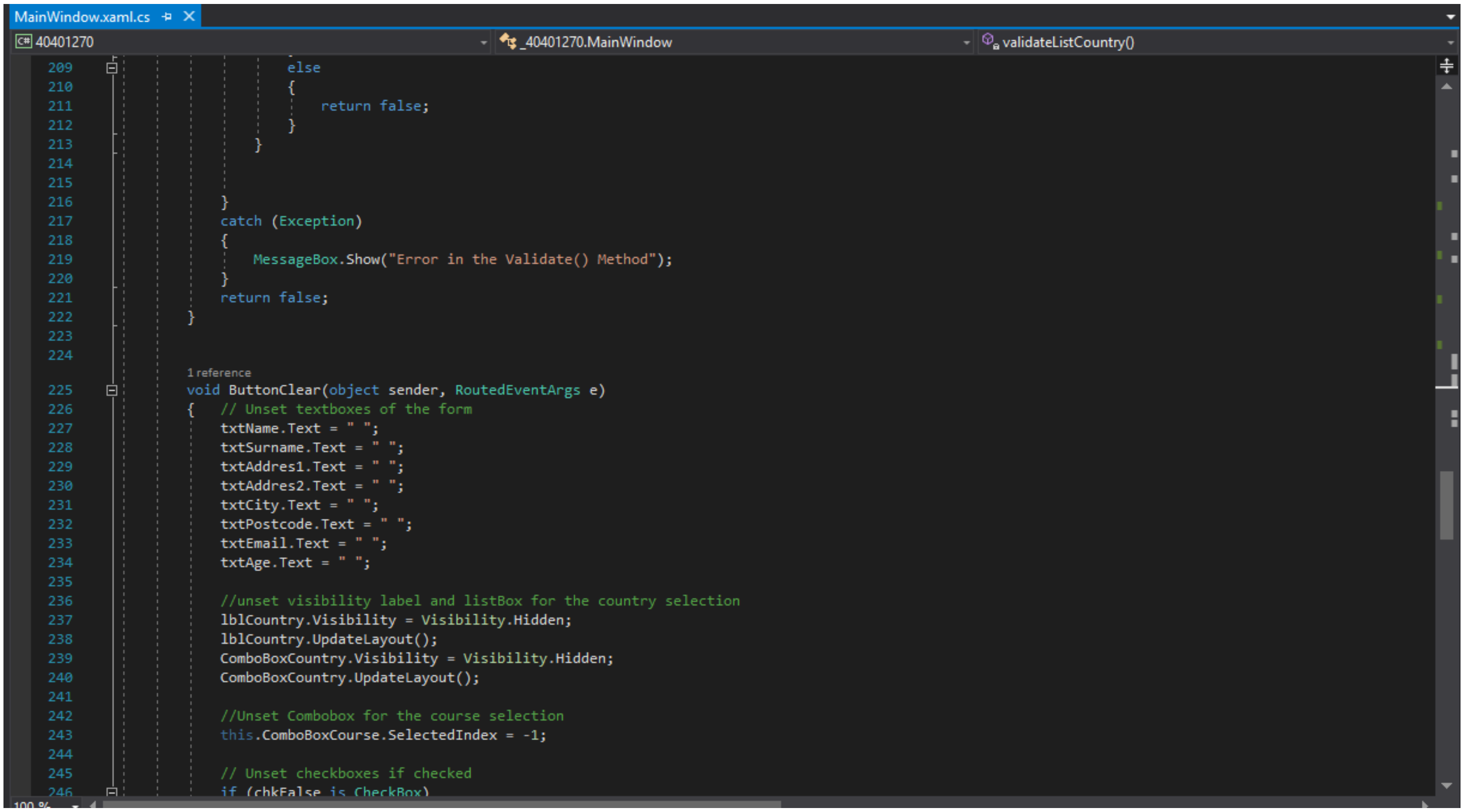
```
            try
            {
                string email = txtEmail.Text;
                char lastCharacter = email[email.Length - 1];
                if (email.Contains('@') && (Char.IsLetter(email[0]) || IsCharDigit(email[0])) && (Char.IsLetter(lastCharacter) || IsCharDigit(lastCharacter))
                {
                    return true;
                }
                else
                {
                    MessageBox.Show("Email validation error: the email mu st contain an @ and It must start and finish with a letter or number");
                    return false;
                }

            }
            catch (Exception)
            {
                MessageBox.Show("Error functionality Validation Email - Method:  ValidateEmail()");
                return false;
            }

        }

        2 references
        public static bool IsCharDigit(char character)
        {   // Check if a char is a number between 0 and 9
            return ((character >= '0') && (character <= '9'));
        }


        5 references
        private bool validatePersonalData(string personalInformation)
        {   //Check if the textBox is empty
            try
            {
                if (String.IsNullOrEmpty(personalInformation))
                {
                    MessageBox.Show("Please, make sure that Name, Surname, Address1 and City are not empty and try again. ");
                    return false;
```



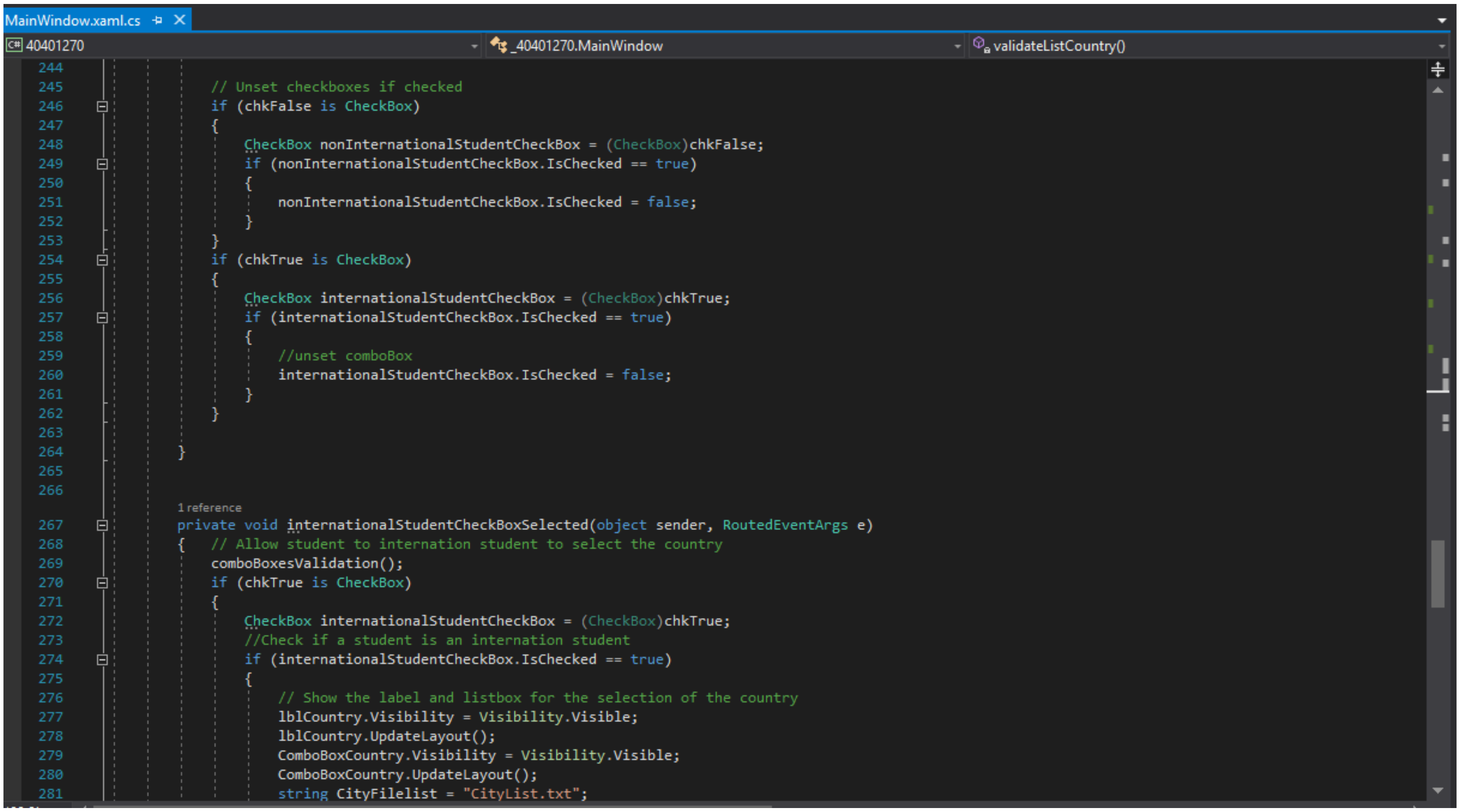
```
                {
                    MessageBox.Show("Please, make sure that Name, Surname, Address1 and City are not empty and try again. ");
                    return false;
                }
                else
                {
                    return true;
                }
            }
            catch (Exception)
            {
                MessageBox.Show("Error functionality Validation Personal Information - Method: validatePersonalData()");
