INF 154
PRACTICAL 4
2023

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA
Denkleiers • Leading Minds • Dikgopolo tša Dihlalefi

WARNING!!!

- After Practical 4, we will not be marking work done on student laptops, **only work downloaded via ClickUP** will be marked (We strongly advise you to re-watch the upload-videos on how to successfully upload homework assignments if you are still struggling with uploading your assignments)
- If you cannot attend the practical classes for your assignments to be marked, you should email us well in advance with a valid excuse, and after we evaluate, we can come up with a solution.
- In other words, ALWAYS TRY TO ATTEND THE PRACTICAL CLASSES if you want your assignments to be marked

At the end this practical you should be able…

1. To use if statements for conditional execution in your programs
2. To use nested if-statements and else-if statements
3. To write conditional statements using conditional operators such as ==,<,>,>=
4. To use logical operators &&, || and !
5. To do basic error handling

What are we going to do?

- Prac 4a: Mass Convertor(Completed in class)
- Prac 4b: Grade Verifier (Completed in class)
- Prac 4c: Take home practical.

Practical Exercise 4a

Objective: In this practical section, you are tasked with creating a windows form program which will convert ***kilograms*** to ***pounds*** or ***grams*** depending on what the user selected.

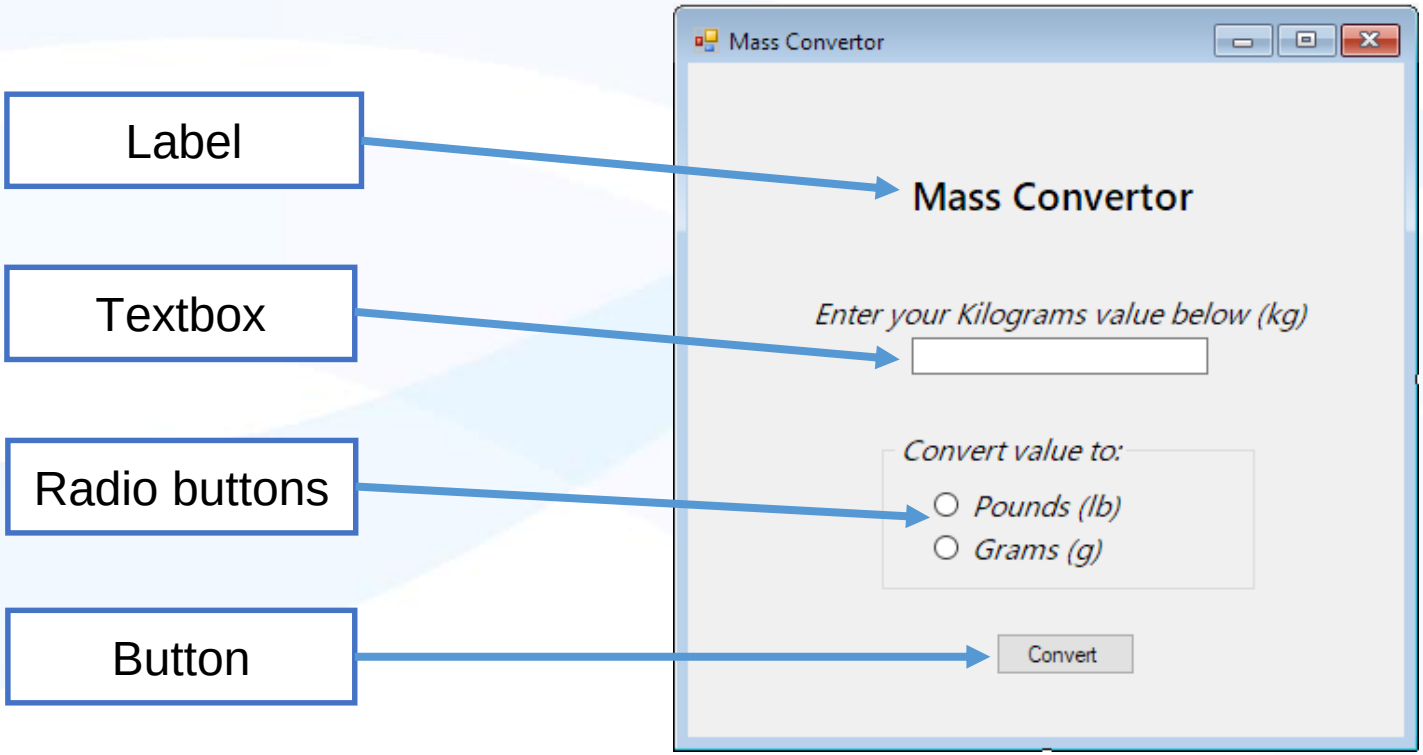
<u>The conversion rates are as follows</u>

