Paint Calculator Software Solution

(Borwell – Software Challenge)

Documentation

# Client’s Requirements

Write a program that takes as input the dimensions of a room and outputs the following:

* Area of the floor
* Amount of paint required to paint the walls
* Volume of the room

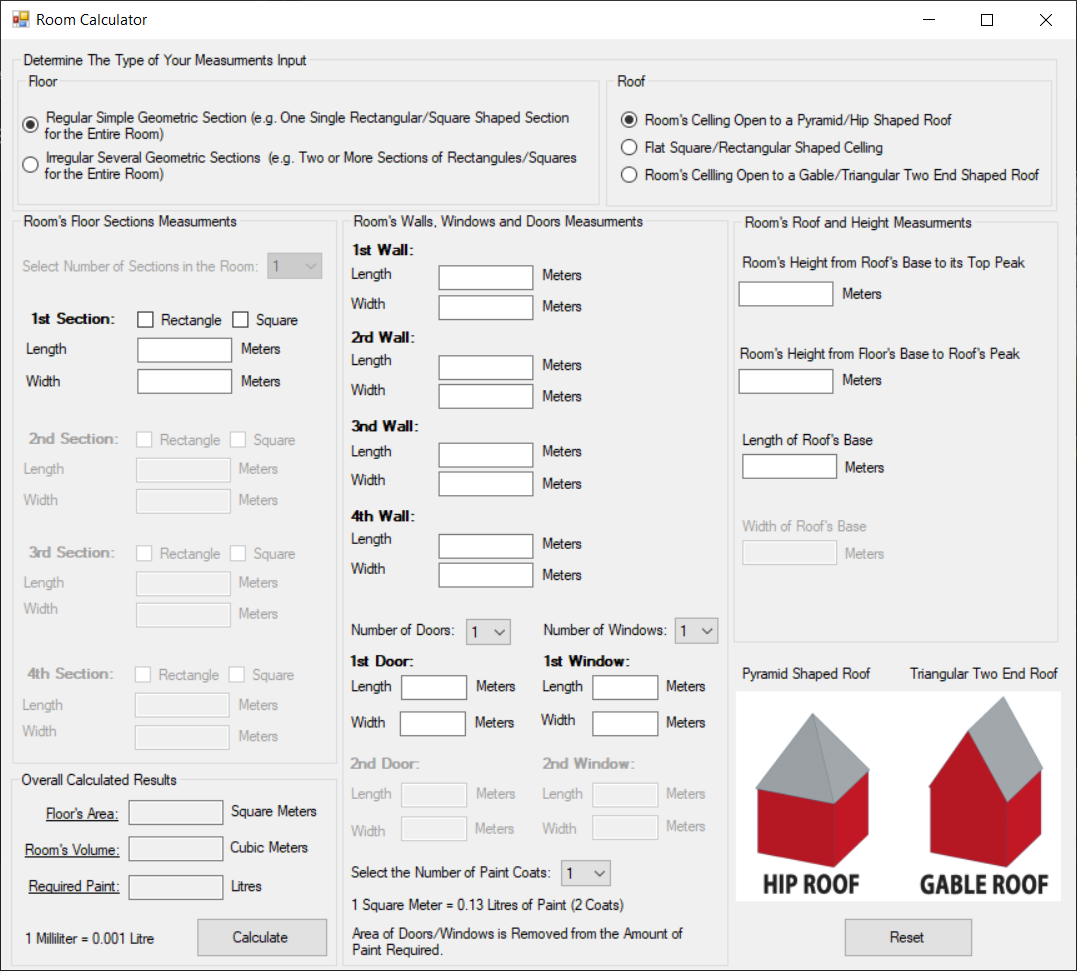
# Assumptions Made

* 1 square meter = 0.065 Litres of paint (1 Paint Coats)
* 1 square meter = 0.13 Litres of paint (2 Paint Coats)
* 1 square meter = 0.195 Litres of paint (3 Paint Coats)

# Limitations

* The user is allowed to input floor dimensions/measurements of only up to 4 different sections of the room’s floor which can either be rectangle or square shaped based on user’s selection.
* The user is limited to select only one of 3 types of a room’s roof/celling which can either be a flat celling, celling open to a pyramid shaped roof or celling open to a gable shaped roof.
* The user is restricted to input dimensions/measurements of only up to 4 walls.
* The user is able to input dimensions/measurements of only up to 2 windows and doors based on user’s choice.
* The user is permitted to select only up to a number of 3 paint coats for the walls.
* The program only supports meters as its main input measuring unit for inputting the dimensions/measurements of both the room’s floor, celling/roof and walls.