                return false;
            }
        }


        private bool validateListCountry()
        {   // Check if at least one country has been selected
            try
            {
                if (ComboBoxCountry.SelectedIndex == -1)
                {
                    MessageBox.Show("Select the country");
                    return false;
                }
                else
                {
                    return true;
                }
            }
            catch
            {
                MessageBox.Show("Error validation - Select the country");
                return false;
            }
        }
```



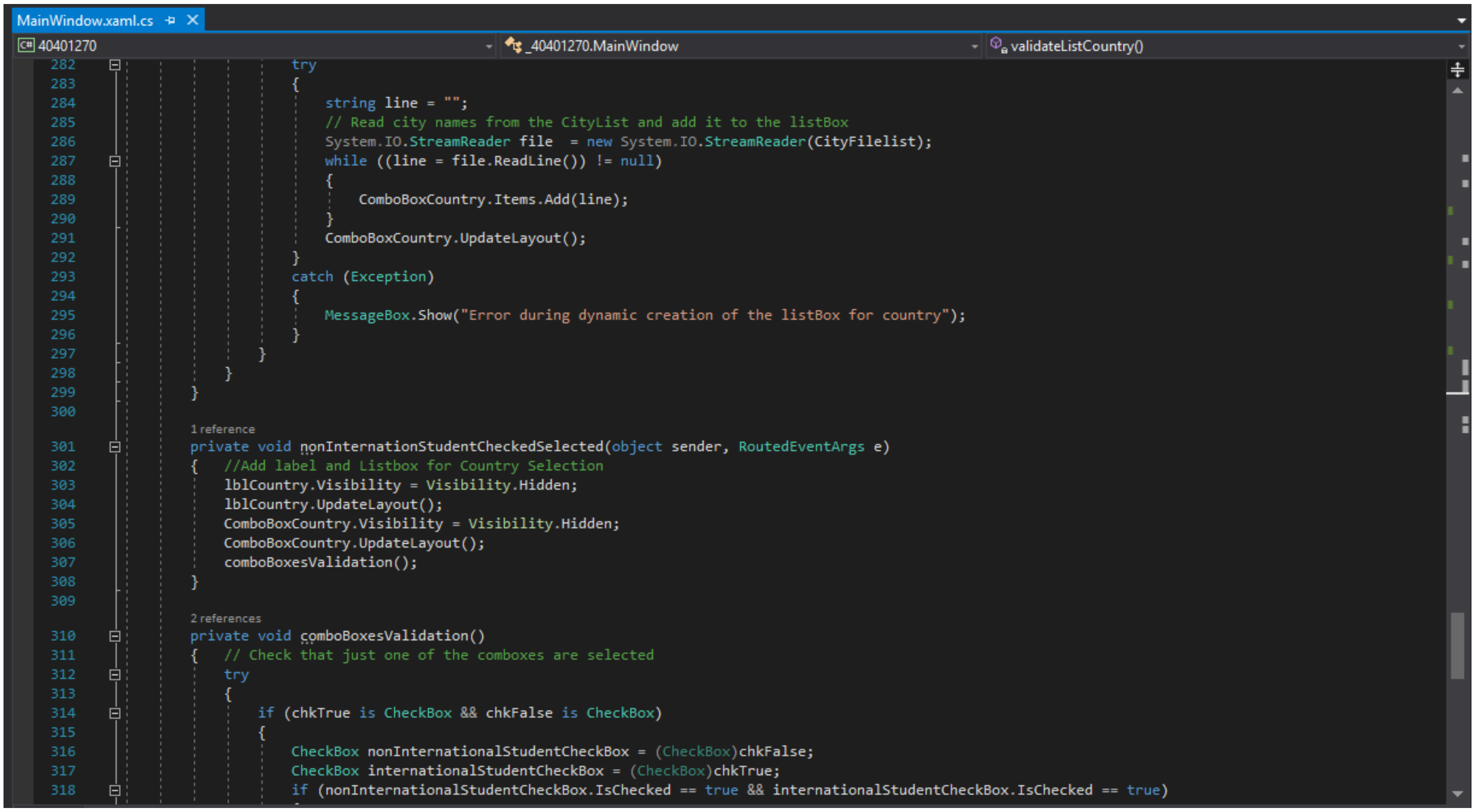
```
        }


        1 reference
        private bool Validate()
        {   // Form Validation
            try
            {
                if (validateAge() && ValidateEmail()  && validateCourse() && validatePersonalData(txtName.Text) && validatePersonalData(txtSurname.Text) && v
                {
                    //Check if also the checkbox have been clicked
                    if (chkTrue is CheckBox && chkFalse is CheckBox)
                    {
                        CheckBox nonInternationalStudentCheckBox = (CheckBox)chkFalse;
                        CheckBox internationalStudentCheckBox = (CheckBox)chkTrue;
                        if (nonInternationalStudentCheckBox.IsChecked == false && internationalStudentCheckBox.IsChecked == true)
                        {
                            if (validateListCountry())
                            {
                                return true;
                            }
                            else
                            {
                                return false;
                            }

                        }
                        else if (nonInternationalStudentCheckBox.IsChecked == true || internationalStudentCheckBox.IsChecked == true)
