Design Report and Evaluation

## **System Development Cycle**

**What is it?**

The systems life cycle is a model used in project management that describes the stages involved in a development cycle, these stages involve everything in designing and creating the software solution, and includes stages from planning, design through to deployment and maintenance. These are just some of the stages involved, there are many more steps that must be taken to develop a successful program. Each of these steps are a crucial stage in the lifecycle of software development if not used correctly the final product can be a mess and be hard to implement into use, additionally it will be hard to maintain and could be easily broken.

**The steps involved.**

The steps involved in the Software development cycle start at number one, planning. This stage is extremely important because at you get to find out what the client’s brief and requirements are this will include their needs and what they want from the final solution. Once you have done this you can start to analyse the specification and break down the requirements into parts that can be managed easier this will make the whole problem more achievable, taking a possibly impossible task and making it easier and more manageable. When planning my solution, I will create a bull pointed list of all the tasks I must complete as well as a list of objectives I want the solution to achieve. This part of the life cycle is possibly the most important stage e.g. the make or break of the task. It will make the following stages a lot easier for example a well thought out/ analysed software idea will be a lot easier implemented if designed well. This leads onto the third stage of the cycle design. This is where all the essential user interface designs are done, in this stage designs are prepared from the requirements which have been given from the client. Types of designs include flowcharts and or pseudocode these represent simplified versions of the final solution they help you construct the final solution by giving you a good idea of how to go about the task. Other designs include user interface mocks, at this point you will create Ideas of your user interface and how it will look this step it also important because they help you get feel for the final solution. This is the stage when I will start to design my solution. These designs will include a mock UI (User interface) and flowcharts for the main body of the code. Once you have completed you designs you can start developing the solution at this stage the application is created. Although there will be problems with the application the main program is created. Next the created application is thoroughly tested to ensure all issues are found. This is where I create testing documentation to ensure any issues found are logged and recorded so they can be fixed. The program is then altered so that the issues found, are dealt with and fixed. Once all the testing Is complete and all bugs are fixed the software package is ready to be released to the client for use. Some more issues may be encountered therefore a well thought out design with a maintainable program is paramount because this means and future issues or features that want to be changed can be easily. To make sure my code is maintainable I enclose similar blocks of code in regions and make sure to thoroughly comment my code so that it can be easily understood.

**The process I will use:**

To ensure I develop the best solution for the client I will follow the system development process. I will firstly, plan the solution to do this I will start by understanding the scope of the project this will include, why I need to create this solution. In my plan I will also understand the requirements the client needs this will help me later understand what the application needs to include. The requirements I pull from the brief are extremely important because they will be the basis of my solution. During the planning phase I will also need to understand the deadline the client has given me; this will be the time/date that the final solution and reports must be finished. After I finished all project planning, I can start designing the software package. These designs will include flowcharts of the main body of code I will use to make up my program, there will also be designs for the user interface, this will be front face of the program this has to be right otherwise the client won’t want to use the solution I have provided. During this phase I will also decide the software architecture and programming language. Once all my designs are complete documentation will be created to list all these designs, this will help to keep them in one place that they can be easily accessed. Once this document is done, I will then get third party review for the designs. This will give me crucial feedback and essentially tell me if the designs need to be changed or redone completely. Once all designs and feedback are complete, I can move onto developing the solution, this is where the program is made. The program is going to be completed in one block. To ensure future phases are easily completed I will clearly comment and region my code. This will make sure maintainability is kept to a high and any future work on the program will be easily done. Once the block of code is completed, I can move on and test it. I will be testing for three main issues these will include, valid data, invalid and extreme data. This will cover a board perspective as to whether the program works successfully. Any issues I come across will be documented, I will also document how I came across the issue and what caused the issue, I will then move onto to fixing the problem and if the issue is fixed I will retest and see if the error persists I will complete this process until I find no more problems with the application. After the testing phase I will now move onto the final phase during this phase any new features or bugs that need to be fixed are added, Maintainability at this phase is paramount and if my code is not organised and well commented this stage will a lot harder.

