**CHAPTER THREE**

**3.0 RESEARCH METHODOLOGY**

This section of the project work presents the details of the software package developed step by step numerical computations.

The Server was developed and it contains electrical models/equations which have already been coded. The server was linked to the client application so that the equations can be called whenever it’s needed for computation. The Client Application was developed. The Client application is a user interface where the user can do the computations.

The first user interface has two buttons namely CONTINUE and EXIT, which are designed and coded to perform the task of the user to another linked interface called the switchboard. The EXIT button was coded to perform the task of exiting the software/application whenever the users want to. The Interface was also designed to show the name of the software and the name of the developer as shown in the fig. 3.1 below

Clicking on the continue button, links the user to another user interface called the SWITCH BOARD. The switch board has five (5) buttons as shown in the Fig. 3.2 .the buttons are designed and coded to perform the following task:

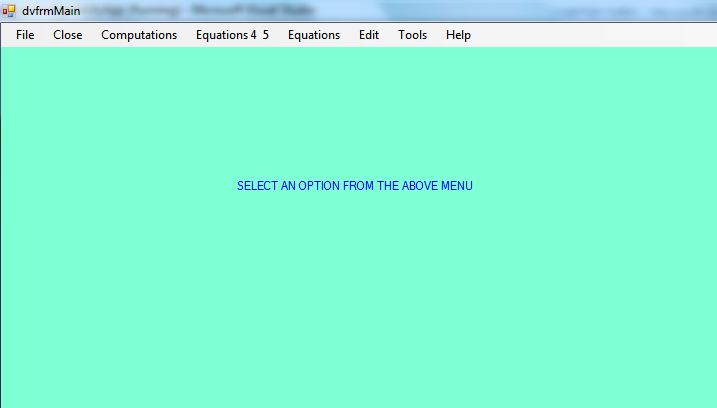
**BUTTON 1:** The button was named “BACK” and it was coded to link the user back to the first user interface.

**BUTTON 2:** The button was named “COMPUTATION” and it was designed and coded to link the user to another interface, THE MAIN MENU, where the user can compute the models.

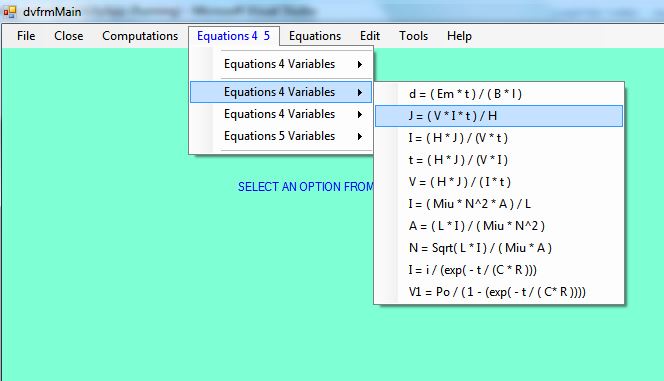
**BUTTON 3:** The button was named “ELECTRICITY” and it was designed and coded to link the user to another interface that gives more information about electricity.

**BUTTON 4:** The button was named “E-PICTURES” and it was coded to link the user to link the user to another interface that shows the pictures about electricity.

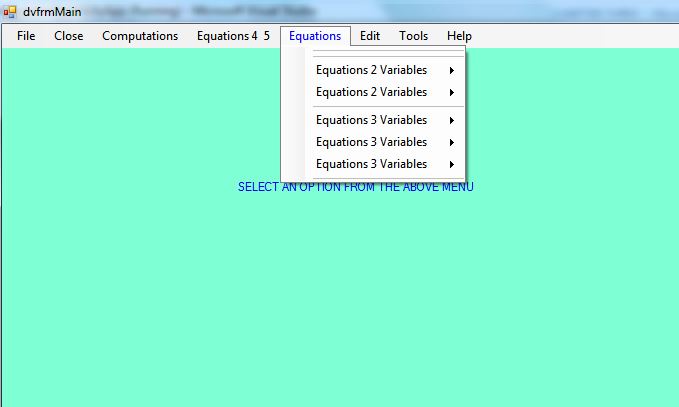
The main menu was constructed using the toolbox. This new interface comprises of eight (8) menus as shown in the Fig. 3.3 below.