                        {
                            return true;
                        }
                        else
                        {
                            return false;
                        }
                    }
                    else
                    {
                        return false;
```



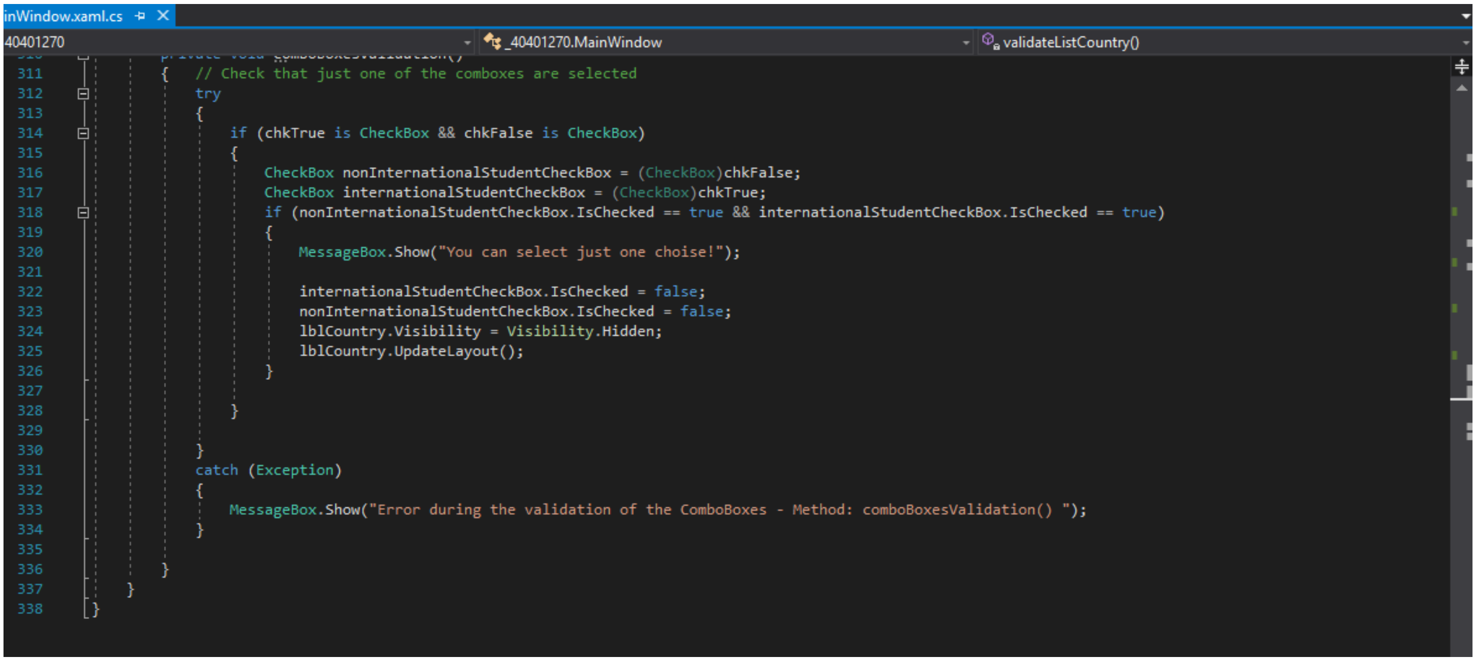
```
                    else
                    {
                        return false;
                    }
                }


            }
            catch (Exception)
            {
                MessageBox.Show("Error in the Validate() Method");
            }
            return false;
        }

        void ButtonClear(object sender, RoutedEventArgs e)
        {   // Unset textboxes of the form
            txtName.Text = " ";
            txtSurname.Text = " ";
            txtAddres1.Text = " ";
            txtAddres2.Text = " ";
            txtCity.Text = " ";
            txtPostcode.Text = " ";
            txtEmail.Text = " ";
            txtAge.Text = " ";

            //unset visibility label and listBox for the country selection
            lblCountry.Visibility = Visibility.Hidden;
            lblCountry.UpdateLayout();
            ComboBoxCountry.Visibility = Visibility.Hidden;
            ComboBoxCountry.UpdateLayout();

            //Unset Combobox for the course selection
            this.ComboBoxCourse.SelectedIndex = -1;

            // Unset checkboxes if checked
            if (chkFalse is CheckBox)
```



```
            // Unset checkboxes if checked
            if (chkFalse is CheckBox)
            {
                CheckBox nonInternationalStudentCheckBox = (CheckBox)chkFalse;
                if (nonInternationalStudentCheckBox.IsChecked == true)
                {
                    nonInternationalStudentCheckBox.IsChecked = false;
                }
            }
            if (chkTrue is CheckBox)
            {
