SE 3XA3: Test Plan
Ohm: Resistor Scanner

Team 4, ohm
Jonathan Brown, brownjs2
Graeme Crawley, crawleg
Ryan Marks, marksr2

October 31, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
Date 1	1.0	Notes
Date 2	1.1	Notes

1 General Information

1.1 Purpose

This document will describe the testing procedure used to ensure the correct functionality of Group 4's 3XA3 Project, Ohm. While the implementaion of the project is not complete, it is important to have plan tests that verify that the project complies with the specifications and requirements set out in the SRS. These tests are necessary in order to produce a high quality end product, as well as track and manage the progress of the group.

1.2 Scope

The tests prescribed in the test plan should verify the efficacy of the resistor band detection, the colour selection and the resistor body detection (note: the resistor body detection has not yet been implemented).

1.3 Acronyms, Abbreviations, and Symbols

Table 2: **Table of Abbreviations**

Abbreviation	**Definition**
Abbreviation1	Definition1
Abbreviation2	Definition2

Table 3: **Table of Definitions**

Term	**Definition**
Term1	Definition1
Term2	Definition2

1.4 Overview of Document

The Test Plan will contain five main sections excluding this introductory one, as well as an appendix.

2 Plan

2.1 Software Description

Ohm will allow users with a desktop computer or smartphone equipped camera to determine the resistance of a standard 4-band resistor by placing it within the camera frame. The software will, using Opencv, detect and read the colour bands of the resistor automatically to determine it's resistance. This software will be implemented in Java.

2.2 Test Team

All members of Group 4, Jonathan Brown, Graeme Crawley, and Ryan Marks will be responsible for testing components of the application as well as the application as a whole.

2.3 Automated Testing Approach

2.4 Testing Tools

JUnit 4 will be the tool for testing individual modules of the software.

2.5 Testing Schedule

INTSERT TABLE HERE
See Gantt Chart at the following url ...

3 System Test Description

3.1 Tests for Functional Requirements

3.1.1 Area of Testing1

Title for Test

1. test-id1
   Type: Functional, Dynamic, Manual, Static etc.
   Initial State:

Input:

Output:

How test will be performed:

2. test-id2
   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

3.1.2 Area of Testing2

...

3.2 Tests for Nonfunctional Requirements

3.2.1 Area of Testing1

Title for Test

1. test-id1
   Type:

   Initial State:

   Input/Condition:

   Output/Result:

   How test will be performed:

2. test-id2
   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

3.2.2 Area of Testing2

...

4 Tests for Proof of Concept

4.1 Band Identification

Title for Test

1. PC-BI-1

   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

2. test-id2
   Type: Functional, Dynamic, Manual, Static etc.

   Initial State:

   Input:

   Output:

   How test will be performed:

4.2 Area of Testing2

...

5 Comparison to Existing Implementation

6 Unit Testing Plan

6.1 Unit testing of internal functions

6.2 Unit testing of output files

7 Appendix

This is where you can place additional information.

7.1 Symbolic Parameters

The definition of the test cases will call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance.

7.2 Usability Survey Questions?

This is a section that would be appropriate for some teams.