# Graphical User Interface (GUI) Design



chkRecSec1

chkSquSec1

txtWidSec3

txtLenSec2

txtArea

txtVolume

cboNumWindows

cboNumDoors

cboNumCoats

txtLenRoof

rdoPyr

rdoFla

rdoTri

txtHeightRoof

txtHeightRoom

txtWidRoof

btnReset

cboNumSecs

rdoReg

rdoIrr

txtWidDoor2

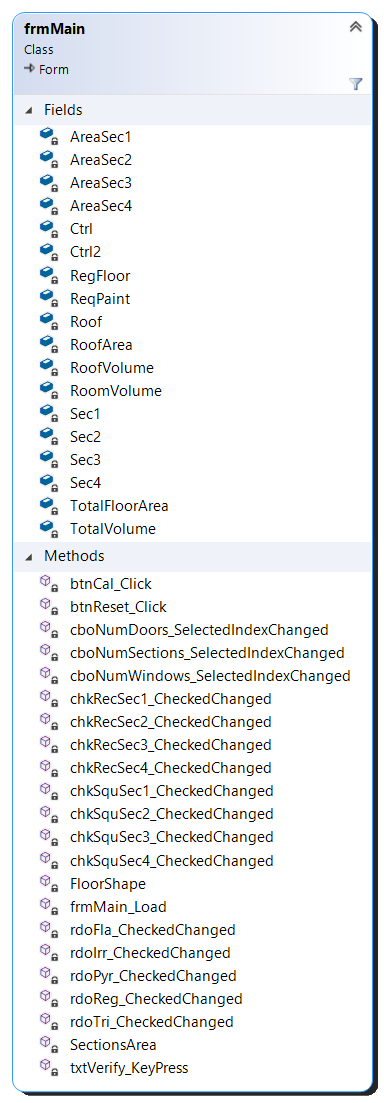
btnCal

txtLenWin1

txtWidWall4

txtReqPaint

# Class Diagram

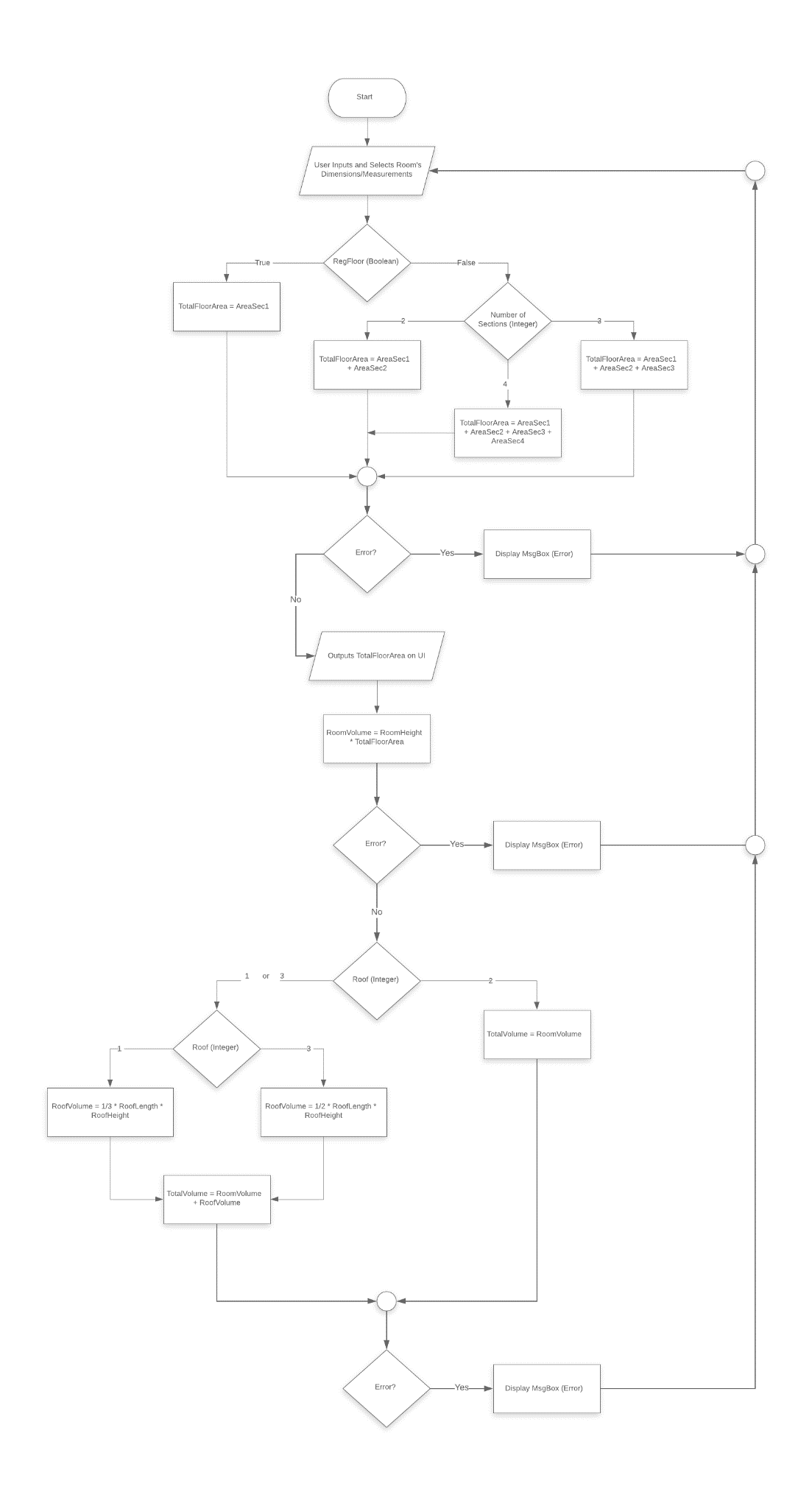


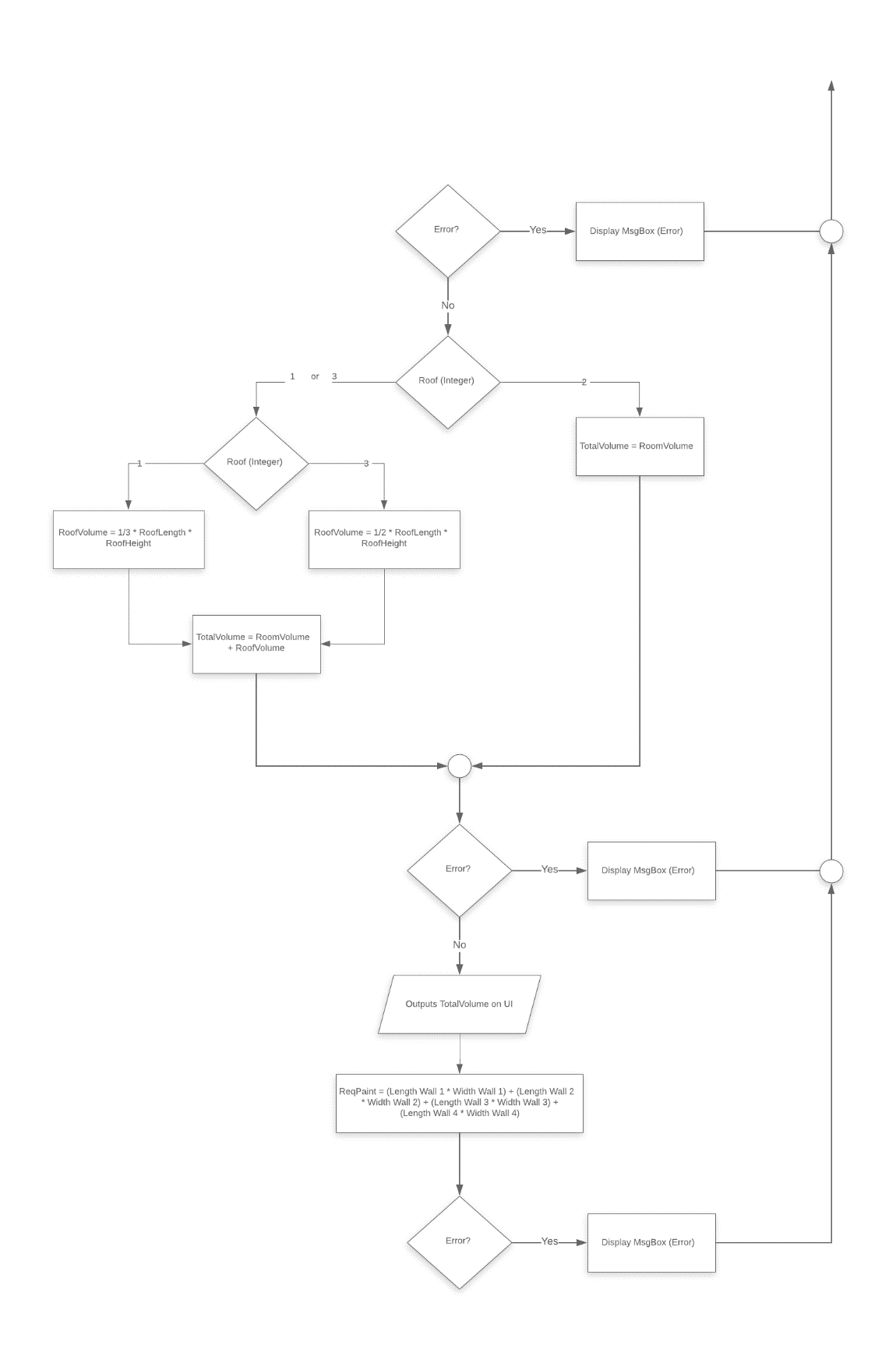
# Data Dictionary for both Fields/Methods