                CheckBox internationalStudentCheckBox = (CheckBox)chkTrue;
                if (internationalStudentCheckBox.IsChecked == true)
                {
                    //unset comboBox
                    internationalStudentCheckBox.IsChecked = false;
                }
            }

        }


        1 reference
        private void internationalStudentCheckBoxSelected(object sender, RoutedEventArgs e)
        {   // Allow student to internation student to select the country
            comboBoxesValidation();
            if (chkTrue is CheckBox)
            {
                CheckBox internationalStudentCheckBox = (CheckBox)chkTrue;
                //Check if a student is an internation student
                if (internationalStudentCheckBox.IsChecked == true)
                {
                    // Show the label and listbox for the selection of the country
                    lblCountry.Visibility = Visibility.Visible;
                    lblCountry.UpdateLayout();
                    ComboBoxCountry.Visibility = Visibility.Visible;
                    ComboBoxCountry.UpdateLayout();
                    string CityFilelist = "CityList.txt";
```



```
                    try
                    {
                        string line = "";
                        // Read city names from the CityList and add it to the listBox
                        System.IO.StreamReader file  = new System.IO.StreamReader(CityFilelist);
                        while ((line = file.ReadLine()) != null)
                        {
                            ComboBoxCountry.Items.Add(line);
                        }
                        ComboBoxCountry.UpdateLayout();
                    }
                    catch (Exception)
                    {
                        MessageBox.Show("Error during dynamic creation of the listBox for country");
                    }
                }
            }
        }