- 1 kilogram is equal to 2.20462262 pounds
- 1 kilogram is equal to 1000 grams

Practical Exercise 4a

Expected Screen Design

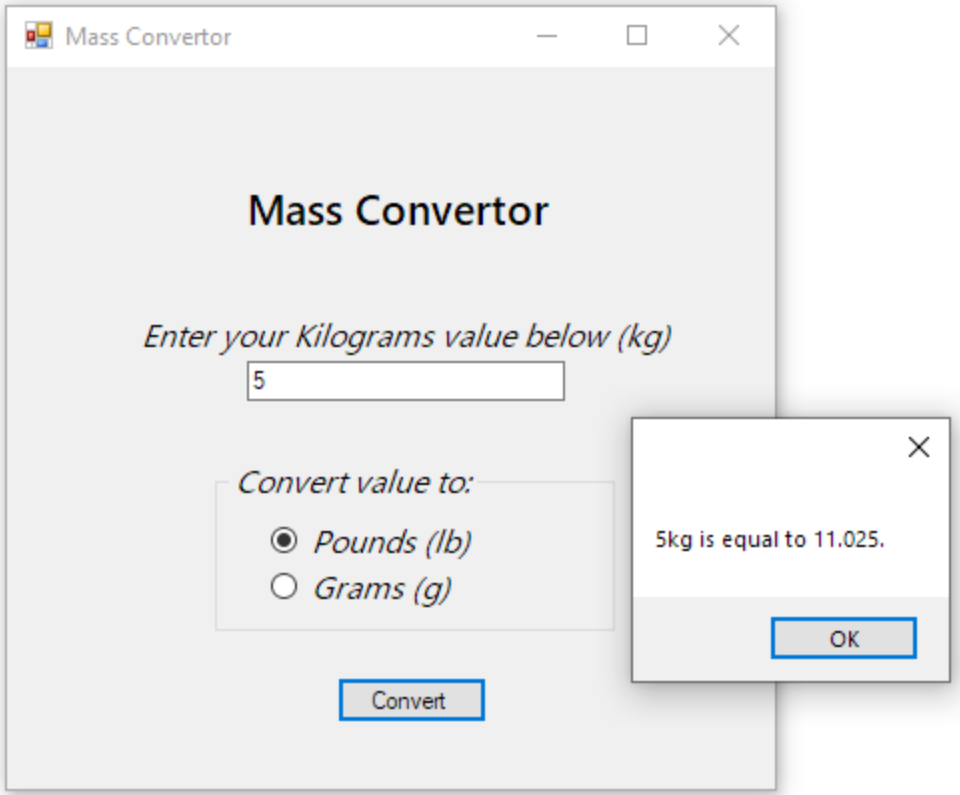
Create the following form with the appropriate controls. Remember to name your controls accordingly (e.g for a button you would use btnConvert).



Practical Exercise 4a

Expected Functionality

When the convert button is clicked and the pounds radio button is selected, A message box will display the result as follows.



Practical Exercise 4a

Solution

```
private void btnConvert_Click(object sender, EventArgs e)
{
    // variables
    double inputKG;
    const double POUNDS_CONVERSION_RATE = 2.205;
    const double GRAMS_CONVERSION_RATE = 1000;
    double output;

    // input
    inputKG = Convert.ToDouble(txtKilograms.Text);

    // process
    if (radPounds.Checked == true)
    { output = inputKG * POUNDS_CONVERSION_RATE;}
    else
    { output = inputKG * GRAMS_CONVERSION_RATE;}

    // output
    MessageBox.Show(inputKG + "kg is equal to " + output + ".");
}
```