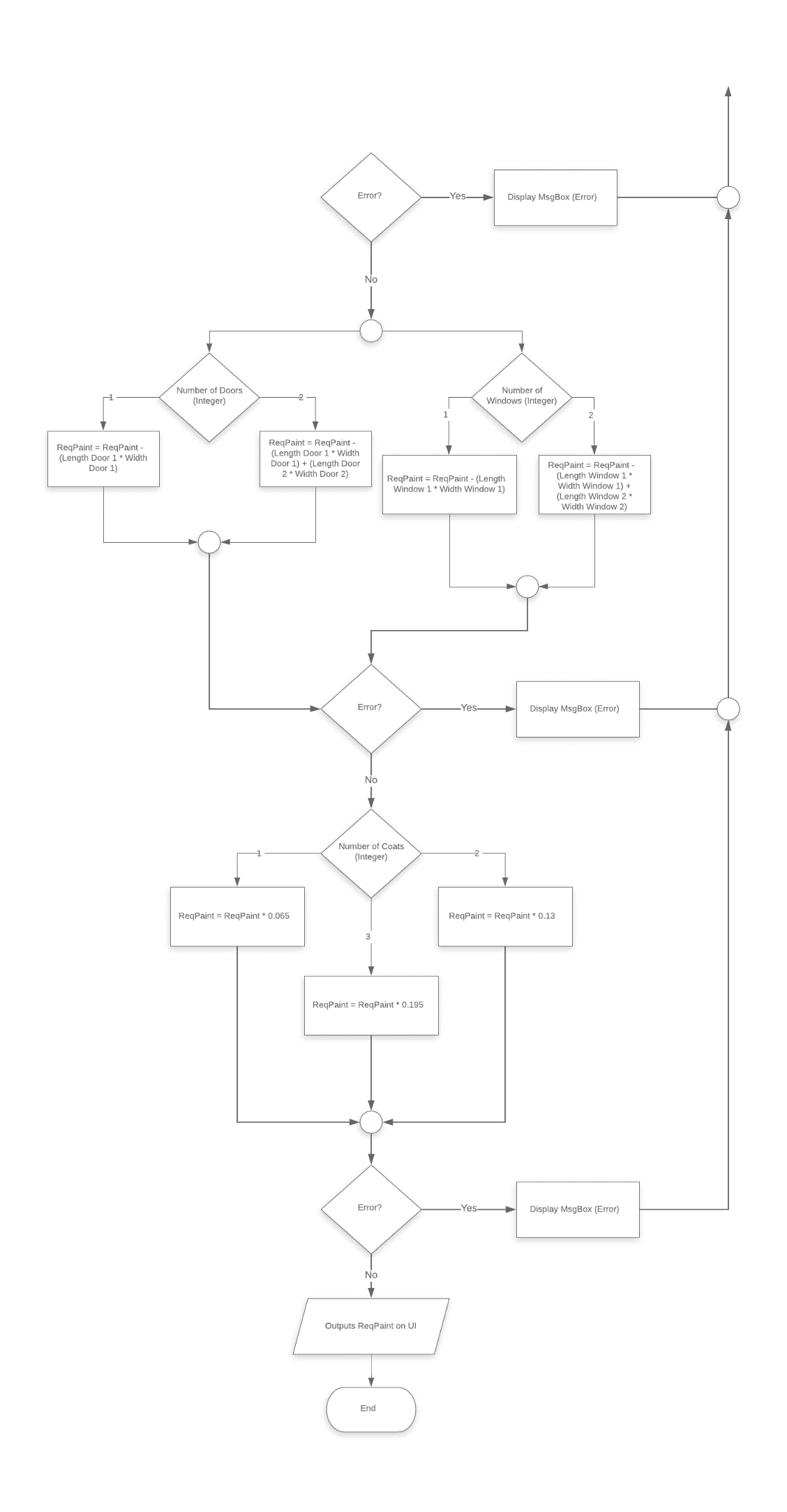
|  |  |  |  |
| --- | --- | --- | --- |
| **Fields** | **Scope** | **Data Type** | **Purpose** |
| AreaSec1 | Private | Decimal | Stores value of floor’s section 1 area. |
| AreaSec2 | Private | Decimal | Stores value of floor’s section 2 area. |
| AreaSec3 | Private | Decimal | Stores value of floor’s section 3 area. |
| AreaSec4 | Private | Decimal | Stores value of floor’s section 4 area. |
| Ctrl | Private | Control | Used as a placeholder to loop through several different UI controls via a for loop. |
| Ctrl2 | Private | Control | Used as a placeholder to loop through several different UI controls via a for loop. |
| RegFloor | Private | Boolean | Defines whether the floor is regularly shaped or not. (Default = True). |
| ReqPaint | Private | Decimal | Stores amount of paint required to paint the walls. |
| Roof | Private | Integer | Stores a value of 1, 2 or 3 to indicate the shape/type of room’s roof/celling. |
| RoofArea | Private | Decimal | Stores value of roof’s area. |
| RoofVolume | Private | Decimal | Stores value of roof’s volume. |
| RoomVolume | Private | Decimal | Stores value of room’s volume. |
| Sec1 | Private | Integer | Stores a value of either 1 or 2 to indicate whether section 1 of the floor as either being rectangle or square shaped. |
| Sec2 | Private | Integer | Stores a value of either 1 or 2 to indicate whether section 2 of the floor as either being rectangle or square shaped. |
| Sec3 | Private | Integer | Stores a value of either 1 or 2 to indicate whether section 3 of the floor as either being rectangle or square shaped. |
| Sec4 | Private | Integer | Stores a value of either 1 or 2 to indicate whether section 4 of the floor as either being rectangle or square shaped. |
| TotalFloorArea | Private | Decimal | Stores value of floor’s total area. |
| TotalVolume | Private | Decimal | Stores value of room’s total volume. |
| txtbox | Local to both btnReset\_Click() & txtverify\_KeyPress() | TextBox | Used as a placeholder to loop through all textbox controls via a for loop. |
| chkbox | Local to btnReset\_Click() | CheckBox | Used as a placeholder to loop through all checkbox controls via a for loop. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Methods** | **Scope** | **Type** | **Description** |
| btnCal() | Private Sub | Click Event | Processes all user inputs to decide how the results are calculated and outputted to the user depending on the user’s inputs and selections. |
| btnReset() | Private Sub | Click Event | Resets and clears the content of all controls on UI back to their default values (blank). |
| cboNumDoors() | Private Sub | Selected Index Change Event | Allows user to select a value for the number of doors in the room which is then used later on for processing the total area of the room’s walls by deducting the area of all doors from the total area of walls. |
| cboNumSections() | Private Sub | Selected Index Change Event | Allows user to elect a value for the number of sections in the room’s floor to enable corresponding UI controls based on user’s selection and disables irrelevant UI controls. |
| cboNumWindows() | Private Sub | Selected Index Change Event | Allows user to select a value for the number of windows in the room which is then used later on for processing the total area of the room’s walls by deducting the area of all windows from the total area of walls. |