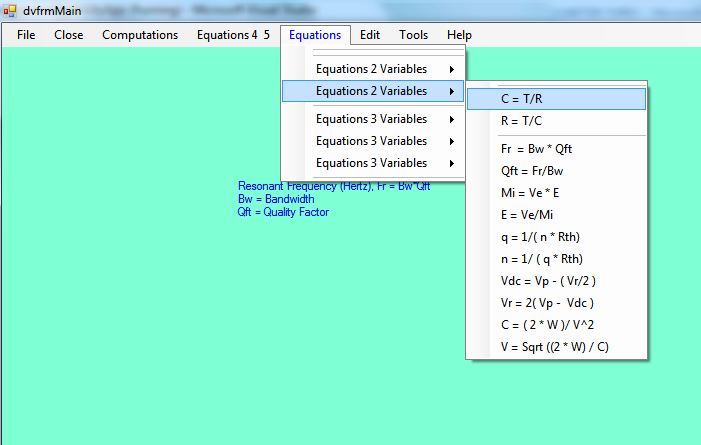
This interface is referred to as the working environment of the software package, when these menus are clicked, some display different number submenus. In each submenu, another submenu are constructed as shown in Fig. 3.4 below.

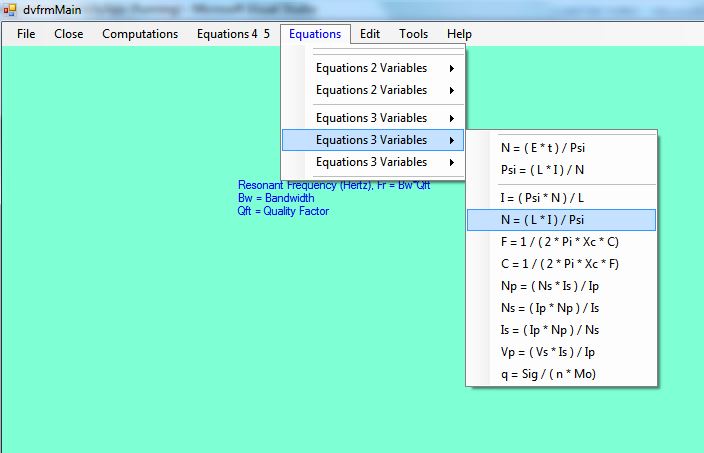


Clicking the Equation menu, six submenu are displayed as; equation\_2 variable, equation\_21 variable, equation\_22 variable, equation\_3 variable, equation\_31 variable and equation\_32 variable as shown in Fig.3.5.