## **Project Plan**

Start date: 17/01/2020

~~\* Create a plan for the whole project.~~

~~\* Produce a design report~~

~~\* Systems development cycle~~

~~\* Problem definition~~

~~\* Intended users~~

~~\* Benefits of the solution~~

~~\* Problem description~~

~~\* list of objectives~~

~~\* Programming language justification~~

~~\* Create Test Plan~~

~~\* Create User Interface design~~

~~\* Flowcharts~~

~~\* main body of code involved~~

~~\* code used for validation checks~~

~~\* Justification for the designs (why they will be effective)~~

--> Expected to spend 4.5 hours

~~\* Third party review of designs~~

~~\* Act on feedback given~~

--> Expected to spend 1/0.5 hours

~~\* Develop Final solution~~

~~\* Comment code for ease of use~~

~~\* Make screenshots of errors that occur~~

~~\* Add error to test plan including how they occurred~~

--> Expected to spend 14 hours

~~\* Annotate final user interface~~

~~\* Finalise Test plan (including test results + data used)~~

~~\* Show Evidence of fixing errors~~

--> Expected to spend 2 hours

~~\* Third party review of my program~~

~~\* store comments on program~~

~~\* act on comments if necessary~~

--> Expected to spend 1 hours

~~\* Show how I managed time~~

~~\* Show how I reviewed and responded to feedback~~

~~\* Show how I justified recommendations and decisions~~

\* Evaluate my final solution

~~\* Evaluate success of solution~~

\* How well design, review and testing helped produce a solution

--> Expected to spend 4 hours

Total expected time: 26.5 hours

Over 6-week period.

MUST COMPLETE BY 28/02/2020

## **Summary of the Problem**

The client, a small traditional plastics manufacturer, making a range of standard items but also offers a customised product where customers can order to exact specifications. They are looking to move from a paper system to computerised system. Currently a salesperson uses a standard calculator and multiplier sheet. The client wishes to replace the old paper system with an electronic custom quote system. The client would like an electronic version where the sale person can enter the variables involved and all the calculations are completed automatically. They would also like the quote details to be saved to a text file, so a formal quote can be made later.

## **Intended Users**

Ideally the program will be used by a salesperson and or the secretary working for the traditional plastic company. They will use the program to calculate quotes for customers and generate the formal invoice. This will be done by the salesperson who enters the exact variables and the program will do the calculations and will produce a cost for the customer.

## **Benefits from this solution**

There will be many added benefits. These will include decreased time calculating the price as the program will do all the calculations this will allow the salespersons to allocate more time to other tasks. There will be less errors on the sales-persons behalf as the program will not make errors when calculating. Finally, by saving the quote in a text file will allow the client to produce a formal quote to send to the customer via mail merge at a later point.

## **Objectives**

* Must be able to enter variables for the customer.
* Must costs total cost depending on the variables entered.
* Must be simple to use.
* Must be able to output the total cost for the quote to a file.
* Be able to validate the inputs the user enters.
* be able to save clients Name, Date and final price to .txt file.
* Be able to show the final cost.

## **Language** **Justification**

I will be using C# due to excellent functionality that it provides in visual studio. I chose C# due to the functionality, including a big library of functions and explanations on Microsoft docs and the ability to use visual studio integrated development environment this would mean I would be able to complete all the objectives for this task confidently.

## **Designs Justification**

The designs I have produced are going to be effective in helping me produce a successful solution to the client’s brief. My designs are going to help me because they show a breakdown of the tasks, I must complete this allows me to manage my time effectively. The user interface designs show me a clear idea of how my user interface is going to look like. This will stop me from wasting time trying to design my UI (user interface) in visual studio. The flowcharts I created will also help me by giving me an idea of where to start on the implementation stage.

## **Third Party Reviews**

Third part review completed by Euan Mason and William Gilbert

**User Interface:**

Euan said:“You need to add a section for the name and date of the client the quote is for because this will be easier for a formal quote to be produced. You also should consider using combo boxes for some of the categories as this will be more effective at selecting the option required.”

After this feedback I decided to make some changes. These changes include adding a box for the client’s name and the quote date. This will be added to the text document containing the final price. I also swapped some of the radio buttons for combo boxes and check boxes. I did this because it made more sense to have combo boxes for the categories such as material type, moulding technique and design and mould creation, because there are multiple options that could be selected but only one option can be selected at once. I also changed the radio buttons for the finishing method to check boxes this is because multiple finishing methods can be used this allowed for one method or all the methods to be selected.