| chkRecSec1() | Private Sub | Checked Changed Event | Calls FloorShape(). Sets Sec1 value to 1 to identify that section 1 of the floor is rectangle shaped. |
| chkRecSec2() | Private Sub | Checked Changed Event | Calls FloorShape(). Sets Sec2 value to 1 to identify that section 2 of the floor is rectangle shaped. |
| chkRecSec3() | Private Sub | Checked Changed Event | Calls FloorShape(). Sets Sec3 value to 1 to identify that section 3 of the floor is rectangle shaped. |
| chkRecSec4() | Private Sub | Checked Changed Event | Calls FloorShape(). Sets Sec4 value to 1 to identify that section 4 of the floor is rectangle shaped. |
| chkSquSec1() | Private Sub | Checked Changed Event | Calls FloorShape(). Sets Sec1 value to 2 to identify that section 1 of the floor is square shaped. |
| chkSquSec2() | Private Sub | Checked Changed Event | Calls FloorShape(). Sets Sec2 value to 2 to identify that section 2 of the floor is square shaped. |
| chkSquSec3() | Private Sub | Checked Changed Event | Calls FloorShape(). Sets Sec3 value to 2 to identify that section 3 of the floor is square shaped. |
| chkSquSec4() | Private Sub | Checked Changed Event | Calls FloorShape(). Sets Sec4 value to 2 to identify that section 4 of the floor is square shaped. |
| FloorShape() | Private Sub | Sub Procedure | A sub procedure that takes two attributes and is used to, for example, if attribute one which is checkbox A is checked then the second attribute which is checkbox B will be unchecked and vice versa if both the attributes are reversed in order. |
| frmMain() | Private Sub | Form Load Event | Sets out the initial and default values for certain variables and the status of particular controls on the UI. |
| rdoFla() | Private Sub | Checked Changed Event | Sets Roof’s variable value to 2. Also, disables irrelevant controls on the UI and enables corresponding relevant UI controls relating to a flat shaped celling. |
| rdoPyr() | Private Sub | Checked Changed Event | Sets Roof’s variable value to 1. Also, disables irrelevant controls on the UI and enables corresponding relevant UI controls relating to a pyramid shaped roof. |
| rdoTri() | Private Sub | Checked Changed Event | Sets Roof’s variable value to 3. Also, disables irrelevant controls on the UI and enables corresponding relevant UI controls relating to a triangular type of roof. |
| rdoReg() | Private Sub | Checked Changed Event | Sets RegFloor’s Boolean value to true. Also, disables irrelevant controls on the UI and enables corresponding relevant UI controls relating to a regular type of floor. |
| rdoIrr() | Private Sub | Checked Changed Event | Sets RegFloor’s Boolean value to false. Also, disables irrelevant controls on the UI and enables corresponding relevant UI controls relating to an irregular type of floor. |
| SectionsArea() | Private Sub | Sub Procedure | A sub procedure that takes two attributes which are either 1 or 2 to signify the type/shape of a floor’s section. This procedure determines and processes how the area of each section of the floor is calculated based on one of two formulas which depend on the user’s selection of type of each section in the floor. It is #called when btnCal is clicked upon by the user. |
| txtVerify() | Private Sub | Key Press Event | This is linked to all textbox controls on the UI to enforce a rule that is the user is only able to input numbers 0-9 and a decimal point (.) disallowing the input of any other letters or symbols. |