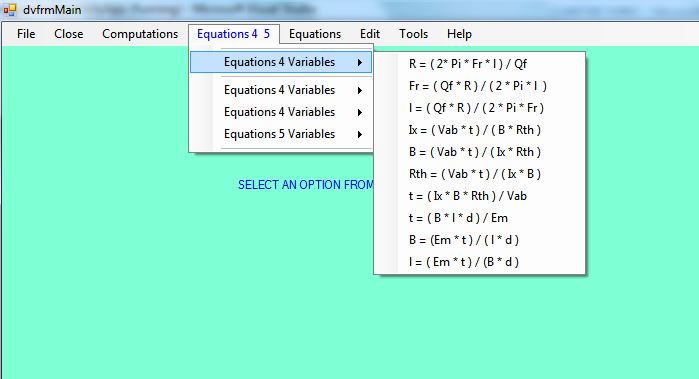


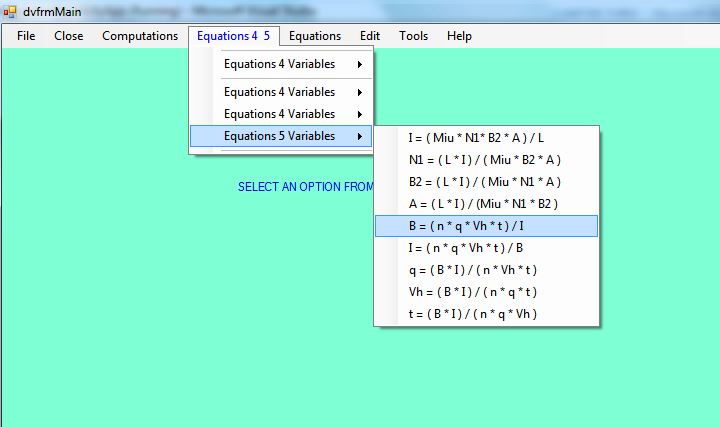
Each of these submenu has another submenus containing 12 equations/models as shown in Fig.3.6. and Fig.3.7, which has been programmed with their respective formula as earlier described in chapter two.



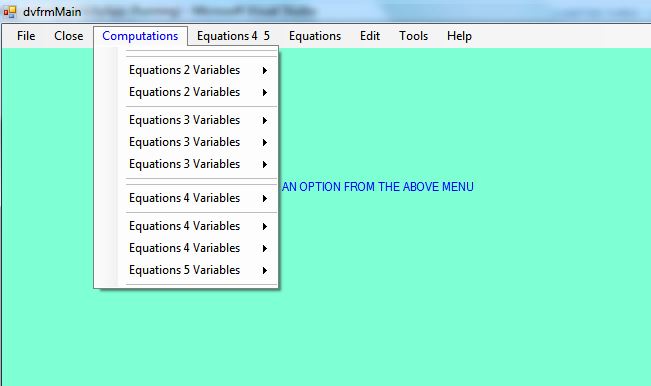


Clicking on each equation, it displays the model and what each parameter of the model represents. Furthermore, clicking on Equation4\_5, it has four (4) submenu and each submenu has another submenu containing 12 equations which comprises of equation four (4) and five (5) as shown in Fig.3.8 and Fig.3.9. (i.e. models with four and five variables). These equations have been programmed to show their parameters and meaning.

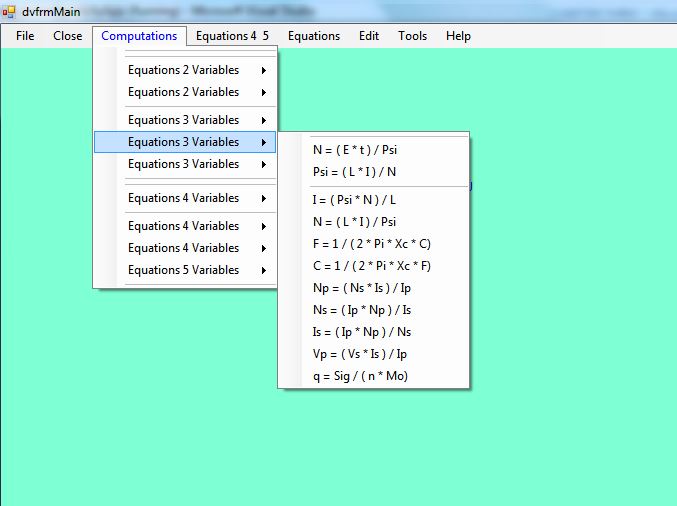




Lastly, the computation menu contains the nine submenus as shown in Fig.3.10 and each submenu contains the electrical models to be computed.



Each submenu comprises of models having two(2) parameters to five(5), which are to be calculated as shown in Fig.3.11.



**3.1 CODING**

The listed electrical models revealed in chapter one were coded using Microsoft Visual Basic.Net 2008, the codes were written in such a way that the entities( such as methods, sub-procedures and function) were repeatedly written for all the electrical models. “Methods”, which is generally declare with the key word ‘public’, is an essential tool in a VB.NET programming language, this basically gives the pattern of displaying these electrical models for a comprehensive display for better understanding of the software to the user, while the “function” is basically declared using a ‘private’ and this is logically used to give a step by step inputting order to the required parameter for each models. These two key words were progressively used in the coding of the whole electrical models in this software package. Here the use Variable is also used for the key folder of an object and an object could be a picture, letter, number, string or anything that form the basis of the programming language, that is, when a variable is declared as object it could hold anything but when declared as double it will not hold as string.

