

REPORT

Coursework 1

OCTOBER 15

Authored by: Davide Pollicino



Documentation

Name:

Surname:

Address 1:

Address 2:

City:

Postcode:

Email:

Course:

Age:

International student: ☒ True ☐ False

Country:

Form validated correctly

OK

The age field cannot contains letters.

Email:

Age:

Course:

International student: ☒ True ☐ False

Country:

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

OK

Age validation: that must be between 16 and 101:

Name:	<input type="text" value="xSX"/>	Surname:	<input type="text" value="xSX"/>
Address 1:	<input type="text" value="xSXS"/>	Address 2:	<input type="text" value="xSX"/>
City:	<input type="text" value="XSXS"/>	Postcode:	<input type="text" value="XSXX"/>
Email:	<input type="text" value="xsx@d"/>	Course:	<input type="text" value="Computing"/>
Age:	<input type="text" value="1"/>	Country:	<div><div>Afghanistan</div><div>land Islands</div><div>Albania</div><div>Algeria</div><div>American Samoa</div><div>Andorra</div><div><</div><div>></div></div>
<input checked="" type="checkbox"/> True		<input type="checkbox"/> False	
<div>Validate</div> <div>Clear</div>			

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

OK

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



Course:

International student:

☒ True

☐ False

International student:

☒ True

☐ False

Country

A vertical dropdown menu with a light gray background. The menu is open, showing a list of countries. The visible items are: Afghanistan, Land Islands (with a small diamond icon), Albania, Algeria, American Samoa, and Andorra. Above the list is an upward arrow, and below is a downward arrow. A horizontal scrollbar is visible at the bottom of the list.

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

Name:	<input type="text" value="wew"/>	Surname:	<input type="text" value="frf"/>
Address 1:	<input type="text" value="ewew"/>	Address 2:	<input type="text" value="frf"/>
City:	<input type="text" value="ewew"/>	Postcode:	<input type="text" value="frf"/>
Email:	<input type="text" value="e@wew"/>	Course:	<input type="text" value="Art"/>
Age:	<input type="text" value="22"/>	International student:	<input checked="" type="checkbox"/> True <input type="checkbox"/> False
<input type="button" value="Validate"/>		Country	<div>Croatia</div>
<input type="button" value="Clear"/>			

Form validated correctly
OK

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

Name:

Surname:

Address 1:

Address 2:

City:

Postcode:

Email:

Course:

Age:

International student: ☐ True ☒ False

Validate

Clear



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

Name:

Surname:

Address 1:

Address 2:

City:

Postcode:

Email:

Course:

Age:

International student: ☒ True ☐ False

Validate

Country:



Also the course must be selected:

Name:

Surname:

Address 1:

Address 2:

City:

Postcode:

Email:

Course:

Age:

✕

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

OK

onal student: ☐ True ☒ False

Validate

Clear

```
MainWindow.xaml.cs
40401270
_40401270.MainWindow
validateCourse()

1
2  ///Author: Davide Pollicino
3  ///Date: 15/10/2019
4
5
6  using System;
7  using System.Collections.Generic;
8  using System.IO;
9  using System.Linq;
10 using System.Text.RegularExpressions;
11 using System.Windows;
12 using System.Windows.Controls;
13
14
15 namespace _40401270
16 {
17     /// <summary>
18     /// Student registration Form -
19     /// Functionality:
20     /// -Validate user input.
21     /// -Reset all the fields of the form.
22     /// -Create dynamically GUI elements to get more information about international students.
23     /// </summary>
24     2 references
25     public partial class MainWindow : Window
26     {
27         0 references
28         public MainWindow()
29         {
30             InitializeComponent();
31         }
32
33         1 reference
34         private void ButtonValidate(object sender, RoutedEventArgs e)
35         {
36             //Event handler of the Validate Button
37             try
38             {
39                 if (Validate())
```



```
40401270 - 40401270.MainWindow - validateCourse()
35
36 {
37     if (Validate())
38     {
39         MessageBox.Show("Form validated correctly");
40     }
41     else
42     {
43         MessageBox.Show("Error int the form. Be sure to to fill " +
44             "all the fields, indicate an age between 16 and 101 " +
45             "and indicate if you are an international student or not");
46     }
47 }
48 catch (Exception)
49 {
50     MessageBox.Show("Error validation functionality - Method: ButtonValidate()");
51 }
52 }
53
54
55 1reference
56 private bool validateAge()
57 {
58     try
59     {
60         int age = Int32.Parse(txtAge.Text);
61         if (age < 16 || age > 101)
62         {
63             MessageBox.Show("Error age validation: it must be between 16 and 101");
64             return false;
65         }
66         else
67         {
68             return true;
69         }
70     }
71     catch (Exception)
72     {
73         MessageBox.Show("Error age validation: it must be between 16 and 101");
74     }
75 }
```

```

71     {
72         MessageBox.Show("Error functionality Validation Age - Age must be between 16 and 101");
73         return false;
74     }
75
76 }
77
78
79 1reference
80 private bool validateCourse()
81 { //Check if one course has been selected
82     try
83     {
84         if (ComboBoxCourse.SelectedIndex > -1)
85         {
86             return true;
87         }
88         else
89         {
90             MessageBox.Show("Please, select a course");
91             return false;
92         }
93     }
94     catch (Exception)
95     {
96         MessageBox.Show("Error functionality Validation Course - You have to select at least one course");
97         return false;
98     }
99 }
100
101
102 1reference
103 private bool ValidateEmail()
104 { //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
105     try
106     {
107         string email = txtEmail.Text;

```

```
40401270 - 40401270.MainWindow - validateCourse()
104 try
105 {
106     string email = txtEmail.Text;
107     char lastCharacter = email[email.Length - 1];
108     if (email.Contains('@') && (Char.IsLetter(email[0]) || IsCharDigit(email[0])) && (Char.IsLetter(lastCharacter) || IsCharDigit(lastCharacter)))
109     {
110         return true;
111     }
112     else
113     {
114         MessageBox.Show("Email validation error: the email must contain an @ and It must start and finish with a letter or number");
115         return false;
116     }
117 }
118 }
119 catch (Exception)
120 {
121     MessageBox.Show("Error functionality Validation Email - Method: ValidateEmail()");
122     return false;
123 }
124 }
125 }
126 }
127 2 references
128 public static bool IsCharDigit(char character)
129 {
130     // Check if a char is a number between 0 and 9
131     return ((character >= '0') && (character <= '9'));
132 }
133 5 references
134 private bool validatePersonalData(string personalInformation)
135 {
136     //Check if the textBox is empty
137     try
138     {
139         if (String.IsNullOrEmpty(personalInformation))
140         {
141             MessageBox.Show("Please, make sure that Name, Surname, Address1 and City are not empty and try again. ");
142             return false;
143         }
144     }
145 }
```

```
40401270 40401270.MainWindow validateListCountry()
138
139     MessageBox.Show("Please, make sure that Name, Surname, Address1 and City are not empty and try again. ");
140     return false;
141 }
142 else
143 {
144     return true;
145 }
146 }
147 catch (Exception)
148 {
149     MessageBox.Show("Error functionality Validation Personal Information - Method: validatePersonalData()");
150     return false;
151 }
152 }
153
154
155 1 reference
156 private bool validateListCountry()
157 { // Check if at least one country has been selected
158     try
159     {
160         if (ComboBoxCountry.SelectedIndex == -1)
161         {
162             MessageBox.Show("Select the country");
163             return false;
164         }
165         else
166         {
167             return true;
168         }
169     }
170     catch
171     {
172         MessageBox.Show("Error validation - Select the country");
173         return false;
174     }
175 }
```

```

174     }
175
176
177     1 reference
178     private bool Validate()
179     { // Form Validation
180         try
181         {
182             if (validateAge() && ValidateEmail() && validateCourse() && validatePersonalData(txtName.Text) && validatePersonalData(txtSurname.Text) && v
183             {
184                 //Check if also the checkbox have been clicked
185                 if (chkTrue is CheckBox && chkFalse is CheckBox)
186                 {
187                     CheckBox nonInternationalStudentCheckBox = (CheckBox)chkFalse;
188                     CheckBox internationalStudentCheckBox = (CheckBox)chkTrue;
189                     if (nonInternationalStudentCheckBox.IsChecked == false && internationalStudentCheckBox.IsChecked == true)
190                     {
191                         if (validateListCountry())
192                         {
193                             return true;
194                         }
195                         else
196                         {
197                             return false;
198                         }
199                     }
200                     else if (nonInternationalStudentCheckBox.IsChecked == true || internationalStudentCheckBox.IsChecked == true)
201                     {
202                         return true;
203                     }
204                     else
205                     {
206                         return false;
207                     }
208                 }
209             }
210             else
211             {
212                 return false;