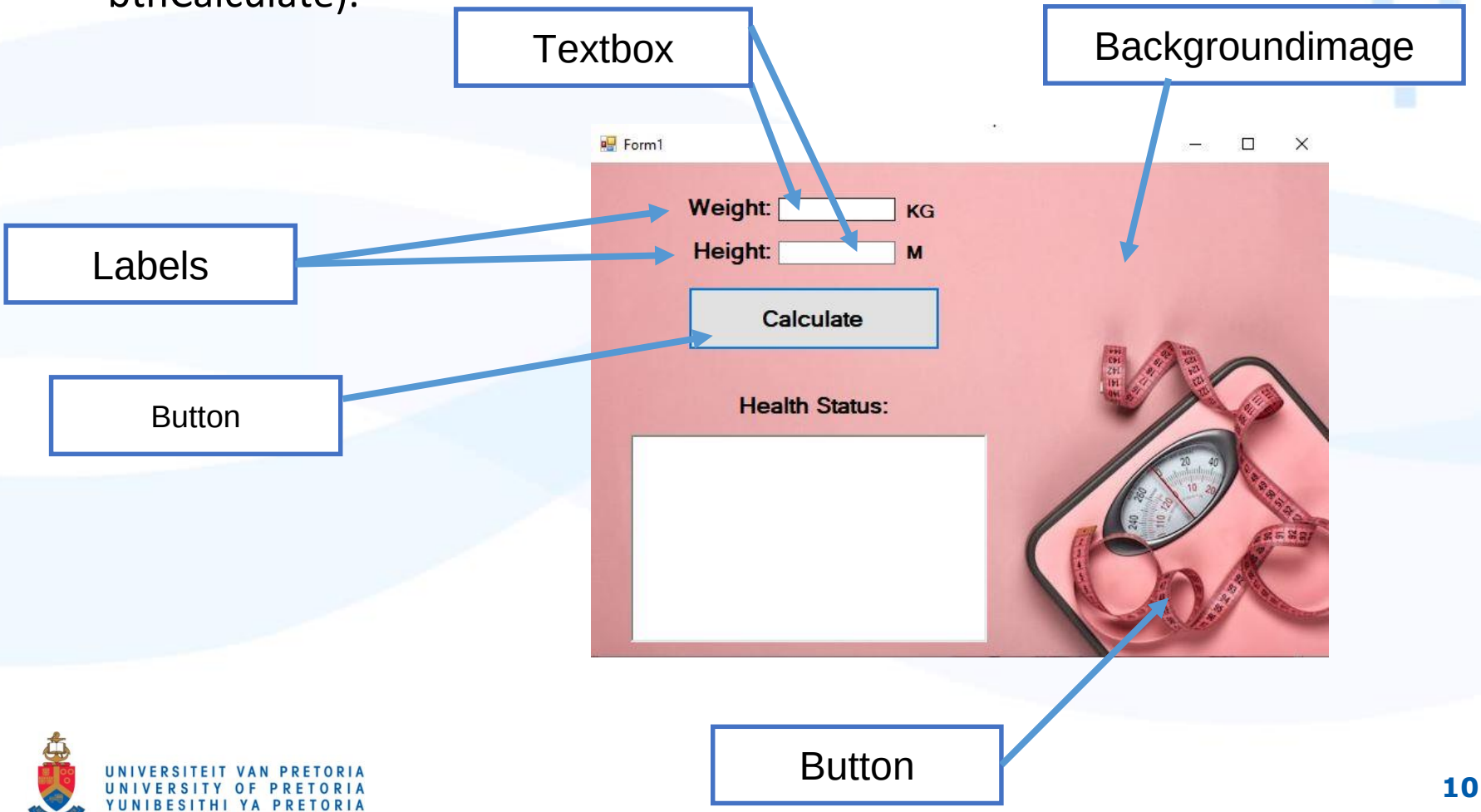
Practical Exercise 4b

Objective: In this practical section, you are tasked with creating a windows form program which will capture the BMI of the user. Upon capturing the two required inputs, the program should calculate the BMI amount and indicate whether the user is either underweight, normal, overweight or obese.

Practical Exercise 4b

Expected Screen Design

Create the following form with the appropriate controls. Remember to name your controls accordingly (e.g for the button you would use btnCalculate).



Practical Exercise 4b

BMI FORMULA:

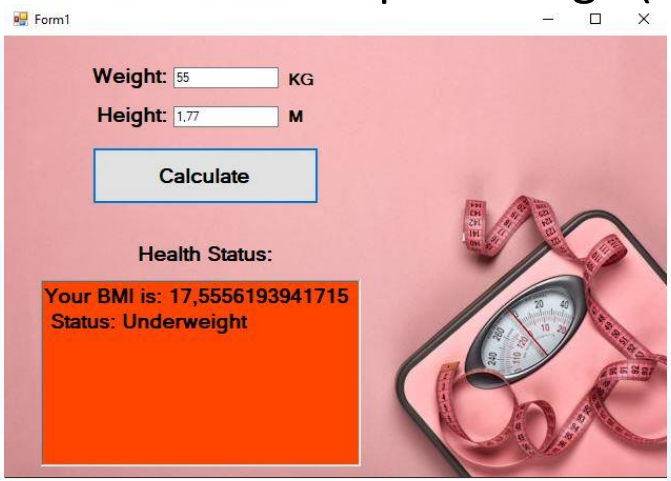
BMI = Weight / (Height) ^{2}

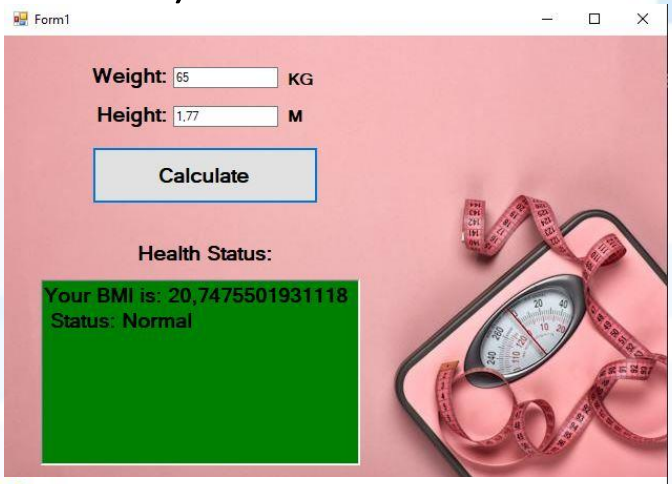
BMI	Status
≤ 18.4	Underweight
18.5 - 24.9	Normal
25.0 - 39.9	Overweight
≥ 40.0	Obese

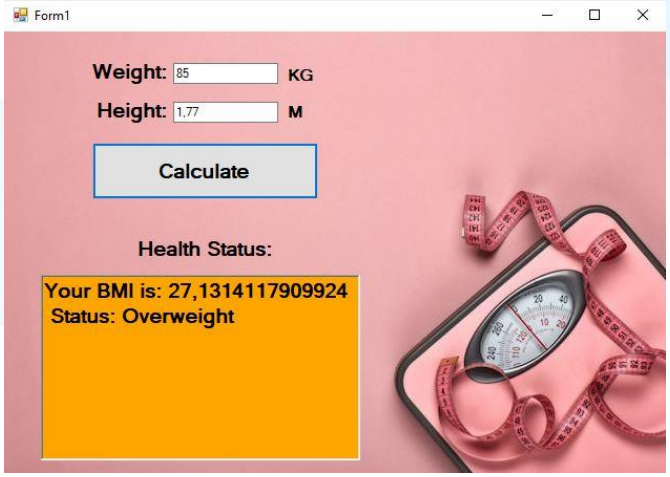
Practical Exercise 4b

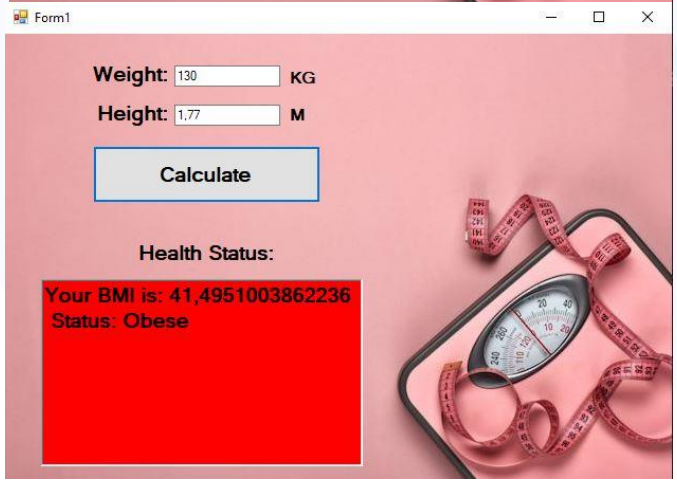
Expected Functionality

When the Calculate button is clicked, the BMI percentage should be displayed in the richtextbox and the color of the textbox should change based on the BMI percentage (indicating the status):