# Flowchart (Three Continuously Connected Pages)







# Assessment Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test ID** | **Test Description** | **Anticipated Results** | **Actual Results** | **Test Status** |
| 1 | User selects either a regular shaped floor or an irregular shaped floor to input dimensions and measurements of different sections in the room’s floor. The user must try to select both selections at a time and input all 4 sections of the room’s floor by also choosing whether each section is a rectangle or a square. The user at the end must click on the calculate button to output the area of the floor which must be correctly calculated and tested using another calculator as well to make sure they are the correct results for the corresponding entered numbers and measurements. | The user must notice 2 selection choices for the type of room’s floor. The regular type will disable the selection that allows the user to select the number of different sections on the floor and the irregular type of floor allows the user to select how many sections are on the floor. Each section can either be selected as a square or a rectangle and corresponding input methods will be prompted to the user for inputting the dimensions of each section in the room’s floor in meters. | The user had only one floor type selection which is regular and is only able to input dimensions of 1 section in the room’s floor due to the lack of UI controls to select more than one section of the floor and also the user couldn’t select any other type of room’s floor other than a regular type of floor which is not what is expected, in addition, the user couldn’t specify what the shape of the single section of the room’s floor was to be either a square or a rectangle because those UI controls were not available and missing. | Fail |
| 2 | User selects a different type of roof/celling in the room one at a time amongst three choices and inputs all their corresponding measurements and dimensions that are required to be inputted to allow for a new output in the volume of the room based on the measurements and dimensions inputted by the user after inputting both the dimensions of the 4 walls in the room and the different sections of the room’s floor. | The user must be able to find three roof/celling selections which can be either gable roof, flat celling or pyramid shaped roof. The user must select each one at a time and the program will prompt the user with the input methods corresponding to the selected choice of the user so that the user can input the dimensions required in meters such as the length of the roof/celling’s base or room’s height. | The user was unable to find the three selections of various room’s celling/roof types due to them being missing from the UI of the program. The user was only able to select a flat room celling and there were no input methods for other corresponding roof/celling types such as the height of the room or length of the roof/celling’s base which is not what is anticipated. | Fail |
| 3 | User selects a different number for both the doors and windows present in the room and input their dimensions to click on the calculate button to see the amount of paint required to paint the walls change depending on different numbers and measurements inputted by the user after inputting the dimensions of the 4 walls in the room. | The user should be able to input the dimensions of 4 walls of the room such as their length and width. The user afterwards must be able to select a different number for each one of the windows and doors present in the room and input their dimensions based on what the user had selected. For example, if 2 windows and 1 door was present then the user can input the corresponding dimensions and measurements accordingly. | The user could input fully the dimensions of 4 of the room’s walls, but couldn’t find any selections or choices to select from for both the windows and doors present in the room, let alone inputting their dimensions and measurements because those UI controls were not available nor yet introduced to the program’s interface which is not what is anticipated. | Fail |

# Resolving Issues relating to Assessment’s Negative Results

## Test 1

New UI controls have been introduced into the program to allow the users to select one of either two choices for the type of room’s floor which can be either a regular shape with a single corresponding section of the floor so that the user can input the dimensions of this single section of the floor, or an irregular shaped floor with different sections in the room’s floor up to a maximum of 4 sections allowing the user to input all dimensions and measurements of all 4 sections in the floor, in addition to, being able to select the type of section whether it is square shaped or rectangular shaped for each and every single section in the room’s floor which can impact the area of the floor overall with all those new selections and input methods introduced for the user to utilise.

## Test 2

New UI controls have been introduced into the program to allow users to select one of either three types of room’s celling/roof which can either be as a gable roof/celling, pyramid/hip shaped roof or a simple flat celling for the room. Each one of the selections will enable the program to prompt the users with different corresponding input methods so that the user can input the required dimensions and measurements which can impact the overall volume of the room because the user can now choose between three different types of roof/celling present in the room and input for example, the length of the roof’s base and height.

## Test 3

New UI controls have been introduced into the program to allow users to select a different number of doors and windows that can be present in the room and with each selection, corresponding input textboxes are prompted to the user allowing inputting the dimensions and measurements of each door and window present in the room. This will impact the overall amount of paint required to paint the walls which will be outputted at the end taking into consideration about the measurements inputted by the user for each one of the 4 walls of the room because the calculated area of both the windows and doors present will be deducted from the calculated area of the 4 walls of the room. This to make sure that the user can have a much more accurate measurements of the amount of paint required to paint the 4 walls of the room because the areas of the windows and doors are not included.

# Suggested Forthcoming Improvements

The developed program can be further improved by the addition of the following functionalities:

* Supports inputting, processing and calculating values of different metric units for lengths other than just meters, such as, centimetres and feet.
* Supports different various types of roofs other than the present ones, such as cross gabled, cross hipped, gambrel and others.
* Supports a wider variety of paint brands with different prices and qualities so that the user can select from a list and this would impact the final costs.
* Supports a larger number of walls or windows/doors present in the room more than 4 for the walls and more than 2 for both the windows/doors and the number of sections present in the room’s floor also more than just 4.

