COMPONENT TOOLBOX APP DESIGN BOOK

Oluwole Oyetoke

oluwoleoyetoke@gmail.com

Table of Contents

NTRODUCTION	2
EQUIREMENT	2
LASS ANALYSIS	2
CLASS DIAGRAM	3
UI DESIGN SKETCHES	9
IVC INTERACTION	18

INTRODUCTION

R-Toolbox is a resistor value calculator which contains 3 other vital Electronic Engineering tools. It is designed for 4 major purposes which are:

- 1. To help get component information from the Farnell website.
- 2. To serve as a quick tool that can be used to generate resistor values from their colour codes during lab experiments.
- 3. To serve as a quick tool that can be used to generate tantalum capacitor values from their colour codes during lab experiments.
- 4. To help perform spectral analysis of sine, cosine and square wave signals

It was built using Java 8.0 on the NetBeans IDE platform

REQUIREMENT

To run this software, the user has to have an up to date version of the Java Runtime environment (JRE). As we know, Java is a portable programming language which can be run on multiple platforms as long as those platforms have the Java Virtual Machine (JVM) which interfaces with the hardware and operating system on behalf of the software programme. The latest JRE can be downloaded on the Oracle website, here.

The spectrum analysis part of this software performs graph plotting. To achieve this, a standard plotting library was imported. Therefore, to run this software efficiently, the 'Component_Toolbox_App.jar' file must be run in a folder containing another folder called 'lib' which has the 'jmathplot.jar' library in it. See figure below for details

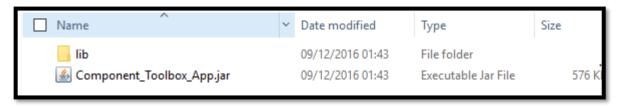


Fig. 1: Snapshot of the Folder Containing the 'Component_Toolbx.App.jar' file

CLASS ANALYSIS

This software is divided into 3 major parts which are the Model, View and Controller (MVC) sections. There are a few other standard classes which are neither serving the view, model nor controller functions. This include the class containing the main method and the function file used to carry out the Fast Fourier Transform for the spectrum plot. The MVC programming technique is carefully implemented across all 13 classes that make up this application. These classes are listed below, accordingly.

Model.java

- AboutView.java
- CapacitorColourBandView.java
- ColourDialogueView.java
- ResistorColourBAndView.java
- ResistorInfoView.java
- MainFrame.java
- ResistorListView.java
- SpectrumAnalyzerView.java
- WelcomeView.java
- Controller.java
- FFT
- Component_Tolbox_App.java

CLASS DIAGRAM

The diagram below shows a detailed Class diagram representation of all the classes that make up the application. This diagram especially shows the relationships and dependencies between each of these classes

