Table 1: Revision History

Date	Developer(s)	Change
2016-09-26	Ryan	Created template with section headings
...	...	...

SE 3XA3: SRS
ohm Resistor Scanner

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

1 Project Drivers

1.1 The Purpose of the Project

The purpose of this project is to re-implement the open source package "AndroidResistanceScanner" by allowing a user to scan a standard resistor and output the resistance value to the screen, and then build upon that previously existing application by creating additional features and putting a greater emphasis on formal docuementation.

1.2 The Stakeholders

The main stakeholders for this application are the development team and our eventual users. However there are a number of different use cases for this application, so we will divide our users into the broad demographics of electronics beginners, colorblind hobbyists, and power users. Beginners are people new to electronics who dont know how to read resistor color codes and dont yet own a resistance meter to allow for measuring resistance rather than reading it. Another group that has difficulty reading resistor color codes are colorblind hobbyists. This particular set of people would benefit greatly as without this application they would have to rely on the word from those who are not color blind, or from the matching of shades. Finally power users who want to quickly find a resistor without manually searching, possibly with other use cases.

2 Project Constraints

2.1 Mandated Constraints

2.2 Naming Conventions and Terminology

2.3 Relevant Facts and Assumptions

3 Functional Requirements

3.1 The Scope of the Work

3.1.1 The Current Situation

Electronics hobbyists will use resistors in almost any project they encounter. Currently there are a few ways they can identify the resistance of the resistors, these methods are as follows:

- Manual Method

  The resistor colour code was developed to allow the ease of reading for people. With practise, people can quickly identify resistors by the colour bands, however this is less helpful when searching for a specific resistor. Additionally building proficiency takes time and can be a frustrating slowdown for a beginner, especially if they misidentify components during assembly.

- Organizational Method

  By storing resistors according to their resistances it becomes a trivial exercise to find a resistor of a given resistance. This method is clearly superior in professional environments where very large numbers of resistors are on hand.

- Meter Method

  The current technique of choice for individuals who cannot discern colour codes is to measure a given resistor with a resistance meter. This technique is immediately faster than reading colour bands, however it is somewhat laborious to have to pick up and test individual components.

3.1.2 The Context of The Work

This work has a very limited context, it serves as a tool to any person using colour band resistors who feels it useful. There are not other subject matter domains that need to be understood or integrated with, beyond the resistor colour codes.

3.1.3 Work Partitioning

This work does not need to respond to any external business events, this section is not relevant

3.2 Business Data Model & Data Dictionary

3.3 The Scope of the Product

3.3.1 Product Boundary

This project is a simple tool, with the user as the only actor. As such the use case diagram is fairly minimal.

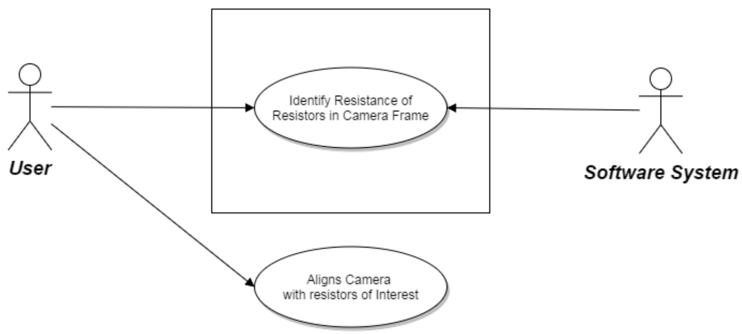

Figure 1: The use case diagram for the system to be showing the product boundary.

3.3.2 Product Use Case List

As our products usability improves between phases of functionality, the use case will expand. A use case for each phase is provided.

Phase 1 Use Case

A user launches the app, and seeks to identify a specific resistor. They hold the resistor up to the camera and aligns the primary axis resistor with an on screen line. The system is constantly scanning, and once a coherent pattern of color bands is identified, the resistance is calculated and displayed. Once the system can no longer identify a coherent pattern of color bands, the displayed resistance value dissappears, until a new resistor is identified.

Phase 2 Use Case

A user launches the app, and seeks to identify a specific resistor. The user points the camera in the general direction of the resistor, with little care for accurate framing. The system is constantly scanning, and once a coherent pattern of color bands is identified, the resistance is calculated and displayed. Once the system can no longer identify a coherent pattern of color bands, the displayed resistance value dissappears, until a new resistor is identified.

Phase 3 Use Case

A user launches the app, and seeks to find a specific resistor in a group. The user points the camera at the group of resistors. The system is constantly scanning, and for any coherent pattern of color bands identified, the resistance is calculated and displayed in frame next to the resistor.

3.4 Functional Requirements

4 Non-functional Requirements

4.1 Look and Feel Requirements

1. The application shall have a very minimal user interface focussed primarily around the view of the camera.

4.2 Usability and Humanity Requirements

1. The application shall be easily usable by people aged 8 to 70.
2. The application shall be suitable for a user with a minimal understanding of resistor colour codes and very little training.
3. The application shall not require the understanding of any particular language as all options symbolically.
4. The application shall be usable by those with colour blindness.

4.3 Performance Requirements

4.3.1 Speed and Latency Requirements

1. The application should launch in no more than 5 seconds.
2. The algorithm used to process the income stream of images should be able to process between 15 and 30 images per second on a standard mobile phone.

4.3.2 Precision or Accuracy Requirements

1. The application should identify an input as the correct class of resistor 95% of the time.

4.3.3 Reliability and Availability Requirements

1. The application will not require internet access.

4.4 Operational and Environmental Requirements

4.4.1 Expected Physical Environment

1. The environment for the use of the application is in a household (hobyist) or light industrial setting.

4.4.2 Productization Requirements

1. The application will be bundled as a .jar file for the desktop and as a .apk file for the Android platform.

4.4.3 Release Requirements

1. The project will feature one pre-release demo with two full releases scheduled for November and December.

4.5 Maintainability and Support Requirements

1. Once the product is launched, no further maintenance is required unless there is a desire for features to be added.
2. No live support for this product is necessary.
3. The application will support execution on the desktop and on Android.
4. The application might eventually be supported on iOS.

4.6 Security Requirements

The application does not possess any security concerns.

4.7 Cultural Requirements

The application does not feature any requirements specific to certain cultures outside of the usability requirements listed above.

4.8 Legal Requirements

1. The application should disclaim that it is not for use in safety critical systems.

5 Project Issues

5.1 Open Issues

5.2 Off-the-Shelf Solutions

5.3 New Problems

5.4 Tasks

5.5 Migration to the New Product

5.6 Risks

5.7 Costs

The project contains no components that pose any cost to the development team.

5.8 User Documentation and Training

5.9 Waiting Room

5.10 Ideas for Solutions