        private void nonInternationStudentCheckedSelected(object sender, RoutedEventArgs e)
        {   //Add label and Listbox for Country Selection
            lblCountry.Visibility = Visibility.Hidden;
            lblCountry.UpdateLayout();
            ComboBoxCountry.Visibility = Visibility.Hidden;
            ComboBoxCountry.UpdateLayout();
            comboBoxesValidation();
        }

        private void comboBoxesValidation()
        {   // Check that just one of the comboxes are selected
            try
            {
                if (chkTrue is CheckBox && chkFalse is CheckBox)
                {
                    CheckBox nonInternationalStudentCheckBox = (CheckBox)chkFalse;
                    CheckBox internationalStudentCheckBox = (CheckBox)chkTrue;
                    if (nonInternationalStudentCheckBox.IsChecked == true && internationalStudentCheckBox.IsChecked == true)
```



```
        {   // Check that just one of the comboxes are selected
            try
            {
                if (chkTrue is CheckBox && chkFalse is CheckBox)
                {
                    CheckBox nonInternationalStudentCheckBox = (CheckBox)chkFalse;
                    CheckBox internationalStudentCheckBox = (CheckBox)chkTrue;
                    if (nonInternationalStudentCheckBox.IsChecked == true && internationalStudentCheckBox.IsChecked == true)
                    {
                        MessageBox.Show("You can select just one choise!");

                        internationalStudentCheckBox.IsChecked = false;
                        nonInternationalStudentCheckBox.IsChecked = false;
                        lblCountry.Visibility = Visibility.Hidden;
                        lblCountry.UpdateLayout();
                    }

                }

            }
            catch (Exception)
            {
                MessageBox.Show("Error during the validation of the ComboBoxes - Method: comboBoxesValidation() ");
            }

        }
    }
}
```