COMPONENT TOOLBOX APP DESIGN BOOK

Oluwole Oyetoke

oluwoleoyetoke@gmail.com

Table of Contents

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.

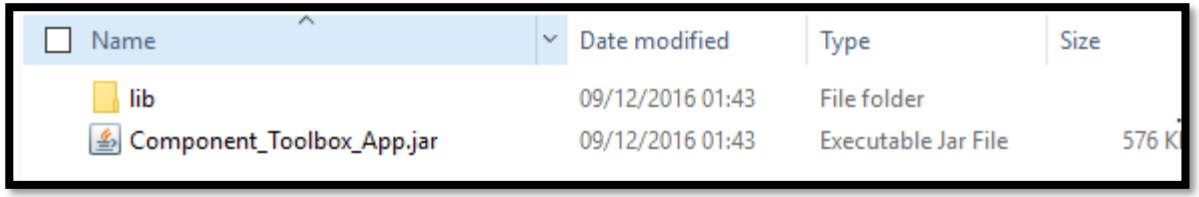
4. To help perform spectral analysis of sine, cosine and square wave signals

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



Name	Date modified	Type	Size
lib	09/12/2016 01:43	File folder	
Component_Toolbox_App.jar	09/12/2016 01:43	Executable Jar File	576 K

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

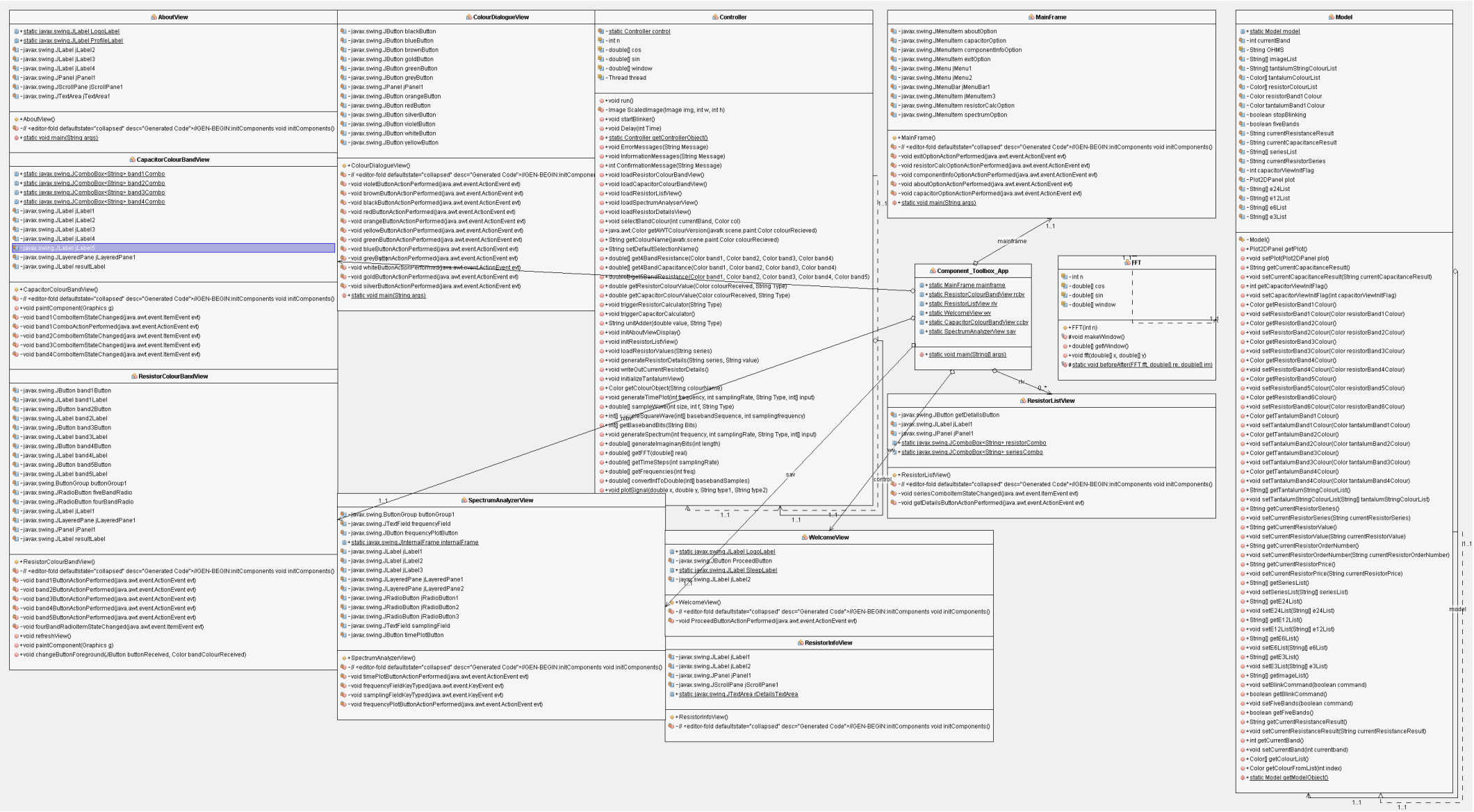


Fig. 2: Snapshot of the R-Toolbox Class Diagram Structure Showing the MVC Dependency