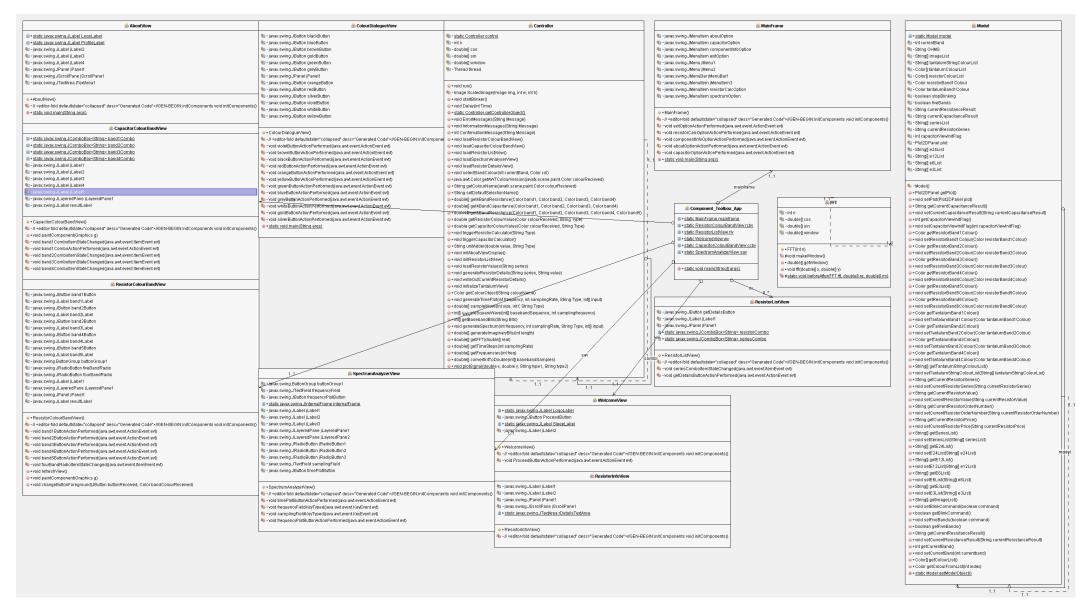
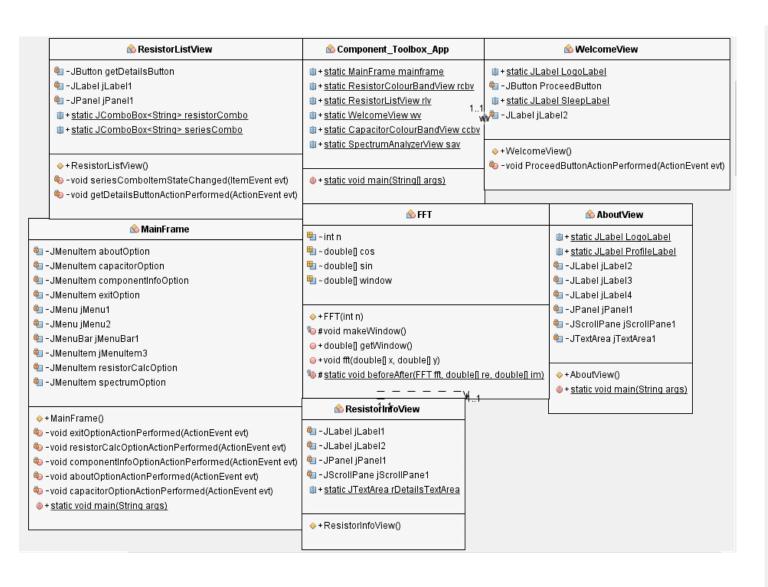


Fig. 2: Snapshot of the R-Toolbox Class Diagram Structure Showing the MVC Dependency

The diagram above is quite compact and may be very difficult to read through. In the light of this, for visibility sake, each of these class diagrams are separately displayed below in a magnified manner



