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

Constraints for this project can be broken into two categories; developer constraints and device constraints. Developer constraints are constraints that pertain to the development of the application while device constraints are con-

straints that pertain to the user's device and what is required to use the application.

2.1.1 Developer Constraints

Developer constraints consist of the due date for the application which is December 7, 2016, the requirement that the application must be a re-implementation of a previously existing application on both Android and Windows, and the requirement that the application must be built at no cost to the developers.

2.1.2 Device Constraints

Device constraints include constraints on the hardware of the device and the resistors being scanned. Hardware constraints include the system having a camera and the CPU being fast enough to run the required algorithms. The resistors must be tan in colour and placed on a flat, uniformly coloured surface.

2.2 Naming Conventions and Terminology

1. Resistor: Beige coloured, axial lead resistor

2. Device: The hardware running the application software

3. User: Any person who is using the final product to scan resistors

4. AndroidResistorScanner: The name of the open-source resistor scanning program

2.3 Relevant Facts and Assumptions

It is assumed that the user knows the function of a resistor and understands how resistors are measured. It is also assumed that the user can see the text on the output screen. It is not assumed, however, that the user could read a resistor given a resistor chart.

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 practice, 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

There is not a business domain to be modeled by this software, as such these sections are not needed.

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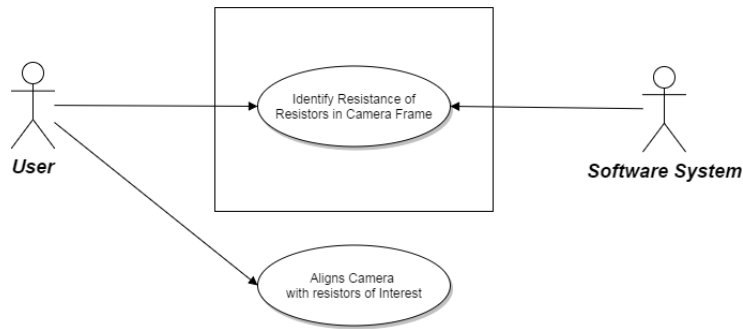
Figure 1: The use case diagram for the system to be showing the product boundary.

3.3.2 Product Use Cases

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 disappears, 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 disappears, 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

The functional requirements format given in the Volere template seems best suited to broad applications with many stakeholders and unclear development order. For this document only the description, rationale, and fit criteria for a requirement will be given. The requirements will be divided by the phases of development as meeting the goals of a given phase hinges on the completion of

the ones before it. There will not be a further prioritization of features beyond this phasing.

3.4.1 Phase 1 Functional Requirements

Description: Display text over video stream
Rationale: To meet the phase 1 use case, the resistance of the scanned resistor must be displayable.
Fit Criterion: The criterion is exceedingly straightforward, if text can be displayed over the video stream, the requirement is met

Description: Identify the resistance of a resistor, two points indicating its primary axis.
Rationale: In Phase 1 a line will be displayed over the video stream indicating where to position the resistor. For greater modularity, and to allow for future phases to build on the source code, the scanning algorithm should work with an arbitrary defined or identified primary axis.
Fit Criterion: This requirement shall be validated by testing it against human categorized resistors with a human annotated primary axis. It shall have an accuracy as defined in the non-functional requirements.

3.4.2 Phase 2 Functional Requirements

Description: Identify the primary axis of a resistor arbitrarily positioned in an image.
Rationale: As is stated in the Phase 2 use case, we must be able to identify a resistor in any legible orientation. The recognized primary axis shall be the same format and structure as is consumed by the resistance identification system so the two can be seamlessly integrated.
Fit Criterion: This will be verified by running the algorithm against the human annotated primary axis.

3.4.3 Phase 3 Functional Requirements

Description: Identify the primary axis of many resistors arbitrarily positioned in an image.
Rationale: This is the necessary to meet the phase 3 use case that is not developed in an existing phase
Fit Criterion: This will be tested against a number of human annotated examples.