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

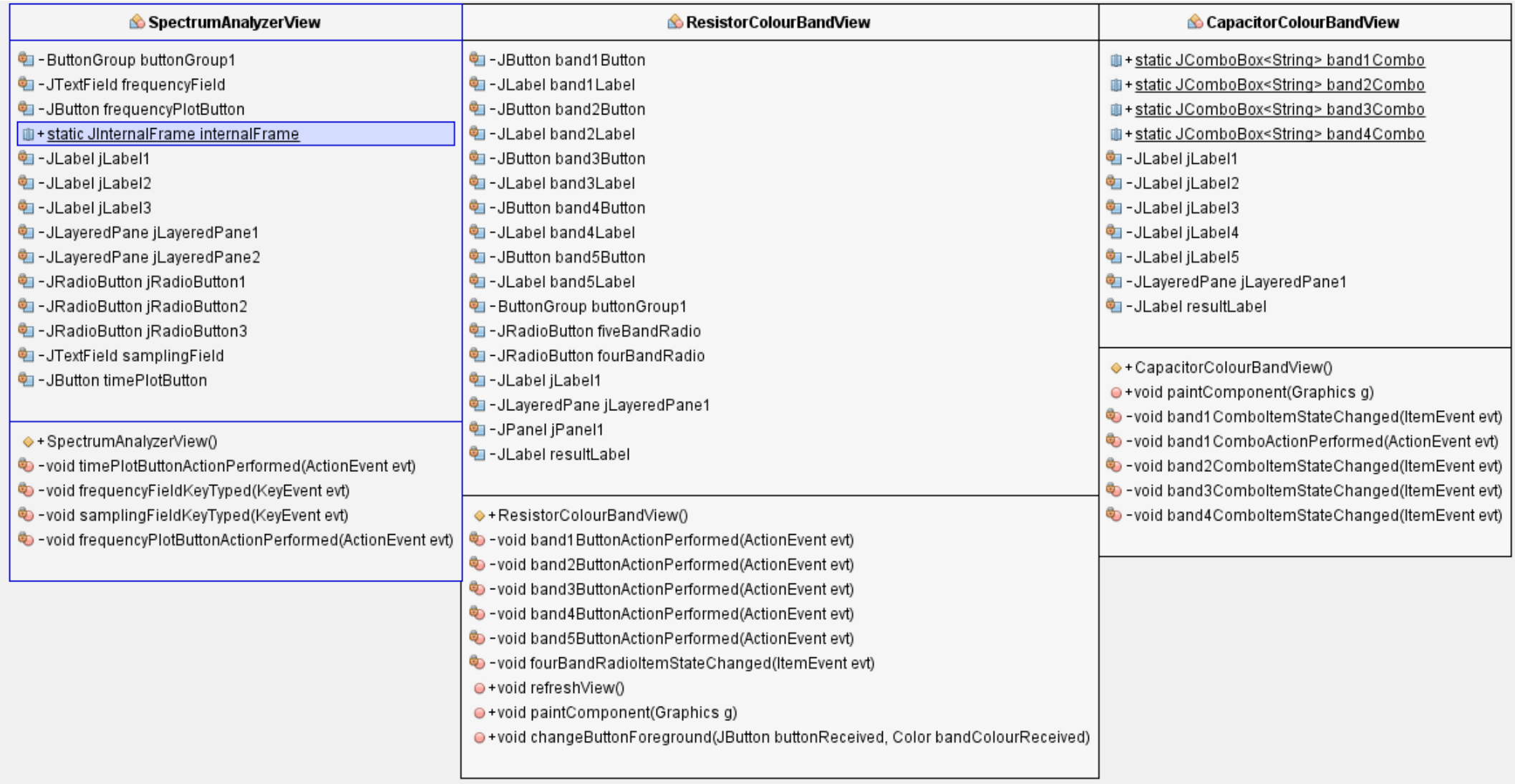
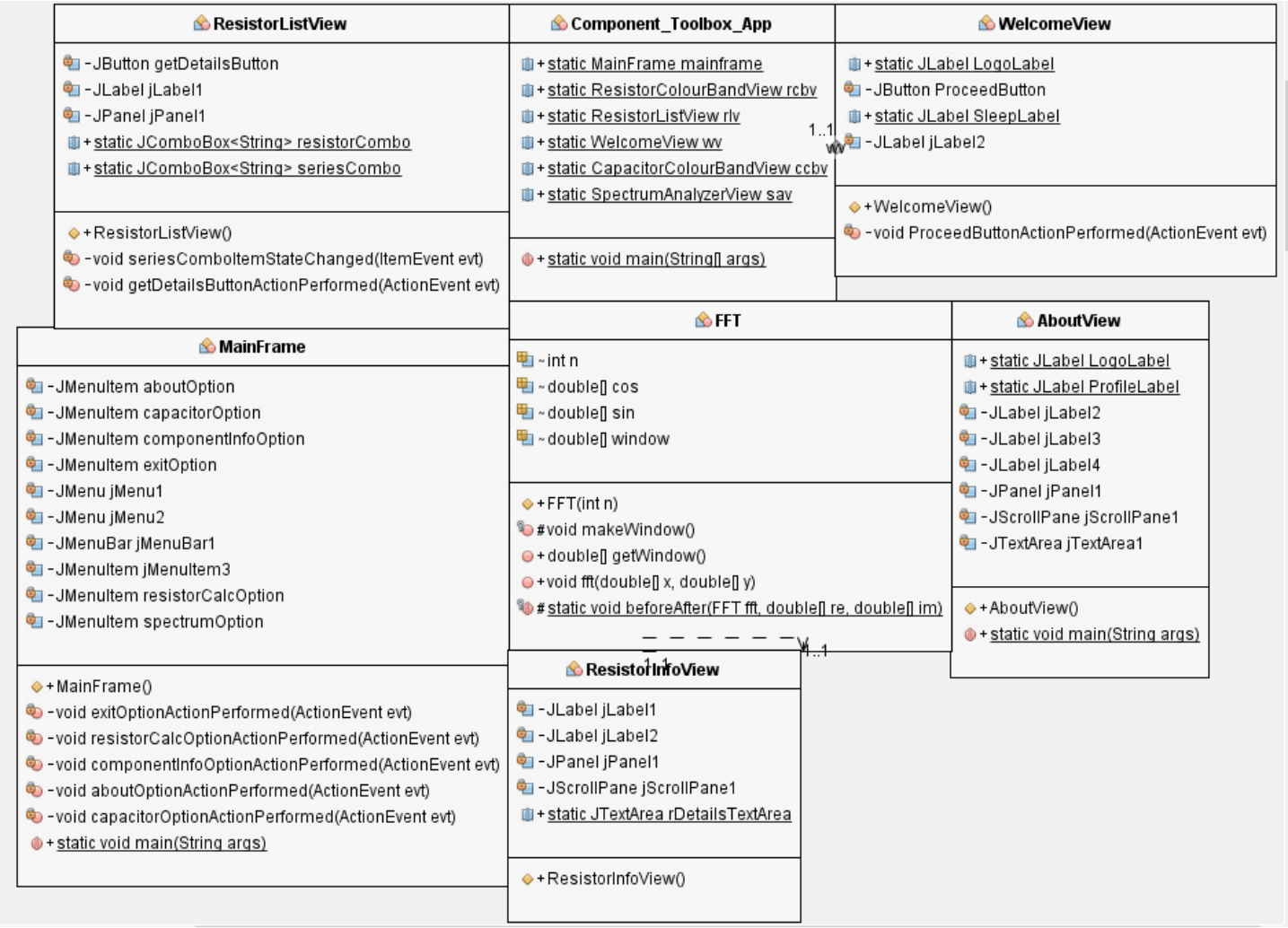