Fig. 3: Magnified Class Diagram 1



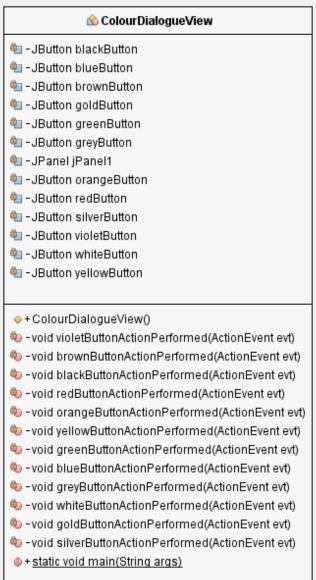


Fig. 4: Magnified Class Diagram 2

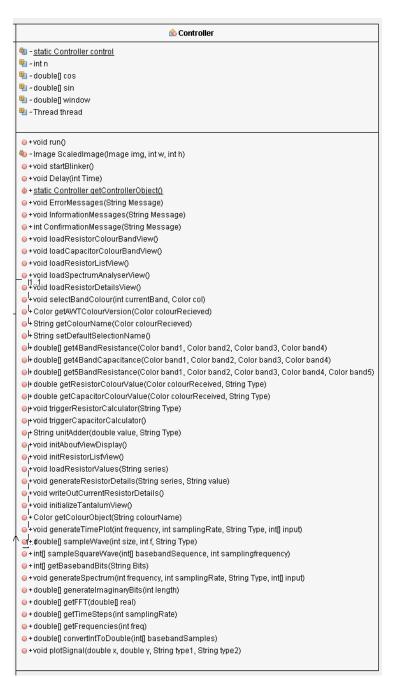


Fig. 5: Magnified Class Diagram 3

** static Model model ** - Int currentBand ** - String O-Max ** - String I mageList ** - String I mageList ** - Colorg I tantalum/ColourList ** - Colorg I tantalum/ColourList ** - Colorg I colour ** - Colour selector Band Colour ** - Color resistor ColourList ** - Colorg resistor ColourList ** - Colour selector Band Colour ** - Subang currentPasalant Resout ** - String currentCapacitanceResut ** - String currentCapacitanceRe	
# - String O-HMS - String I matel.st - String I matel.st - String I matel.st - Color I harbum Colour.List - Color I harbum Colour.List - Color resistorColour.List - Color resistorBand Colour - Color resistorBand Colour - Color List Limit Band Colour - Loole List Limit Band Colour - Loole List Limit Band Colour - String currentResistanceResult - String currentResistanceResult - String currentString List List List List List List List List	
Stringli mageList Stringli matelumStringColourList Colorig tentsumColourList Colorig tentsumColourList Colorig resistorColourList Color colorigum Colorigum	
9L - String] tantalum/String/ColourList 9L - Colorg Tantalum/ColourList 9L - Colorg Tesistor/ColourList 9L - Colorg Tesistor/ColourList 9L - Colorg Tesistor/ColourList 9L - Colorg Tesistor/String Colour 9L - Doolean stop	
Colorij tentalumColourList Colorij ResistorColourList Color resistorColourList Color resistorBand Colour Color stantalumBand1 Colour Color tantalumBand1 Colour Lobolan stopBlinking Lobolan fixeBands String currentNesistancePesult String currentCapacitanceResult String seriesList	
QI - Colorij resistorColourList QI - Colori resistorBand Colour QI - Color tantalumBand Colour QI - boolean stopBlinking QI - boolean stopBlinking QI - boolean fixeBands QI - String current/ResistanceResult QI - String current/ResistanceResult QI - String surrent/SapactanceResult QI - String surrent/SapactanceResult QI - String Servicist	
- Color resistorBand1 Colour - Color tantalumBand1 Colour - Color tantalumBand1 Colour - Looloan stopBlinking - Looloan stopBlinking - Boolean fiveBands - String currentTesistanceResult - String currentCapacitanceResult	
Color tantalumBand1 Colour Lobean stopEllinking Lobean sto	
● - boolean fiveBands - ● - String currentResistanceResult ● - String currentCapacitanceResult ● - Stringl seriesList	
ଷ୍ତା - String currentResistanceResult ଷ୍ତା - String currentCapacitanceResult ଷ୍ତା - String[] seriesList	
ଶ୍ୱା - String currentCapacitanceResult ଶ୍ୱା - String[] seriesList	
🐿 - String[] seriesList	
જ્ય - int capacitorViewInitFlag જ્ય - Plot2DPanel plot	
n - String() e24List - String() e12List	
a - String[] e6List	
a - String[] e3List	
♠ -Model()	
●+Plot2DPanel getPlot()	
+void setPlot(Plot2DPanel plot)	
String getCurrentCapacitanceResult()	
+void setCurrentCapacitanceResult(String currentCapacitanceRe	esult)
o+int getCapacitorViewInitFlag()	
+void setCapacitorViewInitFlag(int capacitorViewInitFlag)	
o+Color getResistorBand1Colour()	
+void setResistorBand1Colour(Color resistorBand1Colour) +Color getResistorBand2Colour()	
+color genesistorBand2Colour() +void setResistorBand2Colour(Color resistorBand2Colour)	
+Void servesistorBand2Colour(Color resistorBand2Colour) +Color getResistorBand3Colour()	
• +void setResistorBand3Colour(Color resistorBand3Colour)	
+Color getResistorBand4Colour()	
+void setResistorBand4Colour(Color resistorBand4Colour)	
+Color getResistorBand5Colour()	
+void setResistorBand5Colour(Color resistorBand5Colour)	
● + Color getResistorBand6Colour()	
+void setResistorBand6Colour(Color resistorBand6Colour)	
+ Color getTantalumBand1 Colour()	
o+void setTantalumBand1Colour(Color tantalumBand1Colour)	
+Color getTantalumBand2Colour()	
+void setTantalumBand2Colour(Color tantalumBand2Colour)	
+ Color getTantalumBand3Colour()	
+void setTantalumBand3Colour(Color tantalumBand3Colour)	
+ Color getTantalumBand4Colour()	
+void setTantalumBand4Colour(Color tantalumBand4Colour)	
+String[] getTantalumStringColourList()	
+void setTantalumStringColourList(String[] tantalumStringColourl	List)
+String getCurrentResistorSeries()	
+void setCurrentResistorSeries(String currentResistorSeries)	
o+String getCurrentResistorValue()	
+void setCurrentResistorValue(String currentResistorValue)	
String getCurrentResistorOrderNumber()	
+void setCurrentResistorOrderNumber(String currentResistorOrder)	ierNum
String getCurrentResistorPrice()	
+void setCurrentResistorPrice(String currentResistorPrice) + Strings actCortool LetA	
String[] getSeriesList() Avoid petSeriesList(Ptring[] period int)	
+void setSeriesList(String[] seriesList) +String[] getE24List()	
+string[] getE24List() +void setE24List(String[] e24List)	
•+voia setE24List(Stringtj e24List) •+Stringt] getE12List()	
+string[] getE12List() +void setE12List(String[] e12List)	
+ String() getE6List()	
•+oid setE6List(String[] e6List)	
String[] getE3List()	
• +void setE3List(String[] e3List)	
String[] getImageList()	
•+void setBlinkCommand(boolean command)	
+boolean getBlinkCommand()	
+void setFiveBands(boolean command)	
+ boolean getFiveBands()	
String getCurrentResistanceResult()	
+void setCurrentResistanceResult(String currentResistanceResult)	ult)
e+int getCurrentBand()	
+void setCurrentBand(int currentband)	
+Color[] getColourList()	
+Color getColourFromList(int index)	
+ static Model getModelObject()	

