**Software Implementation and Testing Document**

**For**

**“Resistor Calulator”**

Version 1.0

**Authors**:

Vinzce Yadao

Ronald Nazaire

Jorge Atencio

Jacob Hobson

Schmidt Jean

# Programming Languages (5 points)

*List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).*

We will be using object-oriented programming.

The programming language that we will be using is java all throughout the project except for the login. For login features SQL will be used. The reason we chose java is because it will be easier to implement the UI and we used SQL for the login because that is what is commonly used.

# Platforms, APIs, Databases, and other technologies used (5 points)

*List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).*

We will be using Github, SQL, IntelliJ IDEA, command prompt, Notepad++.

Github is used to upload all the work that we have done. SQL is used for all login related features. IntelliJ IDEA will be used to write the java code. The command prompt and Notepad++ will be backups just in case some members won’t be able to get IntelliJ IDEA to download or to work. Notepad++ might have some issues but it has worked before for a group member. Xamp as a local webserver with PHP.

# Execution-based Functional Testing (10 points)

*Describe how/if you performed functional testing for your project (i.e., tested for the* ***functional requirements*** *listed in your RD).*

User Log in: Made sure that logging in worked. Messed around with some values, logged out, then logged back in to make sure that the values were being saved.

Multi-band Implementation: So far only supports 4 band resistors. Tested by using test calculations of different bands.

# Execution-based Non-Functional Testing (10 points)

*Describe how/if you performed non-functional testing for your project (i.e., tested for the* ***non-functional requirements*** *listed in your RD).*

We tested to make sure that when we wrote the registration page, we made sure that the password was encrypted so that you can’t see the password being typed in. Password being encrypted has been tested and is successful. We are still working on implementing the three failed password attempts.

# Non-Execution-based Testing (10 points)

*Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).*

Every person that wrote code has tested the code that they wrote and made sure that it’s working and is doing what it is supposed to do. So far, our code has not been reviewed or inspected by a team but will plan to get non-execution-based testing done before iteration 3.