Fig. 3: Magnified Class Diagram 1



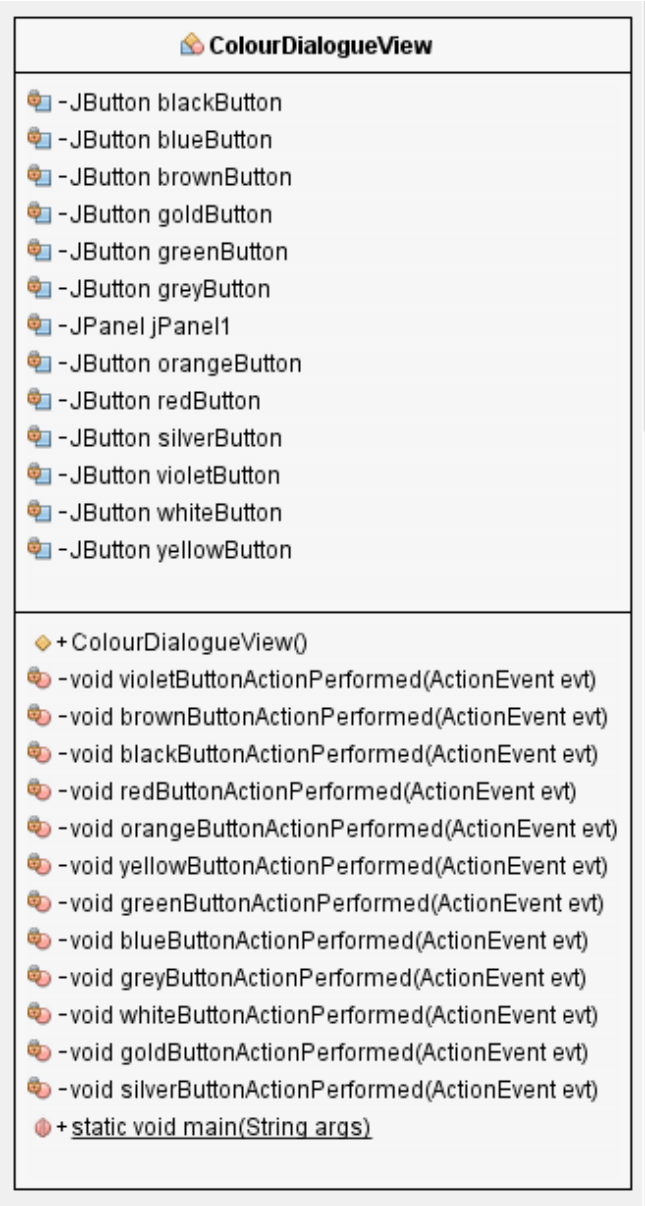


Fig. 4: Magnified Class Diagram 2

Fig. 5: Magnified Class Diagram 3

GUI DESIGN SKETCHES