# Source Code (Written in VB.NET)

Public Class frmMain

Private TotalFloorArea As Decimal 'For a regular floor, (TotalFloorArea = AreaSec1), for an irregular floor, (AreaSec1 + AreaSec2 + AreaSec3 + AreaSec4)

Private RoofArea As Decimal

Private RoomVolume As Decimal

Private RoofVolume As Decimal

Private TotalVolume As Decimal

Private ReqPaint As Decimal

Private RegFloor As Boolean 'Stores floor's type regular (true) or not (false)

Private Sec1 As Integer 'Stores shape of Sec1 of the floor either as a rectangle = (1) or square = (2)

Private Sec2 As Integer

Private Sec3 As Integer

Private Sec4 As Integer

Private AreaSec1 As Decimal

Private AreaSec2 As Decimal

Private AreaSec3 As Decimal

Private AreaSec4 As Decimal

Private Roof As Integer 'Stores selected type of roof/celling (1, 2 or 3)

Private Ctrl As Control 'Used to hide/show specific UI controls

Private Ctrl2 As Control

'Sets initial and default starting values

Private Sub frmMain\_Load(sender As Object, e As EventArgs) Handles MyBase.Load

RegFloor = True

rdoReg.Checked = True

rdoPyr.Checked = True

Roof = 1

cboNumSecs.SelectedIndex = 0

cboNumDoors.SelectedIndex = 0

cboNumWindows.SelectedIndex = 0

cboNumCoats.SelectedIndex = 0

For Each Ctrl In grpRoom.Controls

If Ctrl.Tag = "Door2" Or Ctrl.Tag = "Window2" Then

Ctrl.Enabled = False

End If

Next

End Sub

'Sets type of floor to regular to disable irrelavent UI controls

Private Sub rdoReg\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoReg.CheckedChanged

RegFloor = True

For Each Ctrl In grpFloor.Controls

If Ctrl.Tag = "Irreg" Or Ctrl.Tag = "Irreg2" Or Ctrl.Tag = "Irreg3" Or Ctrl.Tag = "Irreg4" Then

Ctrl.Enabled = False

End If

Next

End Sub

'Sets type of floor to not regular to enable corresponding UI controls

Private Sub rdoIrr\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoIrr.CheckedChanged

RegFloor = False

For Each Ctrl In grpFloor.Controls

If Ctrl.Tag = "Irreg" Then

Ctrl.Enabled = True

End If

Next

End Sub

'Used to detect floor's shape to enable corresponding UI controls and disable irrelavent UI controls depending on user's inputted selection/choice

Private Sub FloorShape(rec, squ)

If rec.checked = True Then

squ.enabled = False

squ.checked = False

ElseIf rec.checked = False Then

squ.enabled = True

End If

End Sub

'Sets selected roof/celling type to enable UI corresponding controls and disable irrelavent UI controls

Private Sub rdoPyr\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoPyr.CheckedChanged

If rdoPyr.Checked = True Then

Roof = 1

lblWidRoof.Enabled = False

txtWidRoof.Enabled = False

lblWidRoofM.Enabled = False

Else

lblWidRoof.Enabled = True

txtWidRoof.Enabled = True

lblWidRoofM.Enabled = True

End If

End Sub

'Sets selected roof/celling type to enable UI corresponding controls and disable irrelavent UI controls

Private Sub rdoFla\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoFla.CheckedChanged

If rdoFla.Checked = True Then

Roof = 2

For Each Ctrl In grpRoof.Controls

If Ctrl.Tag = "Roof" Then

Ctrl.Enabled = False

End If

Next

lblHeightRoom.Enabled = True

lblHeightRoofM.Enabled = True

txtHeightRoom.Enabled = True

Else

For Each Ctrl In grpRoof.Controls

If Ctrl.Tag = "Roof" Then

Ctrl.Enabled = True

End If

Next

End If

End Sub

'Sets selected roof/celling type to enable UI corresponding controls and disable irrelavent UI controls

Private Sub rdoTri\_CheckedChanged(sender As Object, e As EventArgs) Handles rdoTri.CheckedChanged

If rdoTri.Checked = True Then

Roof = 3

lblLenRoof.Text = "Shortest Length of Roof's Base"

lblWidRoof.Text = "Longest Width of Roof's Base"

Else

lblLenRoof.Text = "Length of Roof's Base"

lblWidRoof.Text = "Width of Roof's Base"

End If

End Sub

'Allows to enable relavent UI controls and disables irrelavent UI controls depending on the selected value by the user

Private Sub cboNumSections\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cboNumSecs.SelectedIndexChanged

If cboNumSecs.SelectedItem = 1 Then

For Each Ctrl In grpFloor.Controls

If Ctrl.Tag = "Irreg2" Or Ctrl.Tag = "Irreg3" Or Ctrl.Tag = "Irreg4" Then

Ctrl.Enabled = False

End If

Next

ElseIf cboNumSecs.SelectedItem = 2 Then

For Each Ctrl In grpFloor.Controls

If Ctrl.Tag = "Irreg2" Then

Ctrl.Enabled = True

For Each Ctrl2 In grpFloor.Controls

If Ctrl2.Tag = "Irreg3" Or Ctrl2.Tag = "Irreg4" Then

Ctrl2.Enabled = False

End If

Next

End If

Next

ElseIf cboNumSecs.SelectedItem = 3 Then

For Each Ctrl In grpFloor.Controls

If Ctrl.Tag = "Irreg3" Or Ctrl.Tag = "Irreg2" Then

Ctrl.Enabled = True

For Each Ctrl2 In grpFloor.Controls

If Ctrl2.Tag = "Irreg4" Then

Ctrl2.Enabled = False

End If

Next

End If

Next

ElseIf cboNumSecs.SelectedItem = 4 Then

For Each Ctrl In grpFloor.Controls

If Ctrl.Tag = "Irreg4" Or Ctrl.Tag = "Irreg3" Or Ctrl.Tag = "Irreg2" Then

Ctrl.Enabled = True

End If

Next

End If

End Sub

'Sets shape of Sec1 of the floor to a square (2) and disables the rectangle check box and its corresponding UI controls if square check box is still checked

Private Sub chkSquSec1\_CheckedChanged(sender As Object, e As EventArgs) Handles chkSquSec1.CheckedChanged

FloorShape(chkSquSec1, chkRecSec1)

If chkSquSec1.Checked = True Then

Sec1 = 2

lblLenSec1.Text = "Side Length"

txtWidSec1.Enabled = False

lblWidSec1M.Enabled = False

lblWidSec1.Enabled = False

Else

lblLenSec1.Text = "Length"

txtWidSec1.Enabled = True

lblWidSec1M.Enabled = True

lblWidSec1.Enabled = True

End If

End Sub

'Sets shape of Sec2 of the floor to a square (2) and disables the rectangle check box and its corresponding UI controls if square check box is still checked