**Main Program:**

Euan said: “Your code is well commented and clearly defined by their regions; I could easily find a section of code if I wanted to”

After receiving this feedback, I continued to clearly label my code, and comment when necessary.

Will said:“You should use a date time picker to select the date because it already pre-sets the date, so the user does not have to. This can prevent the user no entering a date and saves the user from getting an error”

Following this feedback, I decided to act on it and change the masked text box to a date and time picker, this will make the date selection process more efficient. If the user forgets to enter a date the program will have it already set.

## **Testing Documentation.**

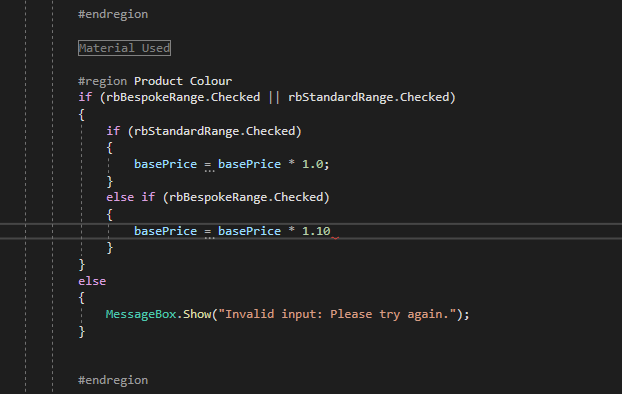
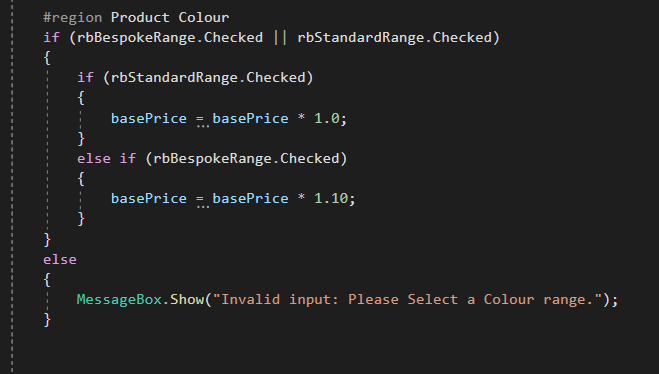
The next few pages show the testing phase to my solution. It shows how I tested, made changes and logged the information. The first page shows the test plan I made to log any errors that occurred it shows a brief description of any error and any changes I made to fix the error. After the test plan it shows in detail the process, I went through to make the error happen and the in-detail process of how I overcome and fixed the problem. There are also screenshots to provide evidence as to the error that occurred and the fix, I applied to solve the issue.

I did a range of tests for example, valid, invalid, extreme data inputs and data validation. this allows me to get an overview of how my program will react when these values are entered, I also did tests to ensure the calculations results are as expected. The data validation test was conducted to ensure the correct data type would be entered.