Practical Exercise 4b

Solution

```
1 reference
private void btnCalculate_Click(object sender, EventArgs e)
{
    double bmi, weight, height;

    weight = Convert.ToDouble(txtWeight.Text);
    height = Convert.ToDouble(txtHeight.Text);

    bmi = weight/ Math.Pow(height, 2);
```

Next slide for more code

Practical Exercise 4b

Solution

```
if (bmi <= 18.4)
{
    rtbOutput.BackColor = Color.OrangeRed;
    rtbOutput.Text = "Your BMI is: " + Convert.ToString(bmi) + "\n Status: Underweight";
}
else if (bmi >= 18.5 && bmi <= 24.9)
{
    rtbOutput.BackColor = Color.Green;
    rtbOutput.Text = "Your BMI is: " + Convert.ToString(bmi) + "\n Status: Normal";
}
else if (bmi >= 25.0 && bmi <= 39.9)
{
    rtbOutput.BackColor = Color.Orange;
    rtbOutput.Text = "Your BMI is: " + Convert.ToString(bmi) + "\n Status: Overweight";
}
else if (bmi >= 40)
{
    rtbOutput.BackColor = Color.Red;
    rtbOutput.Text = "Your BMI is: " + Convert.ToString(bmi) + "\n Status: Obese";
}
```

Practical Exercise 4b

What about NESTED decision statements?

Practical Exercise 4c

Recommended Controls for both sections:

Section 1 (Login section)

–3 labels

–2 textboxes

–2 buttons

Section 2 (Unit Calculation section)

–2 labels

–2 textboxes

–1 button

–1 richtextbox

Practical Exercise 4c

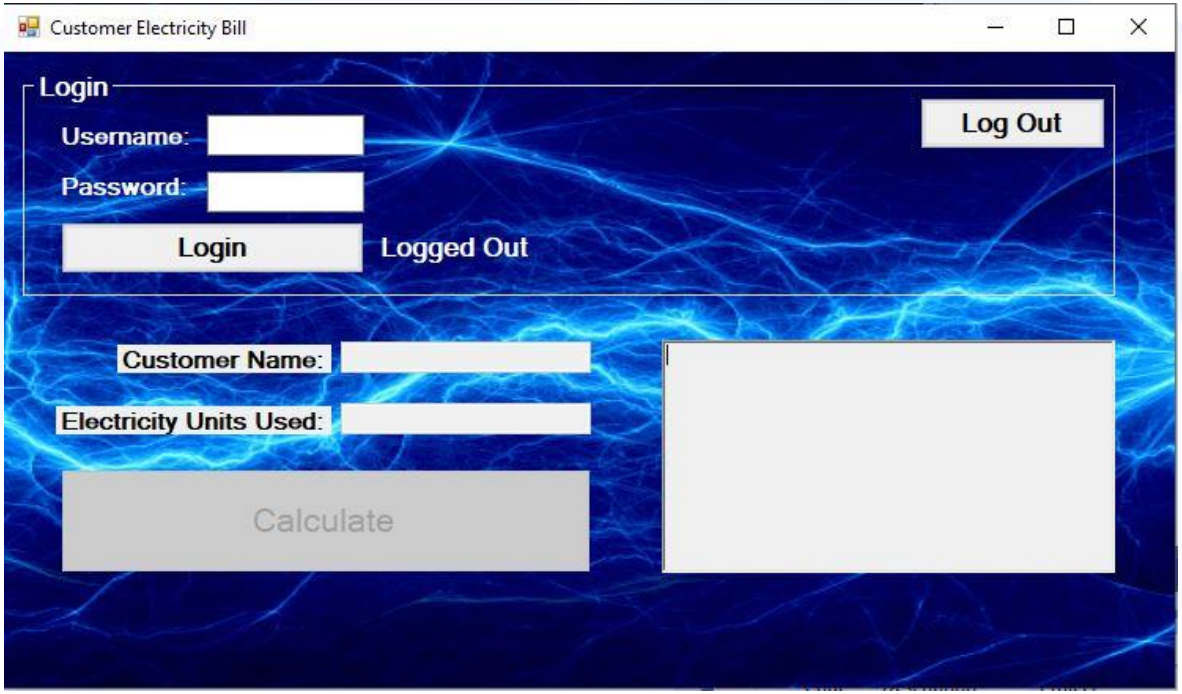
Login section

You will be tasked with the job of validating a user's login credentials before moving onto anything else.