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

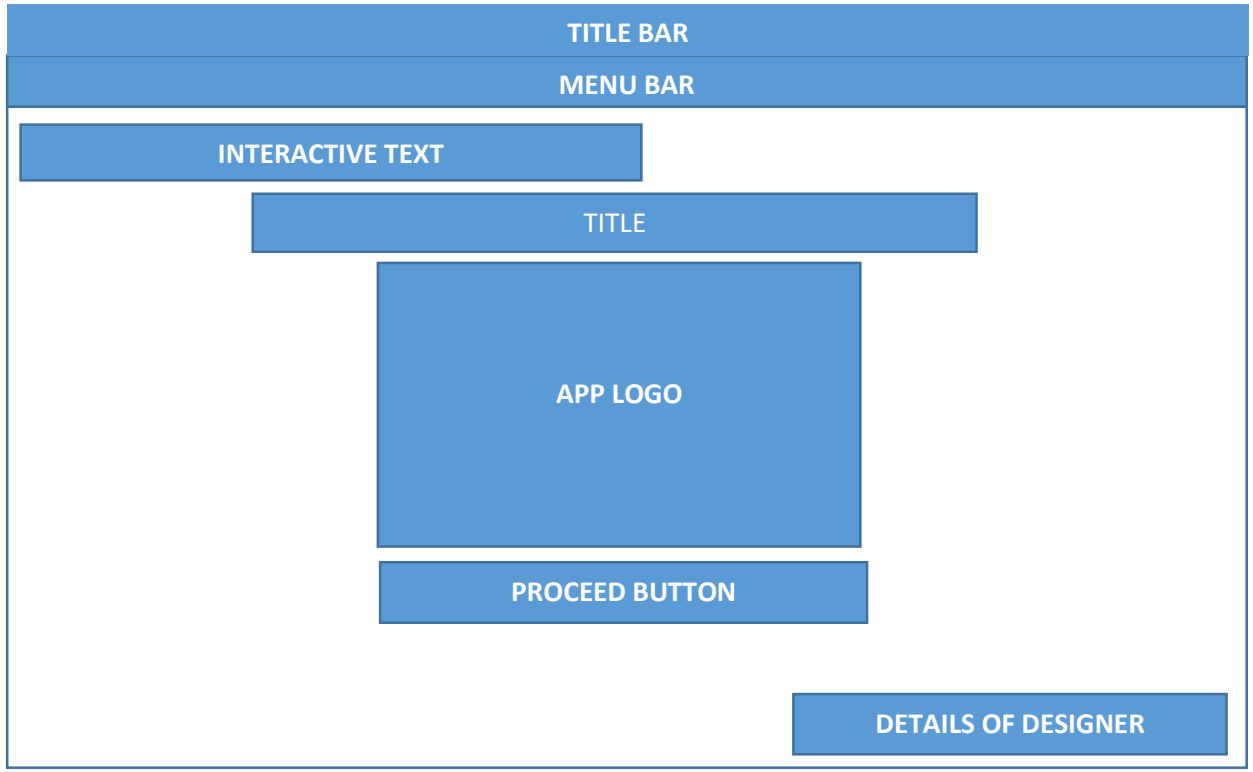


Fig. 6: Welcome View Sketch

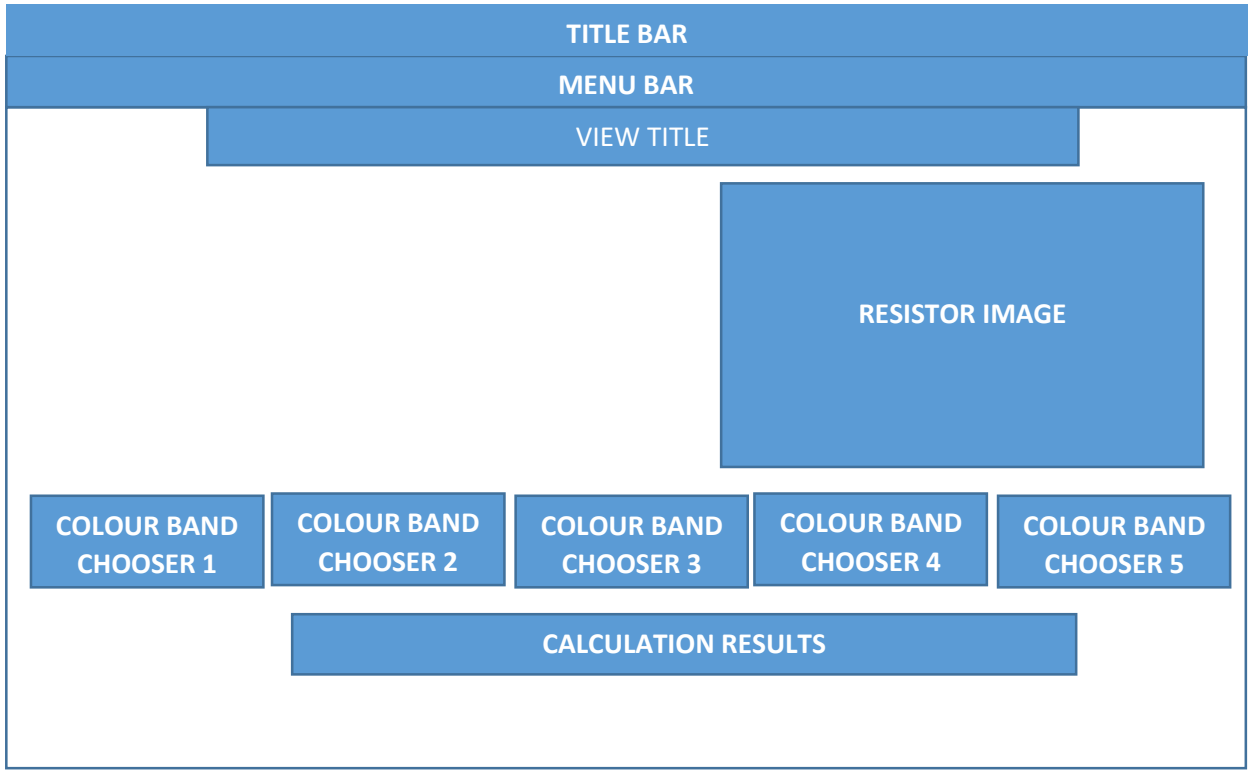


Fig. 7: RESISTOR COLOUR BAND CALCULATOR VIEW SKETCH