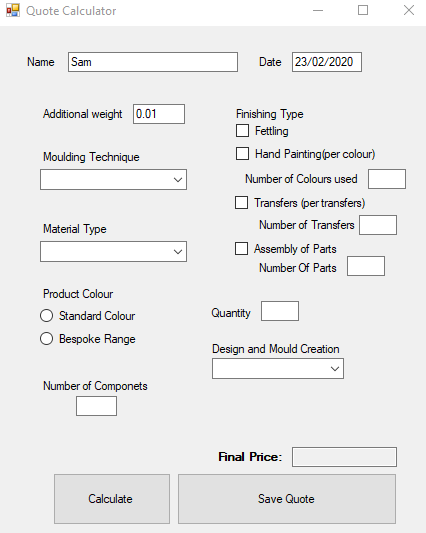
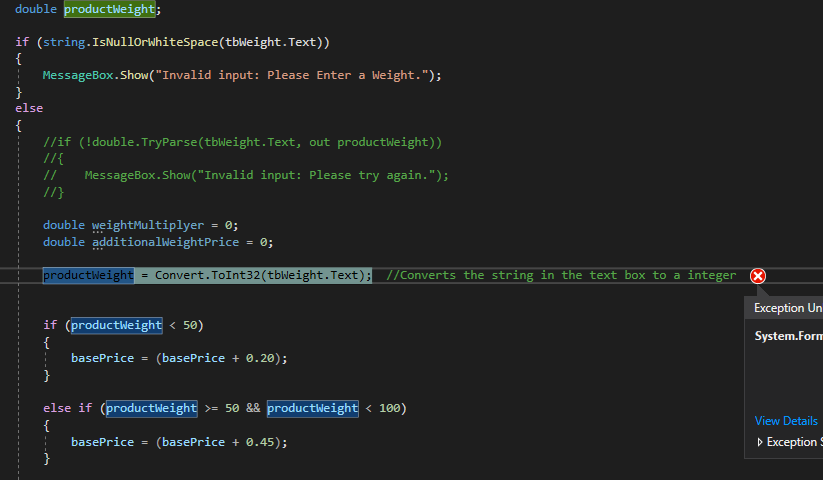
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test number | Testing? | Expected Outcome | Actual Outcome | Changes made | Details of error |
| 1 | Testing for build errors | Program loads with no errors | 1 build error was found, missing semi-colon on line 198. | Semi-colon was added | I forgot to add a semi-colon this caused and error |
| 2 | Testing for extreme values weight = 0.01 | Program continue as normal | Program crashes | I blocked the ability to enter the character full stops | The program crashes when a value less than 1 is entered so the value full stops are blocked |
| 3 | Testing the finishing method transfers by using 75grams, injection moulding, HDPS, standard colour range, 1 component, 1 transfer | 0.54 | 0.45 | Added transfers to if statement | This error occurred because I had forgotten to add a condition to the if statement, so it was not getting picked when the checkbox is checked the code is not registering |
| 4 | Testing finishing methods (assembly of parts using 2) | 0.495 | 0.45 | Fixed indentation | This error was caused by indentation error that was present in the assembly of parts selection statement |
| 5 | Testing the text documents contains three variables, Client name, final price and date of quote.  Name: Sam Date: 23/02/2020 FinalPrice:0.45 | 23/02/2020  The final price for Sam with the options that were selected is: £0.45 | / /  The final price for with the options that were selected is: £0 | Added $ to start of string with {around variables} | The program didn’t save the relevant information that the quote must include |
| 6 | Testing for invalid data entry’s for name date and weight | Name: 12312  Date: AB/CD/EFGH  Weight: AbCdEfG | Name:  Date:  Weight: | N/A | No changes necessary due to not being able to enter an invalid data type. |
| 7 | Testing the Final Price for 75g, coating, PVC, Standard colour range, 5 components, Hand Painting 1 colour, quantity 750, mould already made. | Final price = 2180.95 | Final price = 2180.95 | N/A | The results that I expected was the correct result, so no error occurred. |

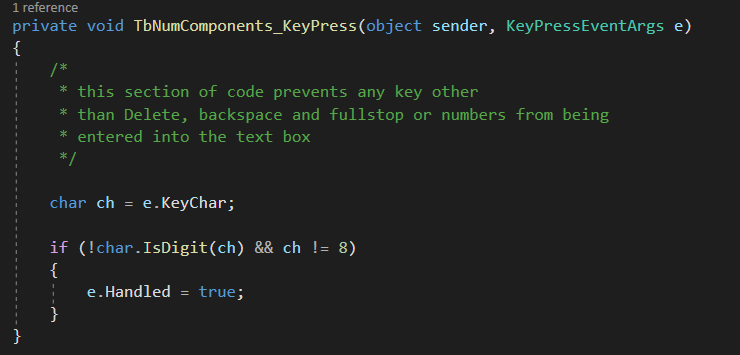
## **Further Details of Errors.**

Error 1: Testing for Build Errors

In this test I simple ran the program in visual studio this would tell me if there were any immediate errors. The test showed one immediate error on line 198. This showed me that a semi-colon was missing. The missing semi-colon prevented the program from working because without it the compiler will not know when the code finishes for that line, so the code becomes one big syntax error. I proceeded to fix this error by adding a semicolon on the end of line 198 this then fixed the error and the program then work as expected.

