Software Requirements Specification

CircuitSnap

Mobile Application

Version: 1.5

March 18th, 2016

Project Manager and Assistant Project Manager
Kyle Baxter
Beverly Abadines

Software Engineers
Luis Pow Sang
Henry Ramirez
Ismael Sanchez

UI/UX Designer
Hyundeuk (Richard) Cho

Table of Contents

Introduction **3**
1.1 Purpose 4
1.2 Scope of the Project 4-5
1.3 Definitions, Acronyms, and Abbreviations 5-7
1.4 References 7
1.5 Overview 7
Overall Description **7**
2.1 Product Perspectives 7
2.1.1 System Interfaces 7
2.1.2 User Interface 7
2.1.2.1 App Startup 7
2.1.2.2 Node and Component Input 7-8
2.1.2.3 Prepare Assembly 8
2.1.3 Software Interfaces 8
2.1.3.1 OpenCV Manager (Version 2.4.9+) 8
2.1.3.2 Google Play 8
2.1.4 Communication Interfaces 8
2.1.5 Memory 8-9
2.1.6 Operation 9
2.1.7 Site Adaptation Requirement 9
2.2 Use Case 9
2.3 User Characteristics 9
2.4 Constraints 9
2.4.1 Operating Systems 9-10
2.4.2 Connections 10
2.4.3 Component Data Input 10
2.4.4 Component Variety 10
Specific Requirements **11**
3.1 External Interface Requirements 11
3.1.1 User Interface 11
3.1.1.1 Loading Screen 11
3.1.1.2 Main Menu Display 11-12
3.1.1.3 Data Input 12
3.1.1.4 Assembly 12
3.1.2 Hardware Interface 12
3.2 Functional Requirements **12**
3.2.1 Navigation to Main Menu 11
3.2.2 Node Placement and Characterization 12
3.2.3 Value Initialization 13

3.3 Performance Requirements 13
3.4 Design Constraints 13
3.5 Software System Requirements 13
3.5.1 Portability.....13
3.5.2 Reliability.....13
3.5.3 Maintainability.....13
3.6 Document Approval……..……..…….....13

1. Introduction

1.1 Purpose

The purpose of this software requirements system is to provide a detailed description for the continuous development of the mobile application, CircuitSnap. It contains both an outline of the software to be developed as well as detailed and specific requirements. This document's expected audience is Dr. Timothy Usher, our client, and Physics Professor at California State University San Bernardino, who promoted the idea and design, as well as Dr. Concepcion for the purpose of review and approval.

1.2 Scope of the Project

The software to be designed will be a mobile application to be used by the students and faculty of CSUSB, as well as hobbyists or those involved with or interested in circuits. Its primary purpose is to assist and promote self-directed learning and a self-paced environment for students.

For this iteration, CircuitSnap will utilize Android Studio to create an ELVIS board simulator to serve as both a virtual practice board and logic error detector. Once the user inputs the node placements and specifies the components, the application will engender a pictorial circuit diagram along with notifications of circuitry misuse and possible corrections.

Upon the creation of the **first prototype**, the application will display a full ELVIS board and navigate to zoomed quadrants. The simplicity of this development is due to the last minute change in direction. As for the **second prototype**, based off the user inputs of both node placements and component specification, which are note: limited to wires, capacitors, LED's and resistors. CircuitSnap will display the logo and navigate to either the user guide or the ELVIS board. The displayed board will serve merely as a reference for input. From there, the user will input the endpoints for the bidirectional components listed above, as well the name of the component and prepare to order the items based off the endpoints. As suggested by the client, the application will only focus on the a fourth of the board, since basic circuits do not expand largely with limited components. The circuit to be assembled should be restricted to complete series circuits.