The username and password has already been given as follows:
Username = user
Password = 1234

Hint: Make these variables/constants

The Expected form design:



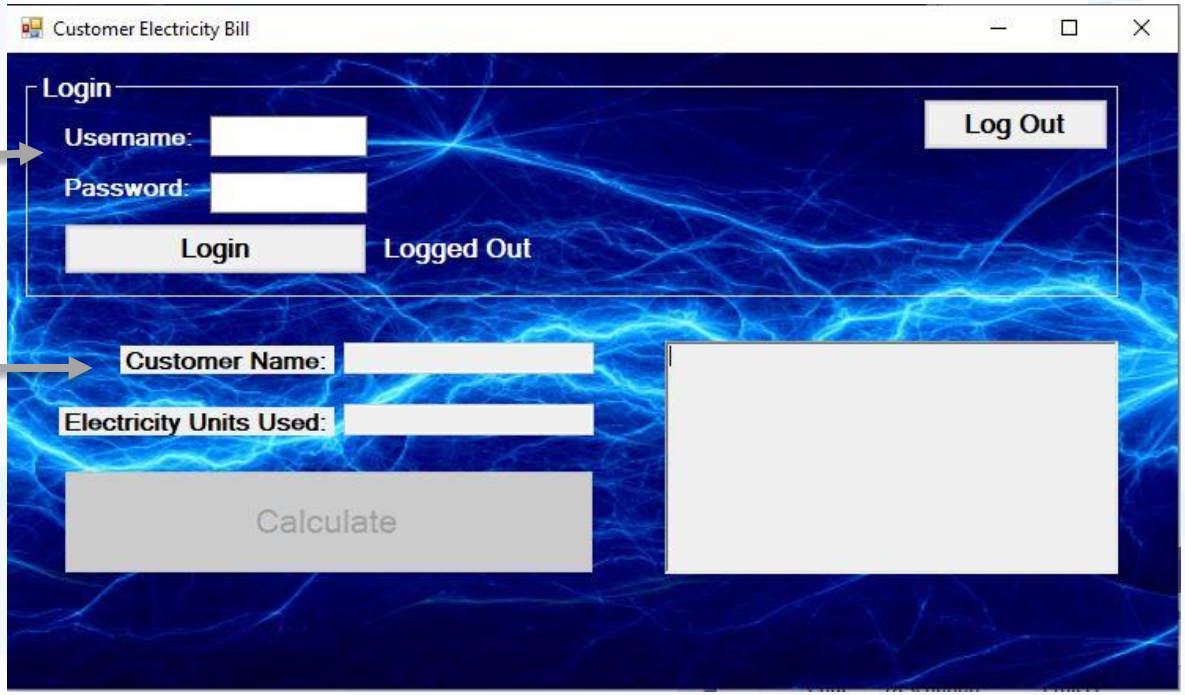
Practical Exercise 4c

Notice that the Calculate button (btnCalculate) as well as the textboxes (txtCustomerName and txtUnitesUsed) are disabled at default. This is because the user needs to first log in successfully before they can access and enter their details below in the unit calculation section.

The Expected form design:

Login Section

Unit Calculation Section



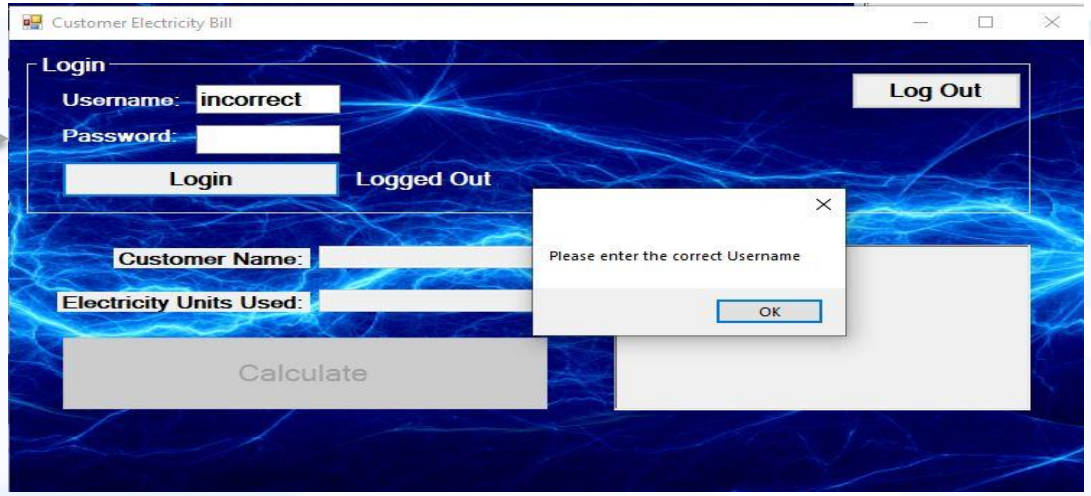
Hint: You will need to first validate the login credentials in order to enable the textboxes (txtCustomerName and txtUnitsUsed) and the calculate button (btnCalculate)

Practical Exercise 4c

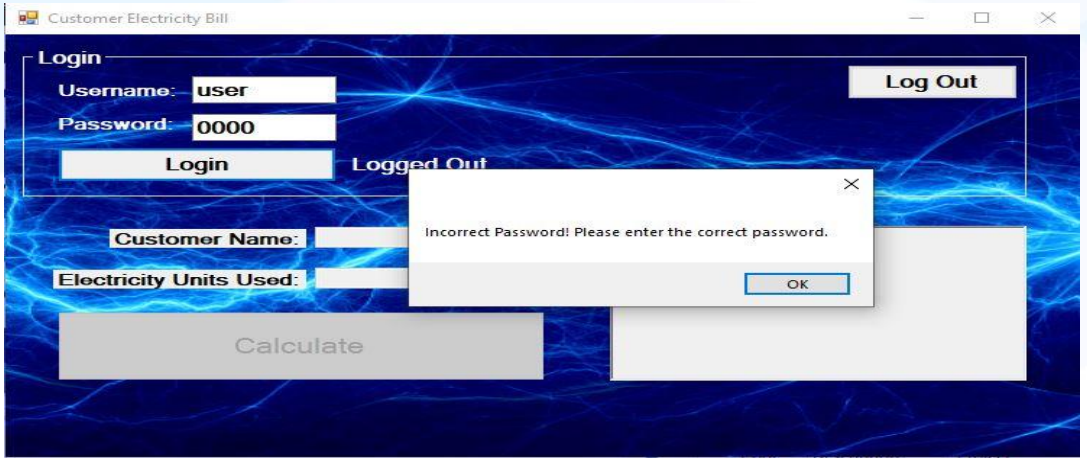
Validation messages via messageBoxes: Communicate your validation outcomes with the user !