Private Sub chkSquSec2\_CheckedChanged(sender As Object, e As EventArgs) Handles chkSquSec2.CheckedChanged

FloorShape(chkSquSec2, chkRecSec2)

If chkSquSec2.Checked = True Then

Sec2 = 2

lblLenSec2.Text = "Side Length"

txtWidSec2.Enabled = False

lblWidSec2M.Enabled = False

lblWidSec2.Enabled = False

Else

lblLenSec2.Text = "Length"

txtWidSec2.Enabled = True

lblWidSec2M.Enabled = True

lblWidSec2.Enabled = True

End If

End Sub

'Sets shape of Sec3 of the floor to a square (2) and disables the rectangle check box and its corresponding UI controls if square check box is still checked

Private Sub chkSquSec3\_CheckedChanged(sender As Object, e As EventArgs) Handles chkSquSec3.CheckedChanged

FloorShape(chkSquSec3, chkRecSec3)

If chkSquSec3.Checked = True Then

Sec3 = 2

lblLenSec3.Text = "Side Length"

txtWidSec3.Enabled = False

lblWidSec3M.Enabled = False

lblWidSec3.Enabled = False

Else

lblLenSec3.Text = "Length"

txtWidSec3.Enabled = True

lblWidSec3M.Enabled = True

lblWidSec3.Enabled = True

End If

End Sub

'Sets shape of Sec4 of the floor to a square (2) and disables the rectangle check box and its corresponding UI controls if square check box is still checked

Private Sub chkSquSec4\_CheckedChanged(sender As Object, e As EventArgs) Handles chkSquSec4.CheckedChanged

FloorShape(chkSquSec4, chkRecSec4)

If chkSquSec4.Checked = True Then

Sec4 = 2

lblLenSec4.Text = "Side Length"

txtWidSec4.Enabled = False

lblWidSec4M.Enabled = False

lblWidSec4.Enabled = False

Else

lblLenSec4.Text = "Length"

txtWidSec4.Enabled = True

lblWidSec4M.Enabled = True

lblWidSec4.Enabled = True

End If

End Sub

'Sets shape of Sec1 of the floor to a rectangle (1) and disables the square check box if rectangle check box is still checked

Private Sub chkRecSec1\_CheckedChanged(sender As Object, e As EventArgs) Handles chkRecSec1.CheckedChanged

FloorShape(chkRecSec1, chkSquSec1)

If chkRecSec1.Checked = True Then

Sec1 = 1

End If

End Sub

'Sets shape of Sec2 of the floor to a rectangle (1) and disables the square check box if rectangle check box is still checked

Private Sub chkRecSec2\_CheckedChanged(sender As Object, e As EventArgs) Handles chkRecSec2.CheckedChanged

FloorShape(chkRecSec2, chkSquSec2)

If chkRecSec2.Checked = True Then

Sec2 = 1

End If

End Sub

'Sets shape of Sec3 of the floor to a rectangle (1) and disables the square check box if rectangle check box is still checked

Private Sub chkRecSec3\_CheckedChanged(sender As Object, e As EventArgs) Handles chkRecSec3.CheckedChanged

FloorShape(chkRecSec3, chkSquSec3)

If chkRecSec3.Checked = True Then

Sec3 = 1

End If

End Sub

'Sets shape of Sec4 of the floor to a rectangle (1) and disables the square check box if rectangle check box is still checked

Private Sub chkRecSec4\_CheckedChanged(sender As Object, e As EventArgs) Handles chkRecSec4.CheckedChanged

FloorShape(chkRecSec4, chkSquSec4)

If chkRecSec4.Checked = True Then

Sec4 = 1

End If

End Sub

'Processes all inputs entered by the user to output (floor's area, room's volume and amount of paint required to paint the walls) for the user

Private Sub btnCal\_Click(sender As Object, e As EventArgs) Handles btnCal.Click

Try

SectionsArea(1, 2)

If RegFloor = True Then

TotalFloorArea = AreaSec1

ElseIf RegFloor = False And cboNumSecs.SelectedItem = 2 Then

TotalFloorArea = AreaSec1 + AreaSec2

ElseIf RegFloor = False And cboNumSecs.SelectedItem = 3 Then

TotalFloorArea = AreaSec1 + AreaSec2 + AreaSec3

ElseIf RegFloor = False And cboNumSecs.SelectedItem = 4 Then

TotalFloorArea = AreaSec1 + AreaSec2 + AreaSec3 + AreaSec4

End If

txtArea.Text = TotalFloorArea

RoomVolume = txtHeightRoom.Text \* TotalFloorArea

If Roof = 1 Or Roof = 3 Then

Select Case Roof

Case Roof = 1

RoofVolume = 1 / 3 \* txtLenRoof.Text \* txtHeightRoof.Text

Case Roof = 3

RoofVolume = 1 / 2 \* txtLenRoof.Text \* txtHeightRoof.Text

End Select

TotalVolume = RoofVolume + RoomVolume

Else

TotalVolume = RoomVolume

End If

txtVolume.Text = TotalVolume

ReqPaint = txtLenWall1.Text \* txtWidWall1.Text + txtLenWall2.Text \* txtWidWall2.Text \_

+ txtLenWall3.Text \* txtWidWall3.Text + txtLenWall4.Text \* txtWidWall4.Text

If cboNumDoors.SelectedItem = 1 Then

ReqPaint = ReqPaint - txtLenDoor1.Text \* txtWidDoor1.Text

ElseIf cboNumDoors.SelectedItem = 2 Then

ReqPaint = ReqPaint - txtLenDoor1.Text \* txtWidDoor1.Text + txtLenDoor2.Text \* txtWidDoor2.Text

End If

If cboNumWindows.SelectedItem = 1 Then

ReqPaint = ReqPaint - txtLenWin1.Text \* txtWidWin1.Text

ElseIf cboNumDoors.SelectedItem = 2 Then

ReqPaint = ReqPaint - txtLenWin1.Text \* txtWidWin1.Text + txtLenWin2.Text \* txtWidWin2.Text