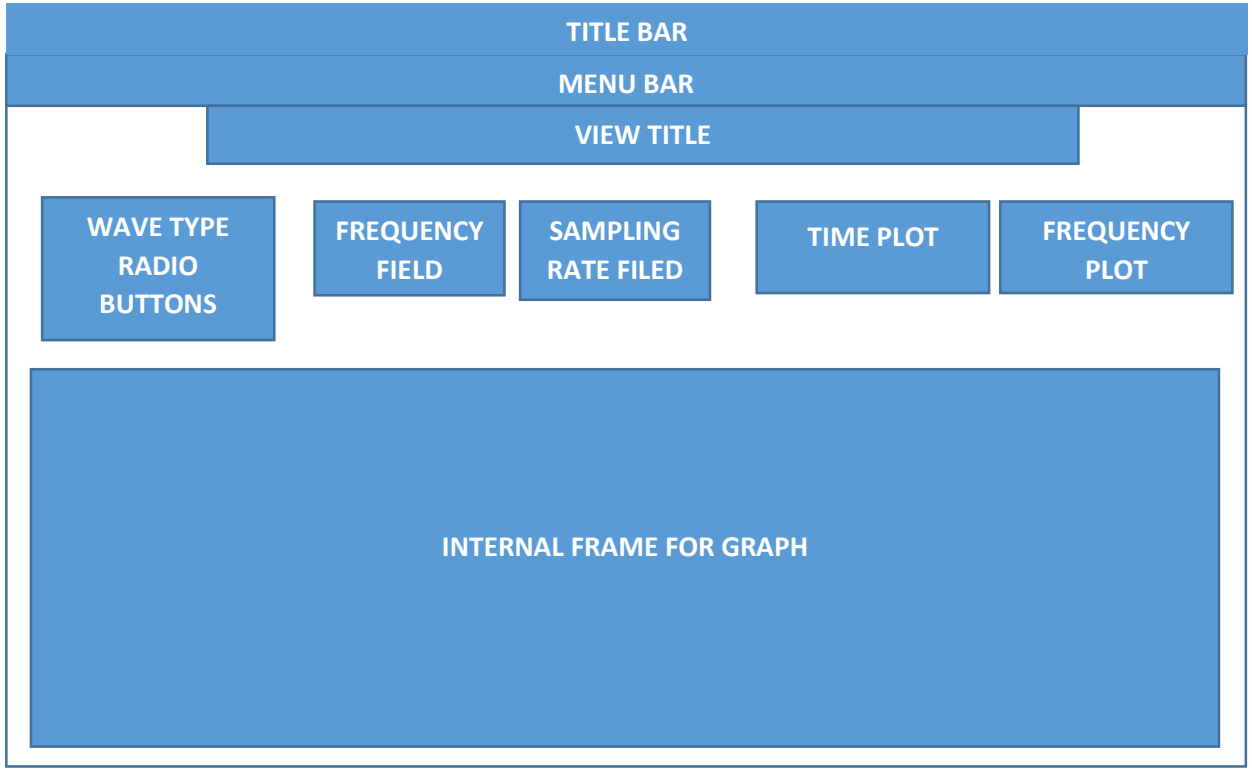


Fig. 8: SPECTRUM ANALYZER VIEW SKETCH

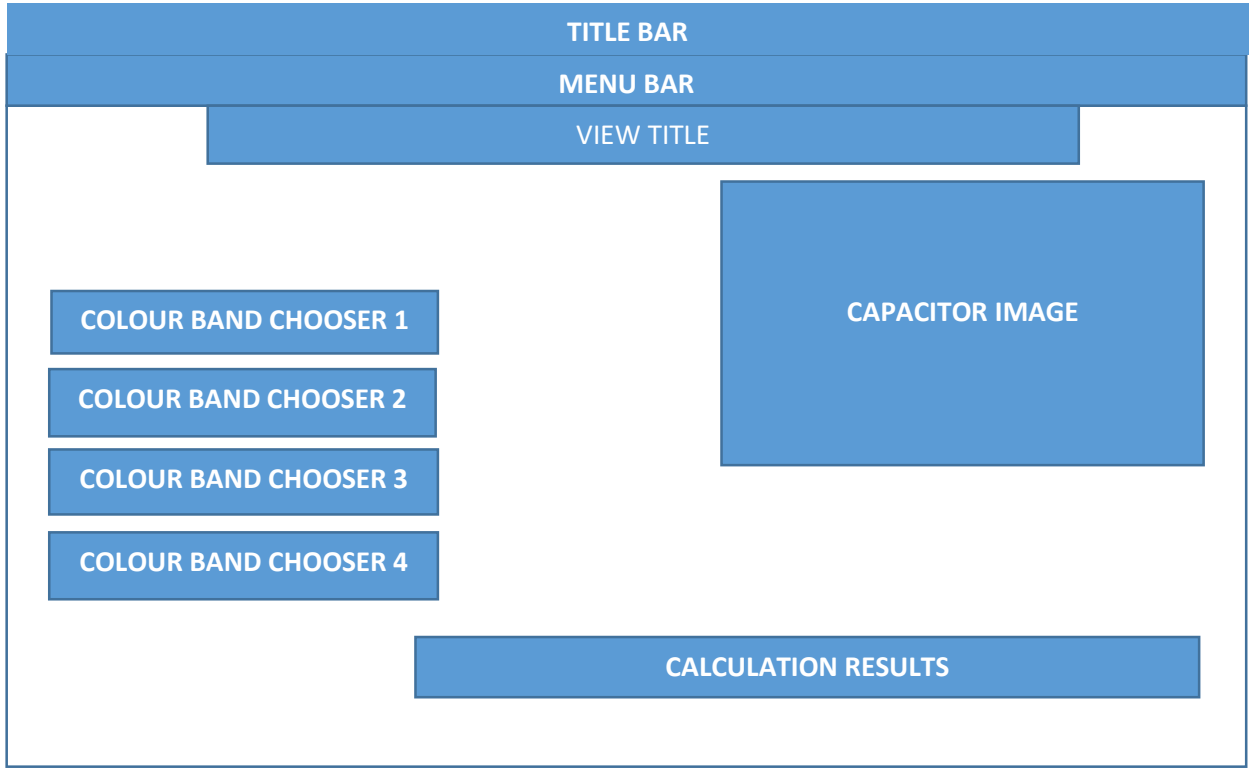


Fig. 9: CAPACITOR COLOUR BAND CALCULATOR VIEW SKETCH