The Expected form design:

If username is incorrect



If username is correct, but the password is incorrect

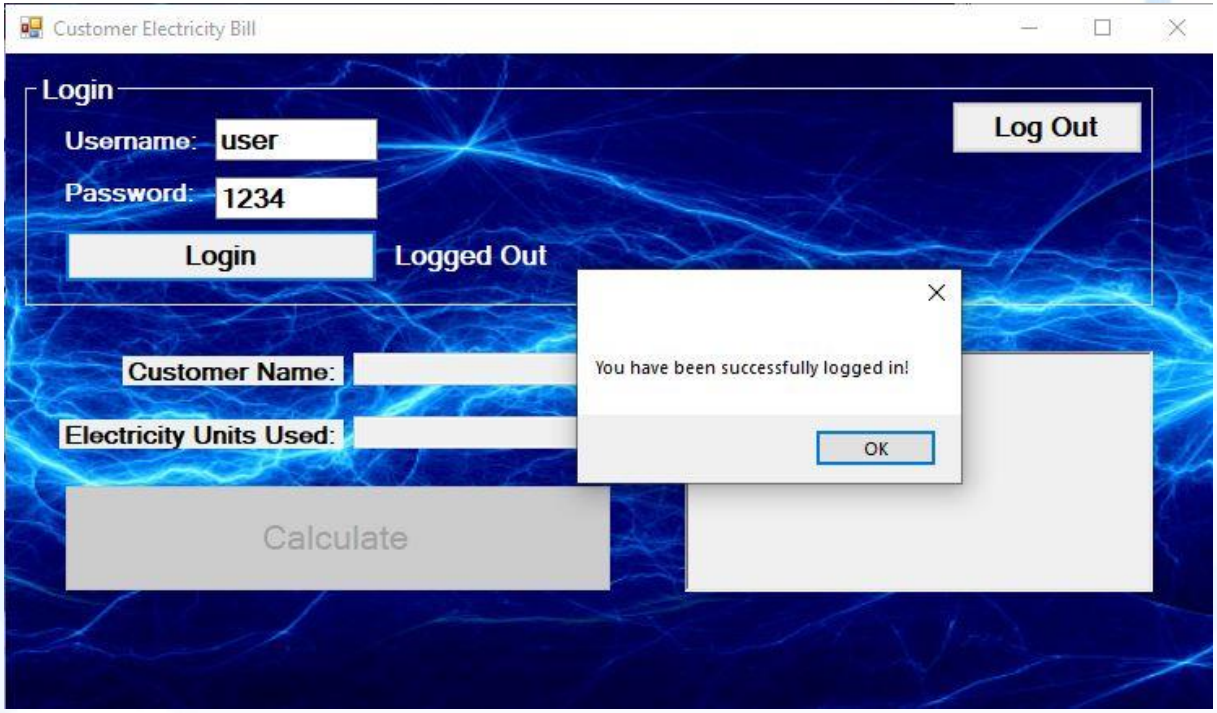


Practical Exercise 4c

Validation visuals via messageBoxes. Communicate your validations with the user.

The Expected form design:

If the username and password both match



All the necessary controls in the Unit Calculation section should now also be enabled.

Practical Exercise 4c

Unit Calculation section

Once you have successfully logged in, you will now need to calculate and display a customer's electricity bill. The Customer's Name, and Unit used should be displayed in a richtextbox along with the total bill amount that the customer must pay at the end of the month.

Charge rate for Units Used are as follows:

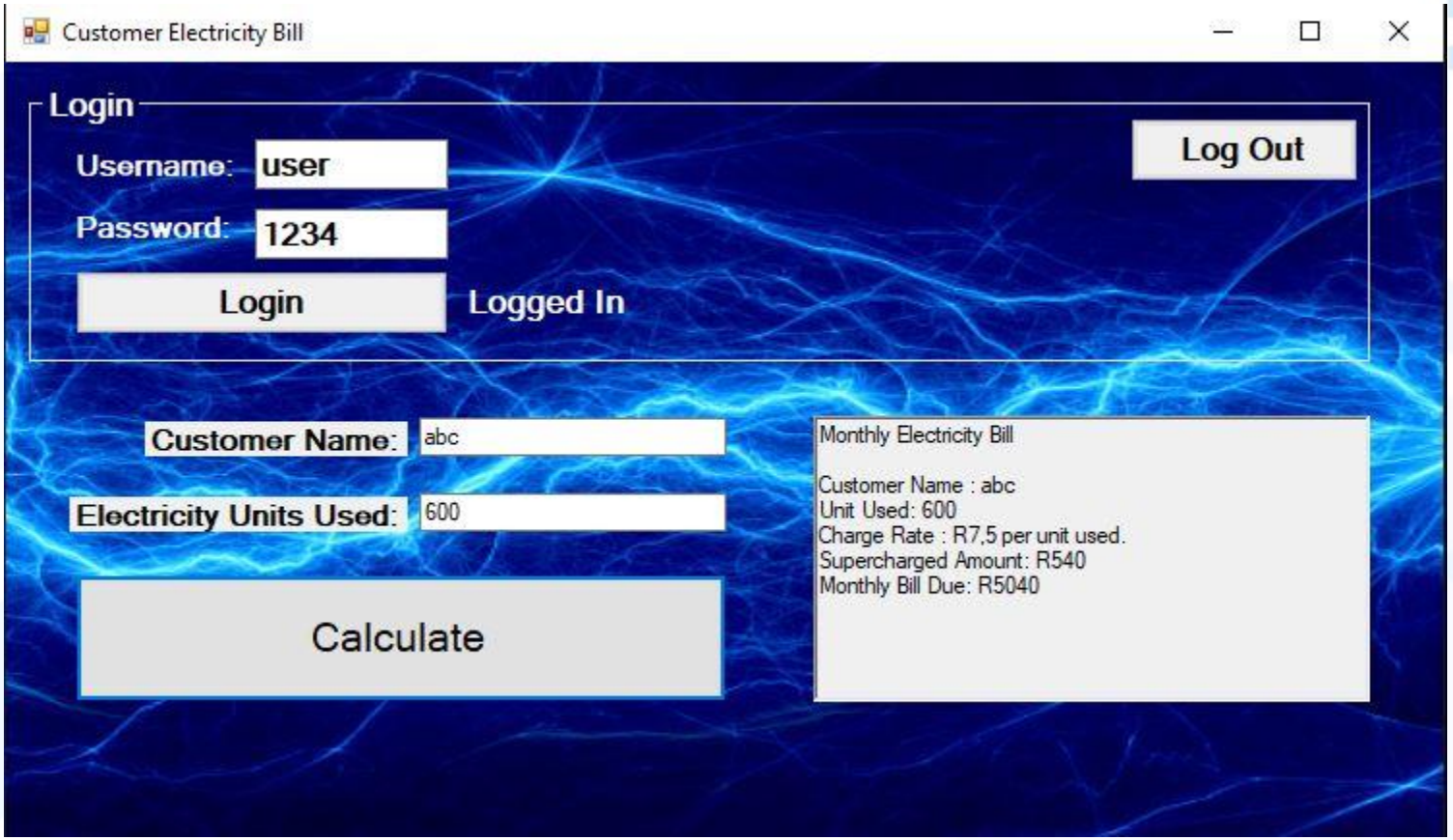
Units Used per Month	Charge Rate per Units Used
0 – 200	@ R2.50
201 – 400	@ R5.00
401 – 600	@ R7.50
601 and above	@ R10.00

If the total monthly bill exceeds R3000.00 then a supercharge of 12% will be added on.

For every customer, the minimum monthly bill should be R100.00.