When this project reaches maturity, before being considered for publishing, it should be capable of detecting and translating more complex circuits from pictures on Android devices and have an 80% recognition accuracy. It will not be limited to the assortment listed above. Further development, beyond that, is expected but can be implemented in updates to CircuitSnap, such as the inclusion of software capable of reading resistor resistance. In the last iteration of this project, there will be two uses of the program. One serving as an ELVIS simulator which is the goal for this project, and the other as an instantaneous picture to diagram translator, which is the goal of other iterations. Although, work on latter is not applied for this quarter's project and goals, it is delineated throughout the software requirements specifications for future endeavors.

To continue on the latter, to go from picture to diagram the application will utilize object detection functionality from the OpenCV library, which serves as the main component for

observing images and extracting information, such as positions of the parts in order to calculate connections between them. Once the desired wirings are identified, it engenders a pictorial circuit diagram, much like the simulator.

1.3 Definitions, Acronyms, and Abbreviations

Table 1 – Definitions

Term	Definition
OpenCV	OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source BSD licensed library that includes several hundreds of computer vision algorithms.
Haar Like-Features	Wavelets used for object recognition and digital image features
Adaboost	The additional component of Haar which determines based on light and dark values of the picture.
Android Studio	Official integrated development environment (IDE) for developing for the Android platform, which is an open-source operating system for mobile devices
SDK	A software development kit (SDK or "devkit") is typically a set of software development tools that allows for the creation of applications for a certain software
Java	A high-level, object oriented computer programming language used especially to create interactive applications running over the Internet.
Resistor	A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element.
Capacitor	A capacitor (originally known as a condenser) is a passive two-terminal electrical component used to store energy electrostatically in an electric field.
DIODE	In electronics, a diode is a two-terminal electronic component with asymmetric conductance, it has low (ideally zero) resistance to current flow in one direction, and high (ideally infinite) resistance in the other.

LED	A light-emitting diode (LED) is a semiconductor light source.
Potentiometer	Device for measuring electromotive force or potential difference by comparison with a known voltage, adjustable in use.
Terminal Strip	An insulated bar that contains a set of screws to which wires are attached.
DIP Support	Gap reserved for Dual Inline Packages on breadboards
Power Rail	Parallel groupings on the board typically used for ground and voltage connections
Node	Holes for component input. -NumNode for main grid ranges from 1-63 -LetNode for main grid ranges from A-J (limited to leftmost side of board) -NumNode for I/O ranges from 1-4 -Placement for I/O ranges from 1-54
DC Power	I/O 53 = GND I/O 54 = + 5V
Power Supply	Section of the ELVIS specifically for variable and direct current and ground
Main Grid	The large center portion of the board
Quadrants	The main grid broken up into fourths
NI ELVIS	National Instruments' Educational Laboratory Virtual Instrumentation Suite designed for interactive learning as a modular platform

1.4 References

IEEE std 830- Software Requirements Specification

CircuitSnap Mobile Application SRS (1.2)

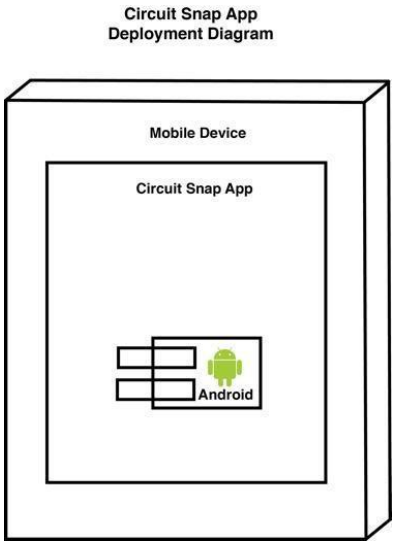
1.5 Overview

For the rest of the document, it is separated into two sections. Section 2 is the functionality and interface of the application. Functionality includes the process of each step of the application as well as the meaning of why it is this way. Section 3 consists of the user interface which is the physical look of the application that the user will be seeing when using this application. The

physical look, directed by our user interface team, will have the beginning splash screen, the main menu, options screen, and the final diagram screen.