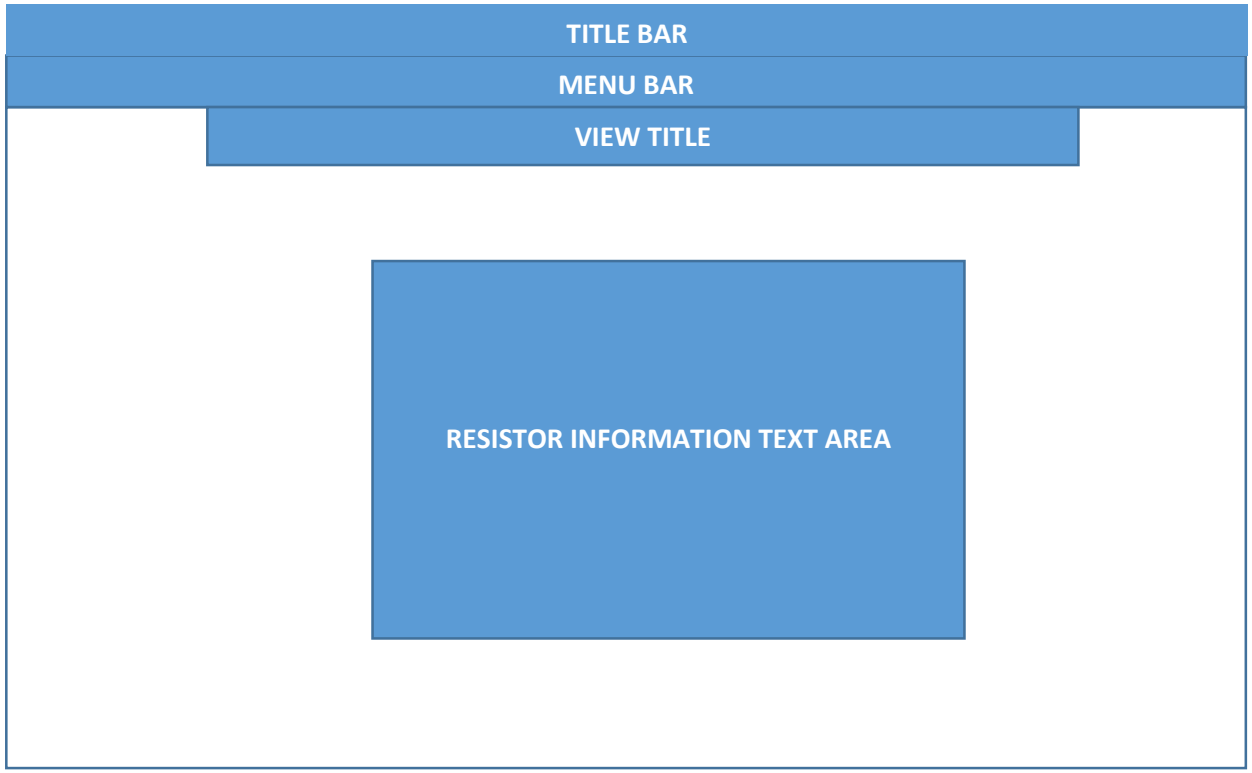


Fig. 10: RESISTOR INFORMATION VIEW SKETCH

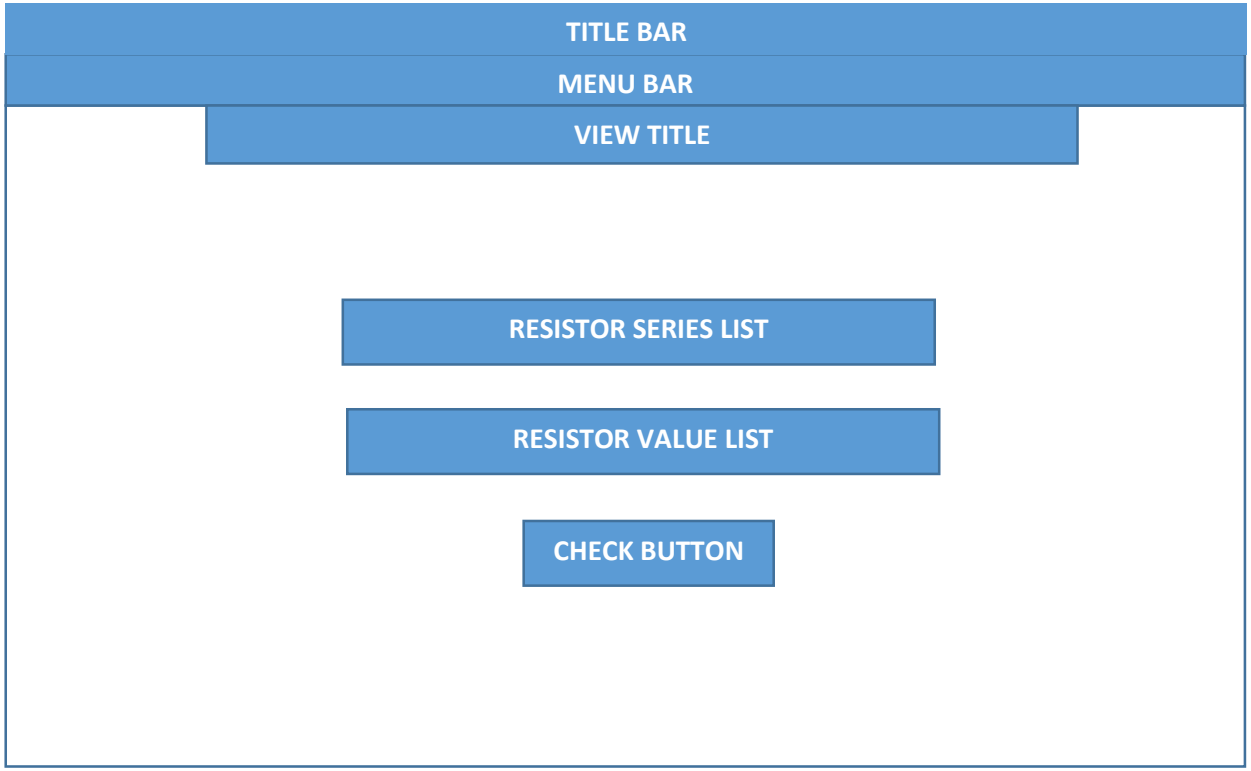


Fig. 11: RESISTOR LIST VIEW SKETCH