```

```
MainWindow.xaml.cs x
40401270 _40401270.MainWindow validateListCountry()
209 else
210 {
211     return false;
212 }
213
214
215
216 }
217 catch (Exception)
218 {
219     MessageBox.Show("Error in the Validate() Method");
220 }
221 return false;
222 }
223
224
225 //reference
226 void ButtonClear(object sender, RoutedEventArgs e)
227 {
228     // Unset textboxes of the form
229     txtName.Text = " ";
230     txtSurname.Text = " ";
231     txtAddress1.Text = " ";
232     txtAddress2.Text = " ";
233     txtCity.Text = " ";
234     txtPostcode.Text = " ";
235     txtEmail.Text = " ";
236     txtAge.Text = " ";
237
238     //unset visibility label and listBox for the country selection
239     lblCountry.Visibility = Visibility.Hidden;
240     lblCountry.UpdateLayout();
241     ComboBoxCountry.Visibility = Visibility.Hidden;
242     ComboBoxCountry.UpdateLayout();
243
244     //Unset Combobox for the course selection
245     this.ComboBoxCourse.SelectedIndex = -1;
246
247     // Unset checkboxes if checked
248     if (chkFalse is CheckBox)
```

```
MainWindow.xaml.cs x
40401270 _40401270.MainWindow validateListCountry()
244
245 // Unset checkboxes if checked
246 if (chkFalse is CheckBox)
247 {
248     CheckBox nonInternationalStudentCheckBox = (CheckBox)chkFalse;
249     if (nonInternationalStudentCheckBox.IsChecked == true)
250     {
251         nonInternationalStudentCheckBox.IsChecked = false;
252     }
253 }
254 if (chkTrue is CheckBox)
255 {
256     CheckBox internationalStudentCheckBox = (CheckBox)chkTrue;
257     if (internationalStudentCheckBox.IsChecked == true)
258     {
259         //unset comboBox
260         internationalStudentCheckBox.IsChecked = false;
261     }
262 }
263
264 }
265
266
267 1 reference
268 private void internationalStudentCheckBoxSelected(object sender, RoutedEventArgs e)
269 { // Allow student to international student to select the country
270     comboBoxesValidation();
271     if (chkTrue is CheckBox)
272     {
273         CheckBox internationalStudentCheckBox = (CheckBox)chkTrue;
274         //Check if a student is an international student
275         if (internationalStudentCheckBox.IsChecked == true)
276         {
277             // Show the label and listbox for the selection of the country
278             lblCountry.Visibility = Visibility.Visible;
279             lblCountry.UpdateLayout();
280             ComboBoxCountry.Visibility = Visibility.Visible;
281             ComboBoxCountry.UpdateLayout();
282             string CityFileList = "CityList.txt";
283         }
284     }
285 }
```

```
MainWindow.xaml.cs
40401270
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");
}

1 reference
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();
}

2 references
private void comboBoxesValidation()
{
    // Check that just one of the comboboxes 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)
```

```
inWindow.xaml.cs
40401270
private void comboBoxesValidation()
{
    // Check that just one of the comboboxes 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 choice!");

                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() ");
    }
}
```