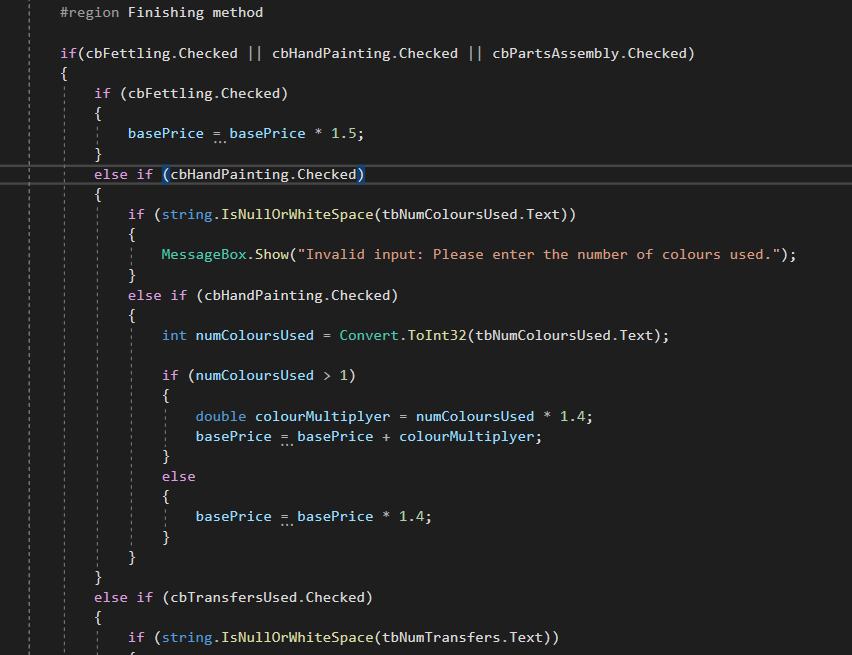
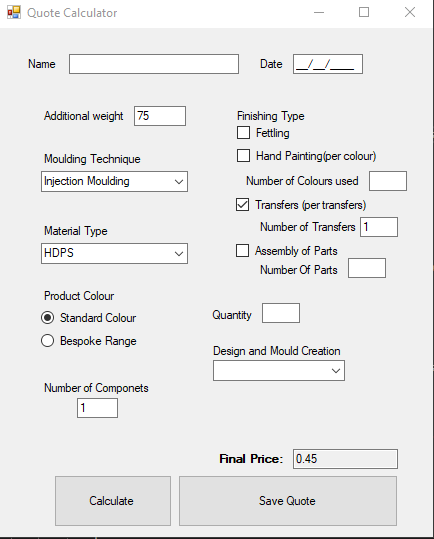
Error 2: Extreme Values

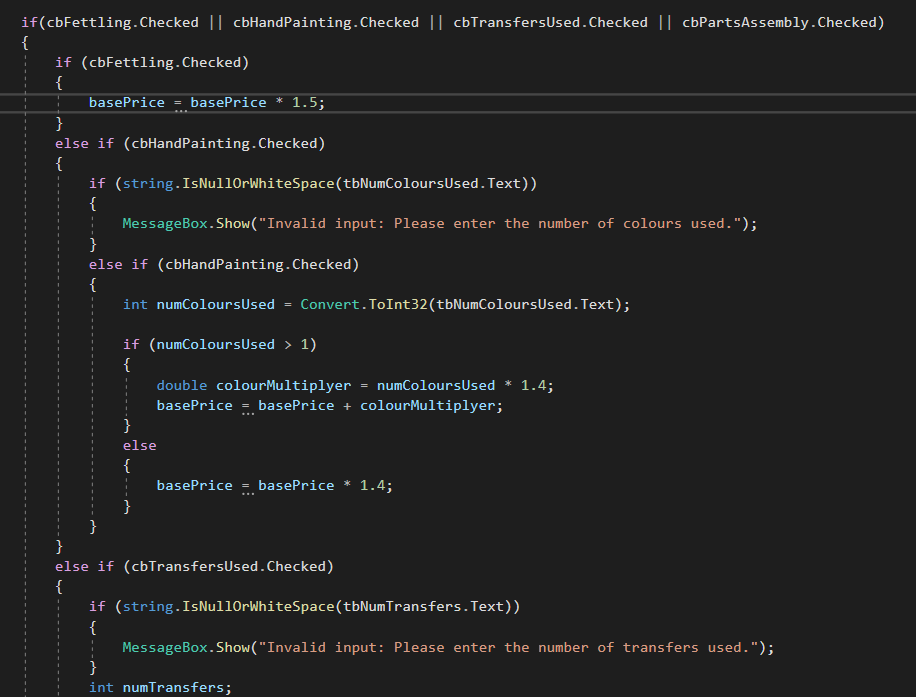
In this test I used the program but instead of using values intended to be used, I used a value that is not supposed to be used, I used 0.01 a value less than 1 this caused the program to crash and stop working. My solution to this was to block the entry of full stops this will prevent number less than one from being entered. This solution is like how I stopped letters being entered to textboxes that should only allow numbers