2. Overall Description

2.1 Product Perspectives



2.1.1 System Interfaces

Android will be used as the operating environment the user will be interacting with. This environment will include a feature where pictures are able to be converted into diagrams where users can implement their designs.

2.1.2 User Interface

2.1.2.1 App Startup

The logo is first shown and then users will have the ability navigate to the board or user guide.

2.1.2.2 Node and Component Input

Users must type the endpoints (First and Second), and Input Component.

First and Second Input must follow the typed sequence:

-a letter between “A-J” (main grid) or the phrase, “I/O53” or “I/O54” (I/O)

- a colon,

-followed by a number from “1-54” for I/O selections or “1-63” for main grid selections

Input Component is limited to strictly being:

-“resistor,” “cap,” “led,” or “wire”

2.1.2.3 Prepare Assembly

The components are ordered due to their endpoint placements.

2.1.3 Software Interfaces

Note this section is for future use:

2.1.3.1 OpenCV Manager (Version 2.4.9+)

OpenCV manager provides the libraries necessary to use the OpenCV functions in this application by:

• Implementing a process that begins when a picture is taken. Next the OpenCV program starts and begins to compress the image into a vector file.

• The vector file is then cascaded through Haar training which compares the positive images (the ones that the user needs) to the negative images (any random images).

• With Haar Training, the positive and negative images are broken down into pixels and then compared to a rectangle that is of white contrast and a rectangle that is of black contrast. When compared, the sum pixels are subtracted by the white rectangle and then subtracted by the black rectangle. Once that's finished, it continues to Adaboost.

With Adaboost. the images are then confined to be specific to the image itself. When it comes to the picture of a circuit board, the darker features are the holes within the board versus the slight crevices between each grouping on holes. The holes would have more a darker value than the crevices, so it would distinguish between the two. Adaboost is within the cascading rather than being a tool by itself. From then on, the detection could continue..

• Through the high amounts of cascading, it begins to learn that the positive images of the proper circuit components are correct and anything else will be excluded.

2.1.3.2 Google Play

This app will utilize Google Play to download to a device, and to receive updates.

2.1.4 Communication Interfaces

3G or better connection, or Wi-Fi, is required for download, and to receive updates, of this application. Later iterations of this project may permit this application to save circuit diagrams to a server.

2.1.5 Memory

- The application is expected to not exceed 80 MB of RAM.
- Not including saved documents and pictures, this application is expected to not exceed 15 MB.
- The minimal platform supported by this application is to be determined

2.1.6 Operation

This app is expected to be operational at any time, with no need for downtime. As updates are applied and submitted for download, it is the user’s responsibility to update their application, if they wish to do so.

2.1.7 Site Adaptation Requirement

Not applicable at this time.

2.2 Use Case

The following is a use case diagram to describe the functions of the app:

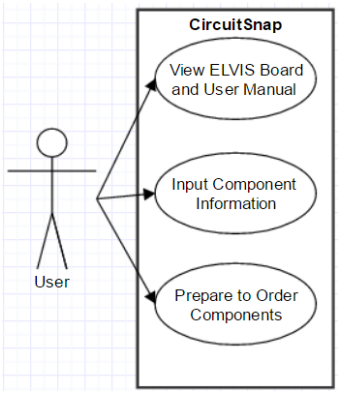
1. When the user opens CircuitSnap the application will run in vertical mode and then in landscape mode.

2. A splash page with the CircuitSnap logo is displayed, and after a couple seconds the screen will be in the main menu.

3. It can then navigation to the ELVIS board or the user manual.

4. The user must input the endpoints after selecting the component with the inputs consisting of a letter, colon, and number resembling the structure “A:7”.

5. Once the user is completed with inputting values, the application will prepare to order the components based on input connections.



2.3 User Characteristics

Typical users of this program are expected to be students or instructors of computer engineering or physics courses who study with circuit design on a breadboard or ELVIS board.

2.4 Constraints

2.4.1 Operating Systems

Use of this application will require using a mobile device able to support Android operating system. Specifications on the model and version are yet to be determined at this time. It is worth noting that as a native-built application, iOS is not supported at this time.

