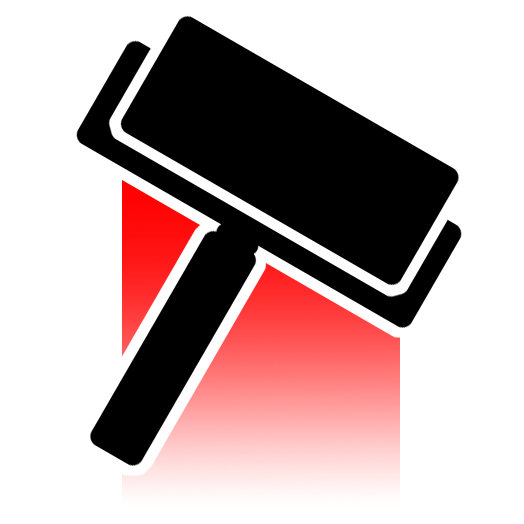
Documentation for PaintMate



# Table of contents:

Analysis & Design…………………………………………………………………………………………………………………………..2, 3

Testing……………………………………………………………………………………………………………………………………………….4

Conclusion…………………..…………………………………………………………………………………………………………………….5

Analysis & Design

# Project identification

This is a project that will be designed and made for the sole purpose of participating in Borwell’s software challenge. The specification outlines the following criteria for the software challenge:

Design and create 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

This challenge only has 1 restriction, it must be written in either C#, Java or C++.

# Project Design

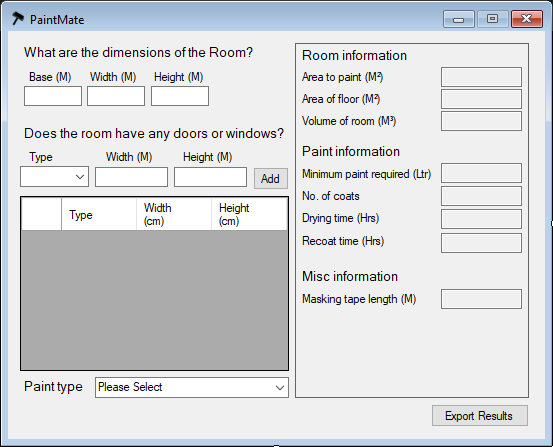
After further inspection of the outlined criteria, I noticed that this specified program would not consider windows and doors, Fixtures that take up wall space and would not require to be painted over. Another problem I noticed is that in a few cases, different paints require a different amount to paint the same area across the board. At this point, I made a few additions to the specification

Pavilion’s personal specification

* Area of floor
* Amount of paint required to paint the walls for each type of paint
* Volume of room
* Be able to account for doors and windows
* Be able to tell you how much masking tape you would require
* Be able to export the results in a text file
* Inform the user about drying time and coats required

# Look and feel

I set myself a time restriction of 6 hours from start to finish, this will include designing the program. I want the program to be user friendly and easy to use despite the extra features I will add. My end result for the look and feel of the project was this:



Testing

The tests I came up with involved simple data input tests, where I test all the input boxes with invalid data, erroneous data, and extreme data. At first, The program failed all tests. So after some tweaking, I implemented code to check for incorrect data, such as:

if(txtRoomBase.Text=="" || txtRoomHeight.Text=="" || txtRoomWidth.Text=="")

if (float.TryParse(txtRoomBase.Text, out f1) && float.TryParse(txtRoomHeight.Text, out f2) && float.TryParse(txtRoomWidth.Text, out f3))

try

{

//code

}

catch (Exception ex)

{

MessageBox.Show(ex.Message, "PaintMate");

}

Conclusion

In conclusion, I have completed the program within the time limit I set myself with all the extra features that I incorporated. If I had more time, I would make the program more user friendly and possibly incorporate even more features such as:

* Unit conversions (Metric and imperial)
* More user friendly design
* Paint shopping options with price approximation