End If

Select Case cboNumCoats.SelectedItem

Case cboNumCoats.SelectedItem = 1

ReqPaint = ReqPaint \* 0.065

Case cboNumCoats.SelectedItem = 2

ReqPaint = ReqPaint \* 0.13

Case cboNumCoats.SelectedItem = 3

ReqPaint = ReqPaint \* 0.195

End Select

txtReqPaint.Text = ReqPaint

Catch ex As Exception

MsgBox("Error, please make sure you have selected both the type of room's floor and roof/celling")

txtArea.Text = ""

txtVolume.Text = ""

txtReqPaint.Text = ""

End Try

End Sub

'Sub procedure used to correctly process and calculate all rectangle/square area values of the 4 different sections/sectors of the floor

Private Sub SectionsArea(a, b)

If Sec1 = a Then

AreaSec1 = txtLenSec1.Text \* txtWidSec1.Text

ElseIf Sec1 = b Then

AreaSec1 = txtLenSec1.Text ^ 2

End If

If Sec2 = a Then

AreaSec2 = txtLenSec2.Text \* txtWidSec2.Text

ElseIf Sec2 = b Then

AreaSec2 = txtLenSec2.Text ^ 2

End If

If Sec3 = a Then

AreaSec3 = txtLenSec3.Text \* txtWidSec3.Text

ElseIf Sec3 = b Then

AreaSec3 = txtLenSec3.Text ^ 2

End If

If Sec4 = a Then

AreaSec4 = txtLenSec4.Text \* txtWidSec4.Text

ElseIf Sec4 = b Then

AreaSec4 = txtLenSec4.Text ^ 2

End If

End Sub

'Clears content of UI controls to reset them back to their default/initial values

Private Sub btnReset\_Click(sender As Object, e As EventArgs) Handles btnReset.Click

For Each txtbox In {txtLenSec1, txtWidSec1, txtLenSec2, txtWidSec2, txtLenSec3, txtWidSec3, txtLenSec4, txtWidSec4, txtLenWall1,

txtWidWall1, txtLenWall2, txtWidWall2, txtLenWall3, txtWidWall3, txtLenWall4, txtWidWall4, txtLenDoor1,

txtWidDoor1, txtLenDoor2, txtWidDoor2, txtLenWin1, txtWidWin1, txtLenWin2, txtWidWin2, txtHeightRoof, txtHeightRoom,

txtLenRoof, txtWidRoof, txtArea, txtVolume, txtReqPaint}

txtbox.Clear()

Next

cboNumSecs.SelectedIndex = 0

cboNumDoors.SelectedIndex = 0

cboNumWindows.SelectedIndex = 0

cboNumCoats.SelectedIndex = 0

For Each chkbox In {chkRecSec1, chkSquSec1,

chkRecSec2, chkSquSec2,

chkRecSec3, chkSquSec3,

chkRecSec4, chkSquSec4}

chkbox.Checked = False

Next

rdoReg.Checked = True

rdoPyr.Checked = True

End Sub

'Allows to enable relavent UI controls and disables irrelavent UI controls depending on the selected value by the user

Private Sub cboNumDoors\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cboNumDoors.SelectedIndexChanged

If cboNumDoors.SelectedItem = 2 Then

For Each Ctrl In grpRoom.Controls

If Ctrl.Tag = "Door2" Then

Ctrl.Enabled = True

End If

Next

Else

For Each Ctrl In grpRoom.Controls

If Ctrl.Tag = "Door2" Then

Ctrl.Enabled = False

End If

Next

End If

End Sub

'Allows to enable relavent UI controls and disables irrelavent UI controls depending on the selected value by the user

Private Sub cboNumWindows\_SelectedIndexChanged(sender As Object, e As EventArgs) Handles cboNumWindows.SelectedIndexChanged

If cboNumWindows.SelectedItem = 2 Then

For Each Ctrl In grpRoom.Controls

If Ctrl.Tag = "Window2" Then

Ctrl.Enabled = True

End If

Next

Else

For Each Ctrl In grpRoom.Controls

If Ctrl.Tag = "Window2" Then

Ctrl.Enabled = False

End If

Next

End If

End Sub

'Used to verify all textboxes' content to only allow numbers and decimal points to be entered by the user

Private Sub txtLenSec1\_KeyPress(sender As Object, e As KeyPressEventArgs) Handles txtLenSec1.KeyPress, txtWidSec1.KeyPress,

txtLenSec2.KeyPress, txtWidSec2.KeyPress, txtLenSec3.KeyPress, txtWidSec3.KeyPress, txtLenSec4.KeyPress,

txtWidSec4.KeyPress, txtLenWall1.KeyPress, txtWidWall1.KeyPress, txtLenWall2.KeyPress, txtWidWall2.KeyPress,

txtLenWall3.KeyPress, txtWidWall3.KeyPress, txtLenWall4.KeyPress, txtWidWall4.KeyPress, txtLenDoor1.KeyPress,

txtWidDoor1.KeyPress, txtLenDoor2.KeyPress, txtWidDoor2.KeyPress, txtLenWin1.KeyPress, txtWidWin1.KeyPress,

txtLenWin2.KeyPress, txtWidWin2.KeyPress, txtHeightRoof.KeyPress, txtHeightRoom.KeyPress, txtLenRoof.KeyPress,

txtWidRoof.KeyPress

For Each txtbox In {txtLenSec1, txtWidSec1, txtLenSec2, txtWidSec2, txtLenSec3, txtWidSec3, txtLenSec4, txtWidSec4, txtLenWall1,

txtWidWall1, txtLenWall2, txtWidWall2, txtLenWall3, txtWidWall3, txtLenWall4, txtWidWall4, txtLenDoor1,

txtWidDoor1, txtLenDoor2, txtWidDoor2, txtLenWin1, txtWidWin1, txtLenWin2, txtWidWin2, txtHeightRoof, txtHeightRoom,

txtLenRoof, txtWidRoof}

If Not e.KeyChar = “.” And Not Char.IsNumber(e.KeyChar) Then

e.Handled = True

End If

If e.KeyChar = Chr(&H8) Then

e.Handled = False

End If

Next

End Sub

End Class