2.4.2 Connections

Note this section is for future use:

This application will require the use of the user's own storage of the images. The images are sent to a CircuitSnap folder that the user can access anytime. The folder will consist of an .xml file that contains many images that will help in the object detection process unless deleted by the user.

2.4.3 Component Data Input

• The user indicates occupied nodes as pairs, since the components applied are all double-ended and must be applied for closed series circuits.

• Only acceptable values for Input First Position and Input Second Position are a letter followed by a colon and then a number. Refer to Node definition to find the range of these values.

• Once the pair of nodes are placed, the user specifies if the component is a wire, resistor, capacitor, LED and must be inputted as “wire,” “resistor,” “cap,” or “led” as the Input Component.

It is capable of recognizing the errors:

-no inputs, going beyond the Node ranges, not inputting a single I/O[53] (GND) and I/O[54](PWR), or repeating an inputted node

Otherwise this will fail if the implications are not followed.

For further iterations:

• If the component is a:

- Resistor, then resistance [ohms] is to be specified
- Capacitor, then capacitance [F] is to be specified as well as polarity
- Wire, then color [black, red, blue, green, yellow] is specified
- LED, then color [black, red, blue, green, yellow] and polarity is specified

2.4.4 Component Variety

Although there is a large amount of circuit components that are going to be implemented, this application will concentrate on the wirings between them and the meaning of their connections. Usable components may be resistors, capacitors, potentiometers, and LEDs,