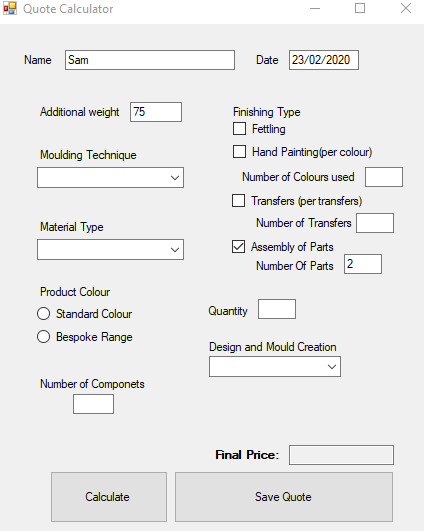
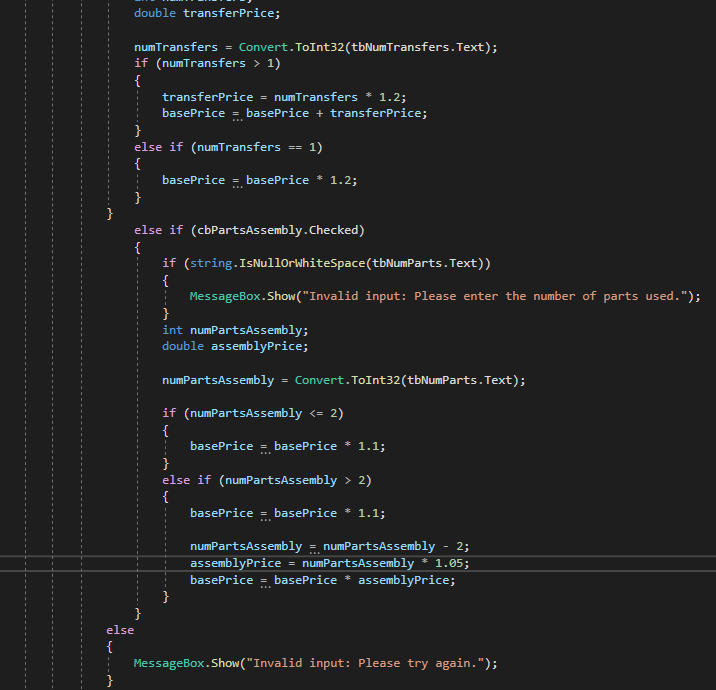
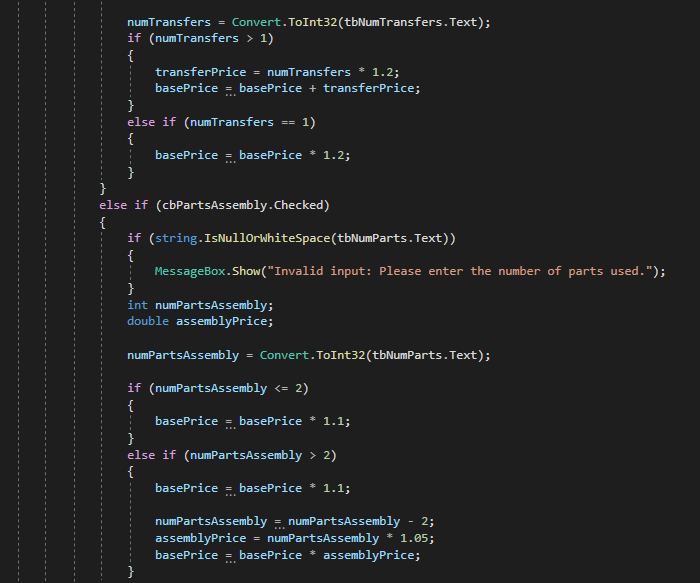
Error 3: Missing Condition in IF statement.