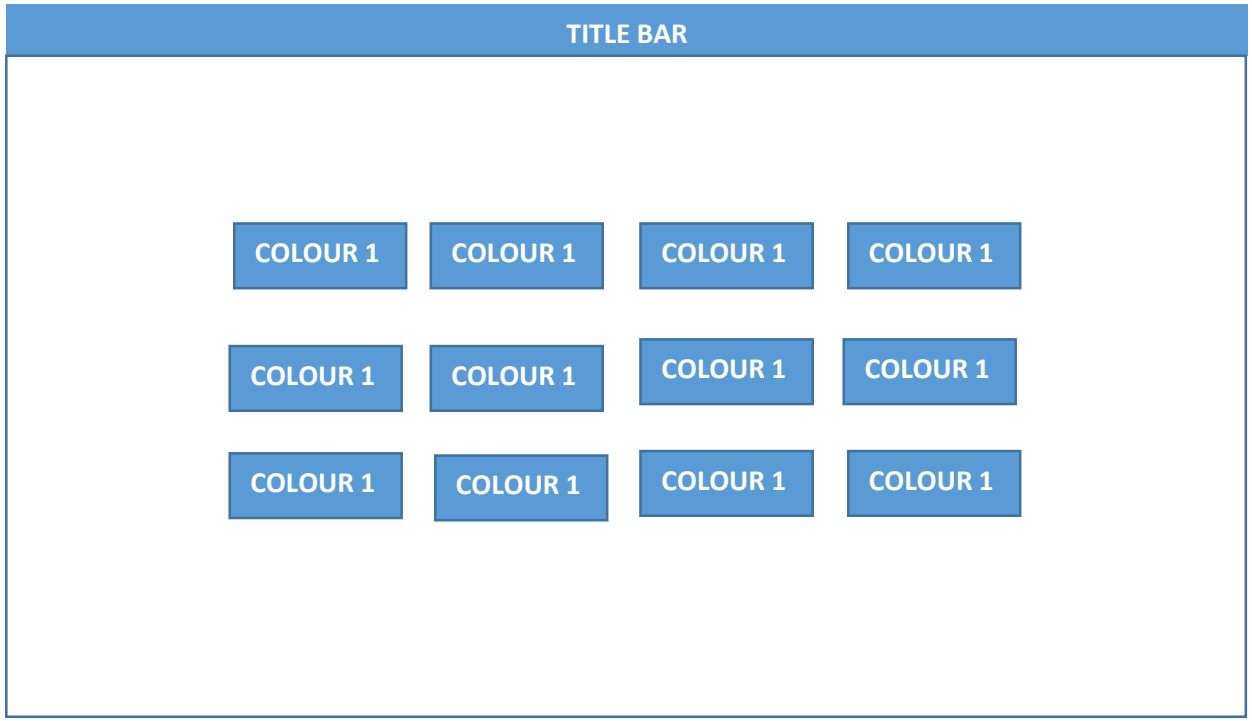


Fig. 12: RESISTOR COLOUR PICKER DIALOGUE VIEW SKETCH

Fig. 13: MAINFRAME VIEW SKETCH

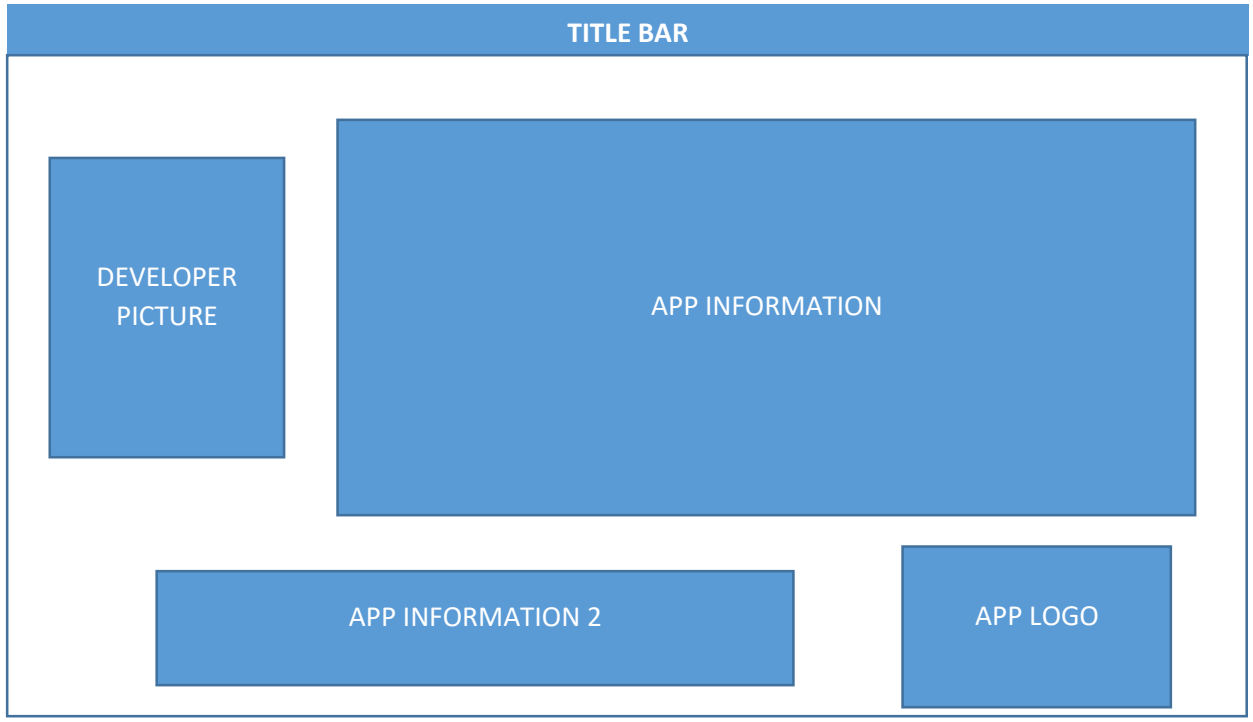


Fig. 14: ABOUT VIEW SKETCH