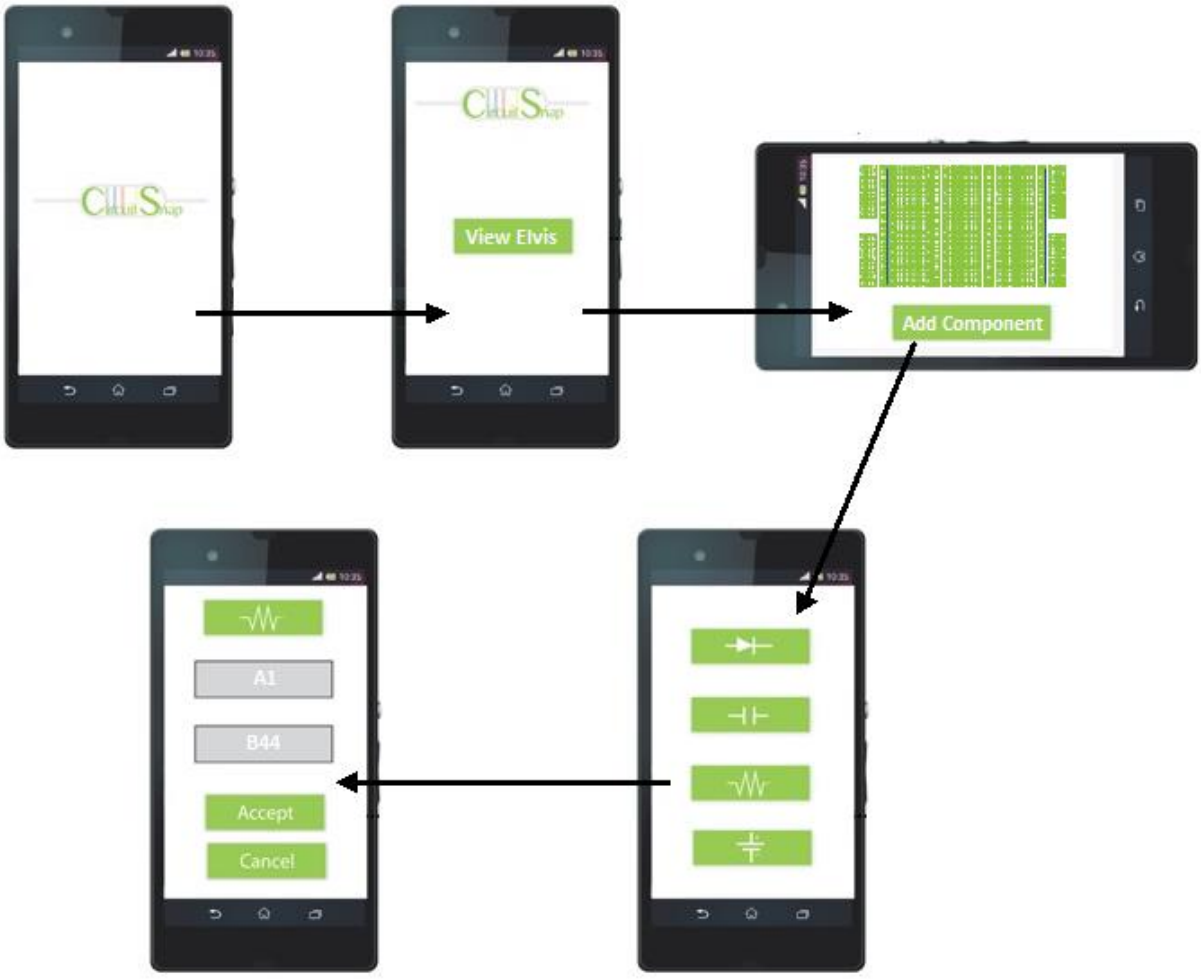
3. Specific Requirements

3.1 External Interface Requirements

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interface



3.1.1.1 Loading Screen

A splash screen will be displayed on the loading screen with the application logo and name “CircuitSnap”.

3.1.1.2 Main Menu Display

On the menu screen, the user will be prompted to press the button “Go to Board” in order to enter the application ELVIS board display or to “User Guide,” which indicates the application implications

3.1.1.3 Data Input

The users are prompted to list out the components and their endpoints. The first two inputs should follow the structure listed on section 3.4.2.

For ex) J:34 is a valid input. The third input is to reiterate the component type.

3.1.1.4 Assembly

Prepares to order the components for a closed, simple series circuit.

3.1.2 Hardware Interface

An Android 2.1+ compatible device will be required to run the application.

3.2 Functional Requirements

The following functions are to be included in the application:

3.2.1 Navigation to Main Menu

• Display logo and a button is going to be available to go to ELVIS board and its ` quadrants

3.2.2 Node Placement and Component Characterization

• User may specify if it was a wire, resistor, LED, or capacitor after which, the user inputs the endpoints which should consist of:

-a letter between “A-J” (main grid) or the phrase, “I/O53” or “I/O54” (I/O)

- a colon,

-followed by a number from “1-54” for I/O selections or “1-63” for main grid selections

Two consecutive selections indicate a double-ended component.

3.2.3 Value initialization

• Refer to section 2.4.3 pg 9

3.3 Performance Requirements

CiruitSnap is designed to run on most android devices. As for its photography feature, it allows a picture is taken to be used via the camera with an 80% accuracy rate. Note that the time to analyze the circuit is dependent on the complexity and length of the circuit in question and for such cases, no response time can be guaranteed.

3.4 Design Constraints

Due to the native programming, the application is limited to the Android operating system. Graphics will need to be able to scale to the size of the device's screen.

3.5 Software System Requirements

3.5.1 Portability

At this time, this application is Android specific, and cannot be implemented on devices with other operating systems.

3.5.2 Reliability

Analysis of much more complicated circuits with abundant wirings, copious components, or ambiguous values, is currently not guaranteed. As development is continued, the application will be more robust in its interpretation of circuits.

Parallel or open circuits are not processed, as well as inputs deviating from the Node Placement and Component Characterization mentioned in 3.2.2. This offers conceptual aid more than calculative measurements.

3.5.3 Maintainability

Maintenance of this program will be required for continued addition of object recognition with camera manipulation.

3.6 Document Approval

This document must be approved by:

• Dr. Tim Usher

Client, CSUSB Physics Professor