In this test I was testing to see if the finishing methods gave the correct values. In this test I used the following values, weight of 75g, Injection moulding, HDPS, standard colours, 1 Component and transfers. The program gave a final price of 0.45, the correct value should be 0.54. The errors lied in the selection statement that decided if the user had chosen a finishing method, I had forgotten to add the statement to check if transfers had been selected. Once I had put the transfers into the if statement it worked as expected.

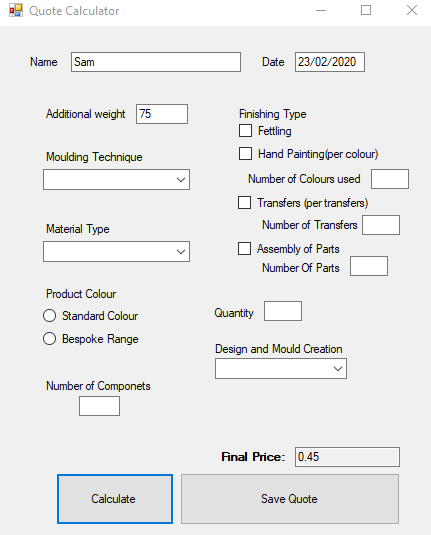


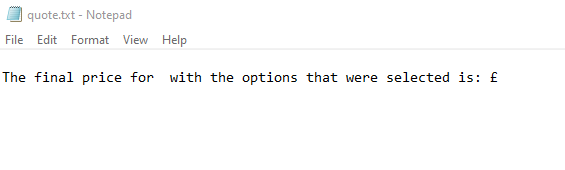
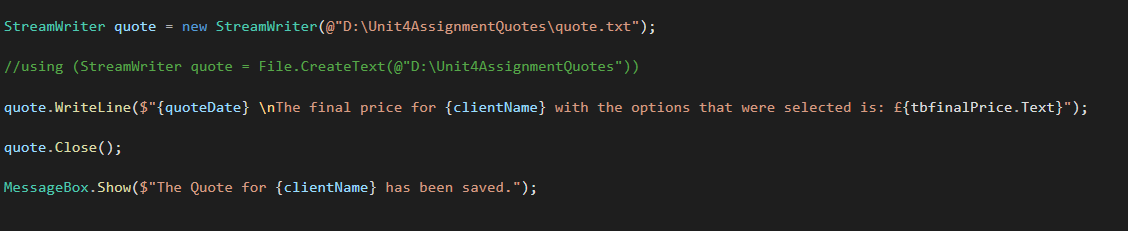


Error 4: Invalid Indentation.

In this error there was a problem with the selection statements because there was an indentation problem with the finishing methods specifically assembly of parts. This happened because the assembly of parts was embedded in a different selection statement when It should have been related to the main if statement. This was fixed by unindenting the Assembly or parts block to realign it with the main if statement.

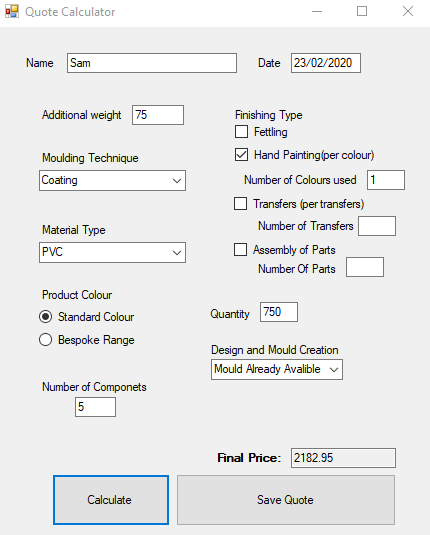
Error 5: Text Document Quote.