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.

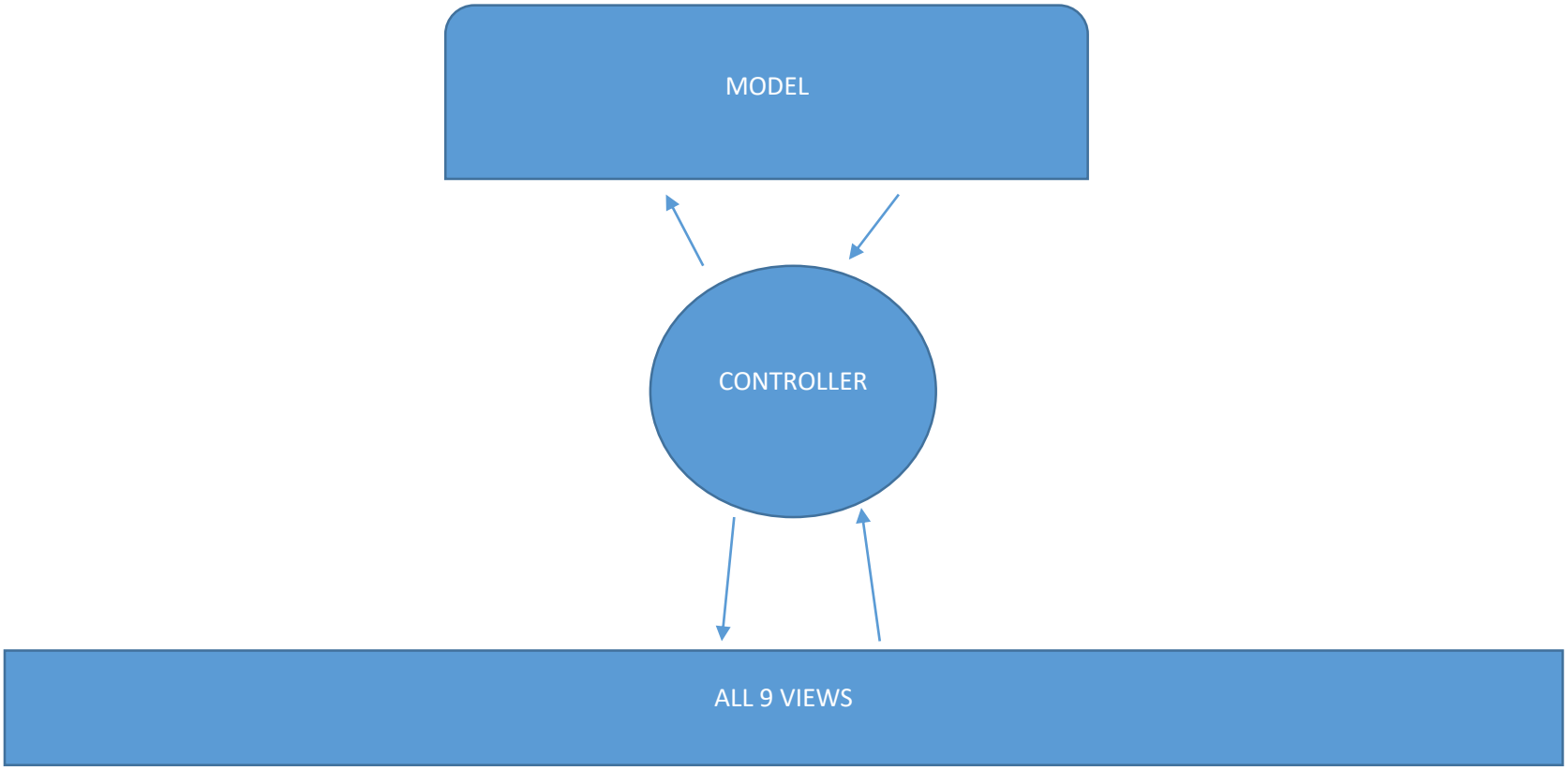


Fig. 15: Diagram explaining the MVC interaction Existing Between All of the Apps Components