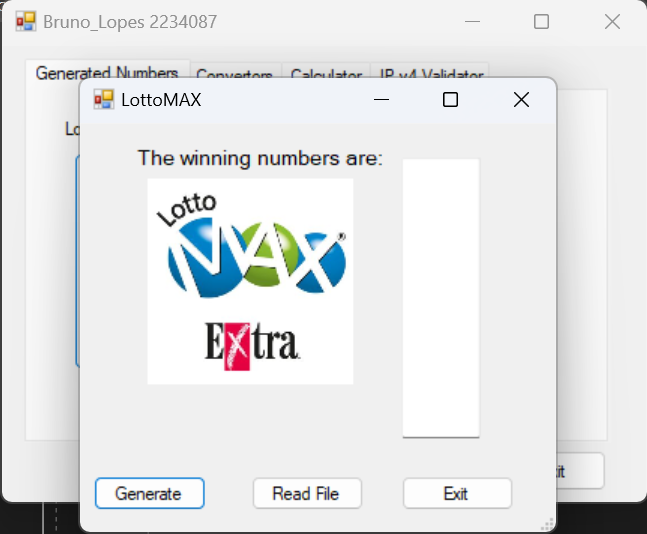
|  |
| --- |
| College LaSalle |
| Project - Oriented Object Programming User and Technical Manual |
|  |
| Presented to: Mihai Maftei. |

|  |
| --- |
| Your name: Bruno Rafael Ferreira Lopes  3/21/2023  Version: 1.2 |

1. **Start by adding a short description of your project, and the languages (technologies) used:**
2. Language: C#
3. Tools (IDE): Visual Studio 2022
4. **Present the print screens of yours forms and have a detailed description of the functionalities (step by step).**
5. If you click on button Lotto Max:

Graphical user interface, application, Word

Description automatically generated

1. If you click on Generate, the 8 unique random numbers will be generated in the textbox and in the text box, and 7 extra random numbers will be generated in the label below the figure:

Graphical user interface, application

Description automatically generated

1. …
2. If you click on the Exit button, …
3. **Present the code of your application (forms).**

**Code in the button Generate:**

private void button1\_Click(object sender, EventArgs e)

{

//creation of extra 7 numbers generated bellow the picture

Random random1 = new Random(); //constructor of the obj random1

int tempNumber;

string tempString = "";

int[] randomNumber1 = new int[7]; //creating an array with random numbers

for (int i = 0; i < 7; i++) {

tempNumber = random1.Next(1,9);

randomNumber1[i] = tempNumber; //variable randomNumber1 is equal to the obj random1 with a random number between 0 and 9

tempString += randomNumber1[i].ToString(); //variable randomNumber1 converted to string

}

label2.Text = tempString; //display the variable at label 2

tempString = ""; //clear the string for the next generation of numbers

//creation of the main 8 random numbers

Random random = new Random(); //constructor of the obj random

string tempLoto = "";

List<int> randomNumber = new List <int>(); //creating a list with random numbers

for (int i = 0; i < 7; i++)

{

int rand = random.Next(1,50); //generate the random number

if (!randomNumber.Contains(rand)) //if the random number is not in the list already

{

randomNumber.Add(rand); //add the random number in the list

tempLoto += randomNumber[i].ToString() + "\t"; //store the number in the variable that will be printed in the textbox

}

else

{

i = i - 1; //if the number generated already exists in the list, the counter back to try again

}

}

//bonus number

string tempBonus = "";

label1: int numBonus = random.Next(1, 50);

if (!randomNumber.Contains(numBonus)) //compare the bonus number with all other 7 numbers in the list

{

randomNumber.Add(numBonus);

tempBonus = numBonus.ToString();

}

else

{

goto label1;

}

textBox1.Text = tempLoto+tempBonus; //display the numbers at the textbox 1

string txtFile = tempLoto + $" Bonus {tempBonus}"; //line of text that will be printed in a txt file

tempLoto = ""; //clear the string for the next generation of numbers

tempBonus = "";

//save the numbers in a text file

StreamWriter txt = new StreamWriter("G:\\My Drive\\AULAS\\LASALLE\\Lasalle - Winter2023\\OOP\\FINAL PROJECT\\LottoNbrs.txt", true); //path where will be generated the text file

string print = txtFile.Replace("\t",","); //change the text in the txt file, replacing the tabs for ,

txt.Write("Max, "+DateTime.Now+" "+print+"\n"); //printing in the txt file

txt.Close();

txt.Dispose();

}

**Code in the button Read File:**

1. **Present the classes and/or methods that you create or you did use in the project.**

|  |  |
| --- | --- |
| **Class/Method Name** | **Description** |
| 1. Class1 | Enter a short description of the class or of the method ………………. |
| 1. Class2 | Enter a short description of the class or of the method ……………….. |
| 1. void Method(int total) | Enter a short description of the class or of the method ……….. |
|  |  |
| 1. Constructor() | Enter a short description of the class or of the method ………… |

1. **Present the difficulties that you have, what was the hardest and the easiest part of your project.**

* The generation of the unique random numbers was an issue, but I could handle it, putting the numbers in a list and using the method Contains(), to verify if the random number already exists in the previous numbers.