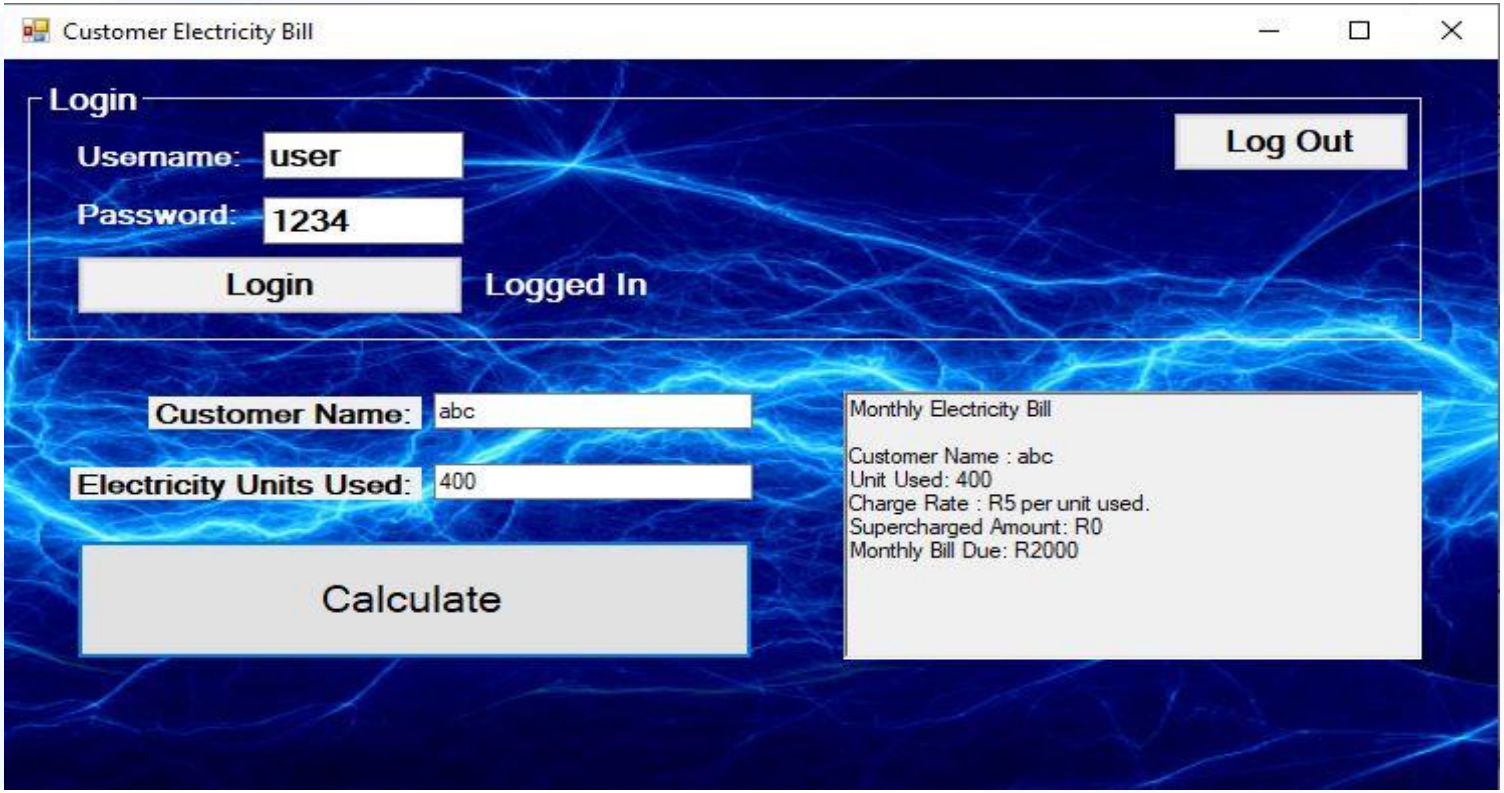
Practical Exercise 4c

Expected Output: When bill is above R3000.00 and super charge is applied



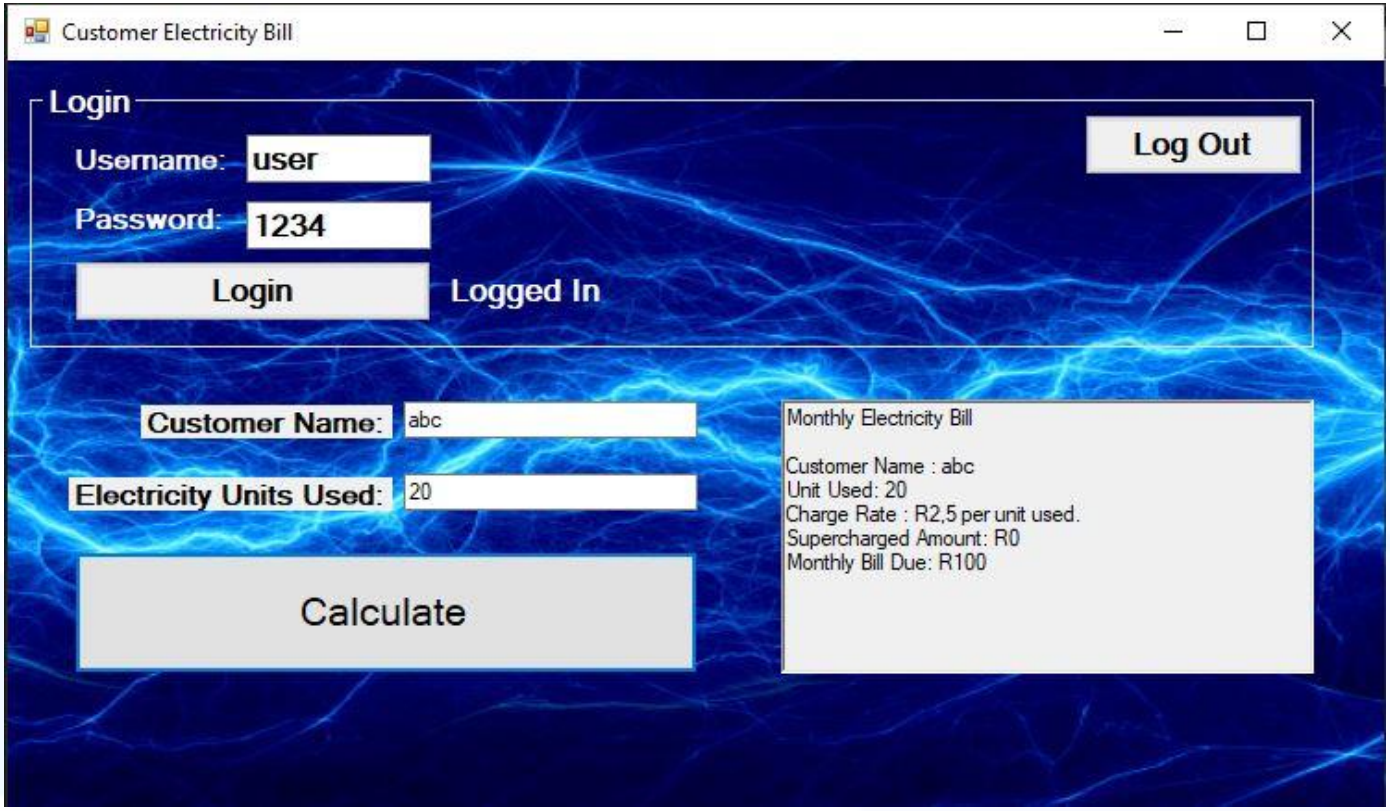
Practical Exercise 4c

Expected Output: When bill is below R3000.00 no super charge is applied



Practical Exercise 4c

Expected Output: Minimun possible bill will always be R100.00



Practical 4 Submission

Submit your Practical 3c project on ClickUP as follows:

–**Due Date: 03 April 2023.**

–Name your project, **INF154Practical4xxxxxxxx** (Where xxxxxxxx is your student number) and compress (zip folder) your project.

–Submit it under the Practical 4 submission link.