ByVal brings in a copy of the value or the argument, anything done to the copy of the value on programming will be lost when exiting. It retains its original value. ByRef didn’t need referencing of the value, it didn’t retain it original value because the manipulation done would be reflected on it.

A typical example of codes writhen for computing the ohms law is given below;

*Public function dvCalRVdivI*

*dvVoltageIn as object,dvCurrentIn asobject,*

*ByRef dvResultOut as Object,*

*Byref dvWorkingOut as string*

*‘Validate inputs*

*‘initiate variables tobe use*

*dvVar1 = “V”: dvVar2 = “I”: dvLeftside = “Resistance,R”*

*dvExprn =dvVar1 & “/” & dvVar2*

*‘Compute result*

*End function*

Similarly, functions were also written for the electrical models. Each function contains maximum of ten (10) models containing the same set of variables. A typical example of the function codes is written for four equations

Public Function dvCalVar2\_1(ByVal dvValuIn As Object, ByVal dvValuIn2 As Object, ByRef dvResultOut As Object, ByVal dvWkgOut As String, \_

ByVal dvCompType As EnumCalvar2\_1) As String

'Validate Inputs

Select Case dvCompType

Case EnumCalvar2\_1.dvLeq2WI 'L = (2 \* W) / I^2

dvVar1 = "W" : dvVar2 = "I" : dvLeftSide = "Inductance,L"

dvExprn = "(2.0 \*" & dvVar1 & ")/(" & dvVar2 & "\*" & dvVar2 & ")"

Case EnumCalvar2\_1.dvCeq2WV 'C = (2 \* W) / V^2

dvVar1 = "W" : dvVar2 = "V" : dvLeftSide = "Capacitance,C"

dvExprn = "(2.0 \*" & dvVar1 & ")/(" & dvVar2 & "\*" & dvVar2 & ")"

Case EnumCalvar2\_1.dvIeq2WL 'I = Sqrt((2 \* W) / L)

dvVar1 = "W" : dvVar2 = "L" : dvLeftSide = "Current,I"

dvExprn = "(2.0 " \* " & dvVar1 & )/" & dvVar2

dvResultOut = Abs(dvResultOut)

dvResultOut = sqrt(dvResultOut)

Case EnumCalvar2\_1.dvVeq2WC 'V = Sqrt((2 \* W) / C)

dvVar1 = "W" : dvVar2 = "C" : dvLeftSide = "Voltage,V"

dvExprn = "(2.0 " \* " & dvVar1 & )/" & dvVar2

dvResultOut = Abs(dvResultOut)

dvResultOut = sqrt(dvResultOut)

End Select

'Compute Result

Return dcomm

End Function

Functions for other eighty six electrical models are given at appendix 1.0

**3.2 USER INTERFACE DESIGN**

Four forms were designed for this project work from the VB.NET integrated development environment (IDE), the properties of each forms were allocated to suit each purpose. Using the VB.NET Framework environment different number of menus were picked and dropped onto each form. These menus were systematically arranged according to what they are coded to do. Colors of various kinds were also assigned for the attractiveness display of the user interface.

Submenus were also linked to some of the menus of these forms, simply by right clicking the menu to display boxes to add submenus to the main menu that was dropped into the forms that led to the formation variable number of submenus and a distinct name were appropriately given to the both the main menus and submenu according to their properties and the electrical model assigned, atypical example of the user interface is given in the Fig.3.12.

**3.3 NUMERICAL COMPUTATIONS**

In developing this software, an important emphasis was made in adding a unique computation pattern with the output result through a step by step working that is displaying the formula for the electrical model, how the substitution was done and the final result with the unit of the measurement for that electrical model (SI unit).

The step by step working pattern of one of the electrical model is shown below;

Computing Voltage

*V = IR*

*V = 0.5 \* 4*

*V = 2.0v*

This display pattern is repeatedly coded for all the other electrical models

**3.4 PACKING**

The packing of software was immediately done after the compilation and the coding were completed with the available tools that accompany the Microsoft Visual Studio 2010.