Description: Insert transformed text over a given primary axis.
Rationale: By combining this subsystem with the scanning from phase 1 and axis identification in phase 3 we have the complete system to meet the phase 3 use case.
Fit Criterion: This will be verified by having text sensibly displayed over the relevant axis of interest.

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 (hobbyist) 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

When it comes to scanning resistors, labelling them with their respective values is just the tip of the iceberg. The team would like to be able to scan a resistor of any size and colour against any surface, or scan by holding the resistor up to the camera. If possible, the application would also allow for a requested ohmage to be inputted and the combination of resistors needed would be highlighted.

5.2 Off-the-Shelf Solutions

There are many available applications to scan resistors and read resistor colour codes. A notable example is an application available on the Google Play store by the name of "Resistor Code Calculator" that allows a user to simply tap-to-highlight the coloured bands of the resistor in order and the app automatically calculates the resistance. The app doesn't solve the problem of colour blind

users, but it solves the issue of reading a resistor in a much simpler way than our team's implementation. Similarly, many apps exist on app store that either allow the user to create a virtual circuit, or contain a large about of data about resistors that our application does not provide.

5.3 New Problems

There are currently none.

5.4 Tasks

See the Gantt chart below for an outline of tasks to be completed.
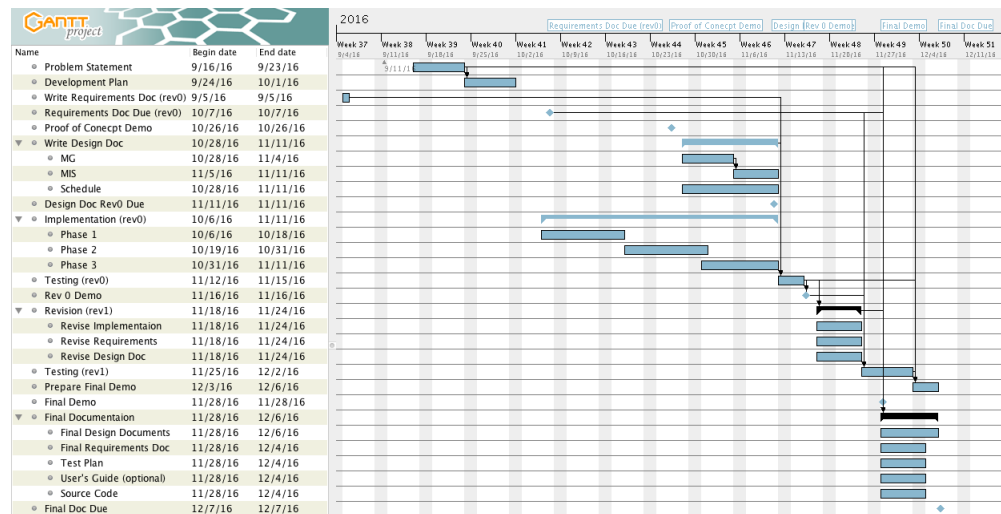
5.5 Migration to the New Product

The product is designed in such a way that the core functionality of processing resistor data and outputting the value of the given resistor will never change. Instead, new features will be implemented in order of their priority, determined in the requirements section.

5.6 Risks

There are technical risks associated with building on the existing resistor product by including features such as the ability to scan multiple resistors at once, or find the requested resistor from an area containing multiple resistors. These risks are mitigated by having a phase-based development plan in which we start with the necessary base of the application and add new features that build on each other in each new phase.

5.7   Costs

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

5.8   User Documentation and Training

No training is required to use this application. A user guide will be available in the form of a popup menu when the app is first launched.

5.9   Waiting Room

See Phase 2 & 3 Functional Requirements.

5.10   Ideas for Solutions

Using OpenCV, colours can be "binned" a small number of colours based on a range of RGB values.
ie: All colours with low R, low G and medium to high B can be reduced to simply one shade of blue to make identification easier.