In this test I was testing that the text file would save the relevant information for the formal quote to be generated later. I tested that the quote would save the name, date and final price would be saved to the text file, I conducted the test by entering the information I wanted to be save to the text file. I then saved the quote and checked the text file. The information I wanted it to save did not save. I checked the code that was responsible for writing to the text file and I found out that I needed to change the string to allow for variables to be allowed. I then changed the string and the text file saved the information I wanted it to.



Error 6: Invalid Data Entries.

In this test I tested the input or invalid entries. I conducted this test by trying to enter letters into text boxes that should only allow numbers and vis versa. I had already setup data validation in my program so there wasn’t an error that occurred. Because there was no error, no changes were necessary.



Error 7: Testing the Correct Final Price.

In this test I tested for the correct final price for a weight of 75g, Coating, PVC, Standard, 5 components, Hand-painting 1 colour with 750 units(quantity) and no new mould required. I as I tested this, I expected a result of £2182.95 and the actual result was £2182.95 the test results were correct, so no changes were needed.

## **Evaluation**

I created a successful solution to solve the client’s brief. Which was, an electronic calculator to calculate the cost of the customers specifications. This program allows the user, ideally a salesperson that works for the traditional plastics company, to enter variables that the customer has specified and will calculate the final total price for that product. This is the alternate to the standard multiplier sheet and traditional calculator.

The solution I created was effective at meeting the client’s requirements, although I encountered one concern, this was encountered with my solution is, when the final price is saved to the quote text file It doesn’t show the specifications the customer ordered to get to that final price, it only states the clients name, the date the quote was produced and the final price. This will make it harder for the formal quote to be generated as they do not know the specifications of the product in question. This was the only part of the clients brief that did not fully meet their requirements. I found some parts of this assignment harder than others, the main part I found hard included the software development phase, I found this particularly hard because some of the functions I used I needed to spend time researching how to use them, I spent quite a lot of time doing this, but to ensure I created the best solution I could this had to be done.

The project planning, I undertook helped me manage my time by helping me breaking down each task needed to be complete while also giving me an idea of how much time they would take. I then produced a set of plans this would include flowcharts of the main program this would help me by giving me a place to start when it came to the development phase, and flowcharts for the data validation this was to ensure the data that I entered would be the correct data type; this gave me relatively good idea of the code that was going to make up my solution. I then proceeded to create some user interface designs this gave me an idea of what the program would look, so I wouldn’t be trying to design it on visual studio, essentially wasting valuable time when It came to start programming. These plans allowed me to really focus on the code that would make up the entirety of the solution. Once I had completed the designs I could start thinking about the development of my solution in visual studio. I decided to develop my code in blocks that were broken down by each stage of the multiplier sheet for example, weight, material type and colour range these were just some of the categories that would make up the solution. This allowed me to test the code of each block one at a time, this allowed me to control any errors that might occur and fix them while I was working on that block of code. I came across lots of issues ranging from problems with my maths to having incorrect indentations for my selection statements. When programming I realised, I made some of the easiest mistakes. When I found an error, I filled out my test plan this meant that I could keep a log of not only the errors but also what caused the error and any changes I made to the prevent the error in the future. Once I had finished with a block of code, I could enclose it in a region clearly labelled and giving a brief description of what that block of code would do. Then I would move onto another block ensuring to clearly label using comments, this ensured I preserved maintainability for any future work to be done on my solution.

Overall my solution was effective at meeting the clients brief. Due to all the planning and testing I did. The plans helped me structure this assignment, significantly by helping me lay out the tasks that needed to be done instead of just going for it, not really knowing what I was doing and ending up with a unorganised mess, the plans helped me understand what needed to be done, this was underlying reason for my effective solution. Not to mention the testing that underwent improve the final solution. Without the testing my solution would not be nearly as polished due to the shear amount of errors that would be present. In conclusion my knowledge of computational thinking and the ability to plan effectively is the main reason to the successful solution I can deliver to the client.