GUI DESIGN SKETCHES

This software has 9 views in total, and these views are sketched out in the diagrams below

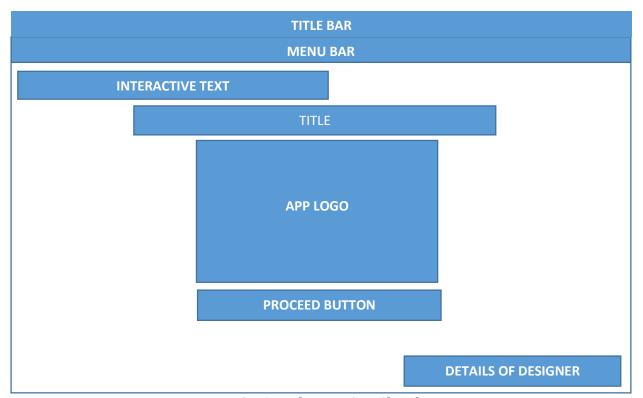


Fig. 6: Welcome View Sketch

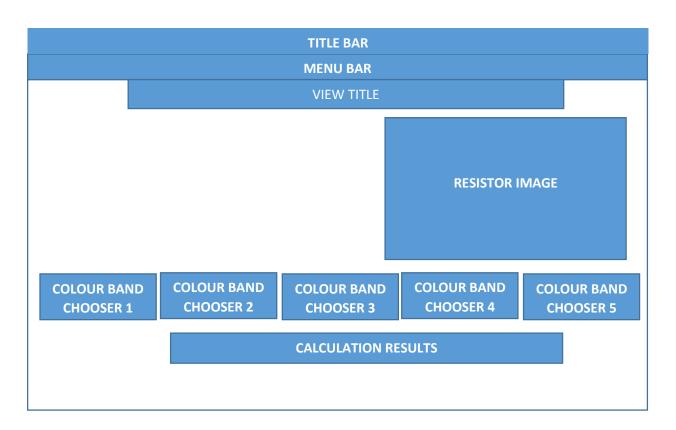


Fig. 7: RESISTOR COLOUR BAND CALCULATOR VIEW SKETCH

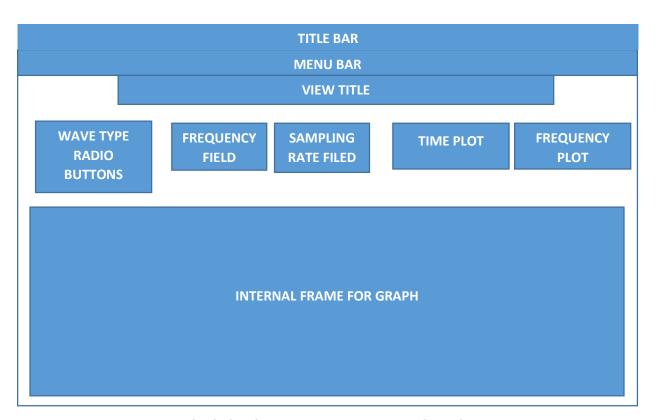


Fig. 8: SPECTRUM ANALYZER VIEW SKETCH

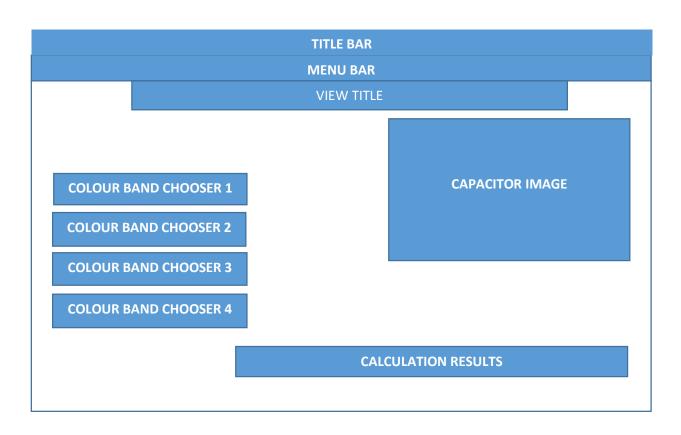


Fig. 9: CAPACITOR COLOUR BAND CALCULATOR VIEW SKETCH



Fig. 10: RESISTOR INFORMATION VIEW SKETCH



Fig. 11: RESISTOR LIST VIEW SKETCH

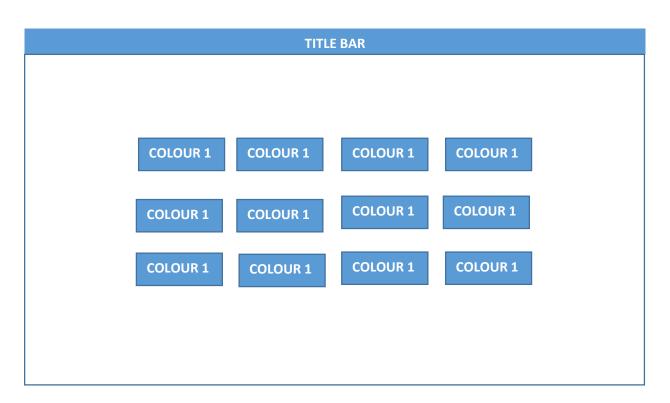


Fig. 12: RESISTOR COLOUR PICKER DIALOGUE VIEW SKETCH



Fig. 13: MAINFRAME VIEW SKETCH

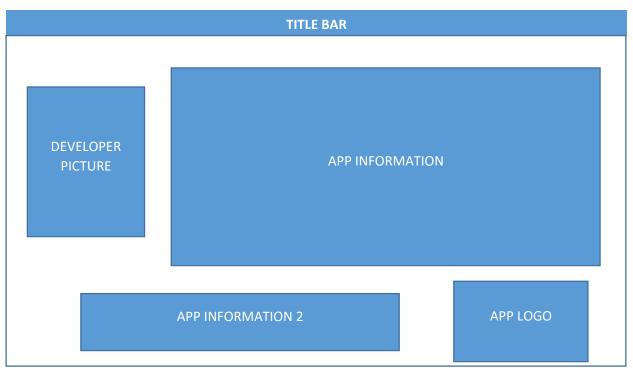


Fig. 14: ABOUT VIEW SKETCH

MVC INTERACTION

The App has only one model and also one controller class which services all 9 views with the needed functions and storage information. Fig. 15 below graphically depicts the relationship between the Model, View and Controller in this app.

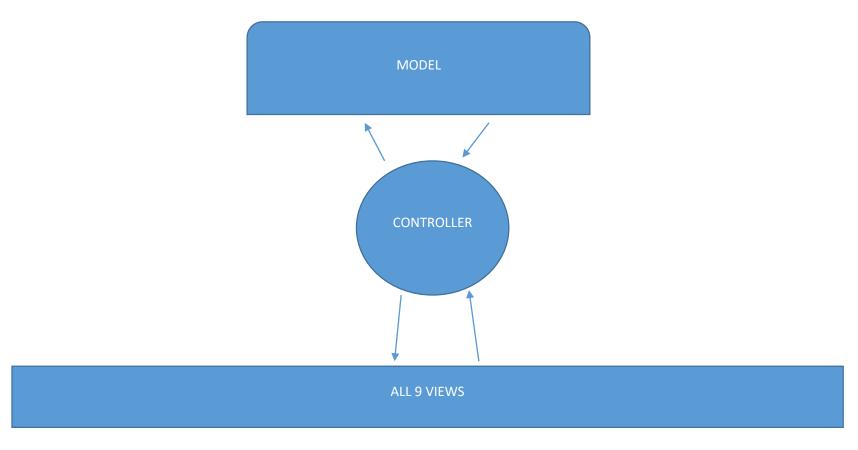


Fig. 15: Diagram explaining the MVC interaction Existing Between All of the Apps Components