

CircuitSolver Requirements

Version 1.1

Product Vision Statement

Text description

CircuitSolver is a mobile app that allows one to analyze the voltage across, current through and resistance of components in their hand-drawn circuit. It aims to eliminate the inconvenience (and time) of using in-app circuit builders and printable components kits when trying to analyze circuits.

Problem Statement

The problem of	Analyzing a circuit schematic drawn on paper
affects	Students, electricians and engineers
the impact of which is	Loss of time manually solving for values
a successful solution would be	A mobile application which can interpret hand-drawn circuits and solve for missing circuit values automatically

Product Position Statement

For	Engineers and engineering students
Who	Prototype or solve circuits on paper
Our System	Is all software
That	Interprets photos of circuit schematics and solves them
Unlike	Other products which require manual work rebuilding circuit in in-app circuit editor OR Other products which require drawing directly on touch-screen device, which is difficult on small touch screens OR Other heavyweight products that have lots of complex features not necessary for solving circuits
Our Product	Is simple, lightweight, and automatically understands hand

User Demographics

CircuitSolver is a mobile app that targets engineers and students, the end users.

End Users: End users are the users that will be interacting with the app the most. It is expected that the users have an understanding of mobile phones. Also, that they are able to interact with their phone and take pictures. The users will be ones who are doing calculations with AC and DC circuits in either work or school. These students and engineers are looking for a quick and simple way to solve their circuits, allowing them more time to design instead of calculate.

Feature List

The app will allow the user to solve any hand-drawn circuit. To get to this objective one will have the following features :

- Scan his hand drawn circuit
- Solve the circuit and know about the voltage and the current at any point of the circuit
- Have a graphical plot of these two values
- Modify any circuit made in the past to reevaluate it

Constraints

Being a mobile app, the group will have to deal with resource constraints because of the huge variety of Android phones, ranging from powerful to barely useable. As well, using OpenCV brings up possible constraints with processing power as well as the ability to detect shapes and lines in the image.

Scopes and Limitations

The app includes image recognition and circuit analysis for drawn circuits on a two-dimensional surface. It does not support recognition of actual circuit components or any circuits drawn at an obscure angle

The camera can focus on a target using the app and a camera light can be turned on if the phone has one. However the application does not support any further enhanced camera features. For these the user must exit the app and take the picture separately and then load the result

The application includes template matching recognition algorithms for, and only for, the following circuit components: Battery, resistor, light, wire. No other circuit component will be recognized unless there is time to implement further template matching

The application will not include advanced file management for stored circuits. Users will be able to search for a circuit by a name they have given it. No folder creation or addition of tags

or any other File system features will be included other than search and basic navigation. The user will have to exit the app and use a file system app for more advanced features

Our app aims for simplicity and will refrain from frivolous aesthetic features (such as animations when solving circuit). Only potential difference, resistance, current, and direction of current will be displayed for each component the user taps.

Assumptions and Dependencies

- We assume that OpenCV and/or Haar's algorithm will provide us with sufficient template matching allowing for adequate circuit component identification performance
- We assume that genymobile will be sufficient for testing different Android platforms

Use Cases

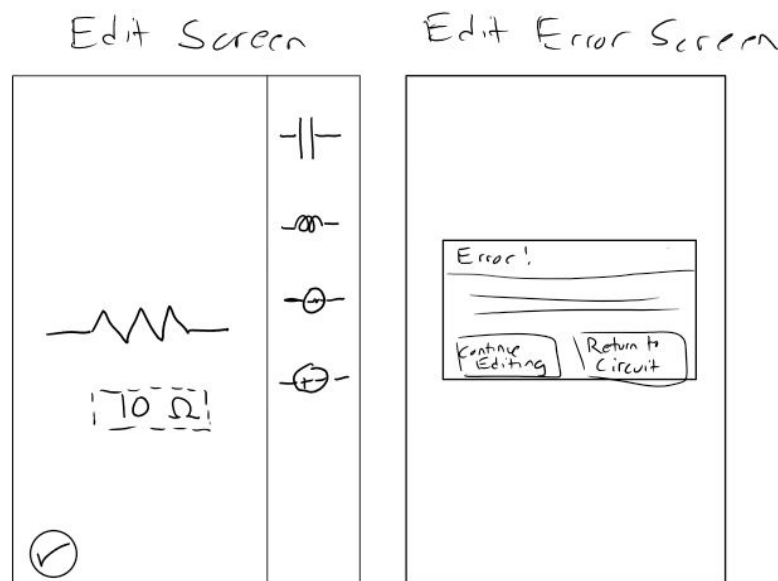
CircuitSolver Use Case Document

USE CASE 1: Edit a previous saved circuit in order to change some components or change numerical values.

Version	Description	Changed By	Date
1.0	First draft	Jennifer Lam	09/26/16

1. **Brief Description:** User has a saved circuit saved on his device that he wants to edit and re-analyze.
2. **Preconditions:** User has a View Circuit page open on CircuitSolver
3. **Business Trigger:** User previously used the app to analyze a circuit, but now wants to re-analyze the same circuit with different value(s) or/and different component(s). He goes to the home screen and taps "Edit".
4. **Basic Flow:**
 5. User chooses the previous circuit he/she wants to edit.
 6. User taps "Edit"
 7. Application displays Edit Circuit screen
 8. User taps on component he/she wants to edit
 9. Application displays enlarged component and it's corresponding value, as well as a sidebar displaying other component options
 10. User taps on one of the component options
 11. Application replaces the previously displayed component with the selected component. The displayed value changes to default value.
 12. User taps on displayed value.

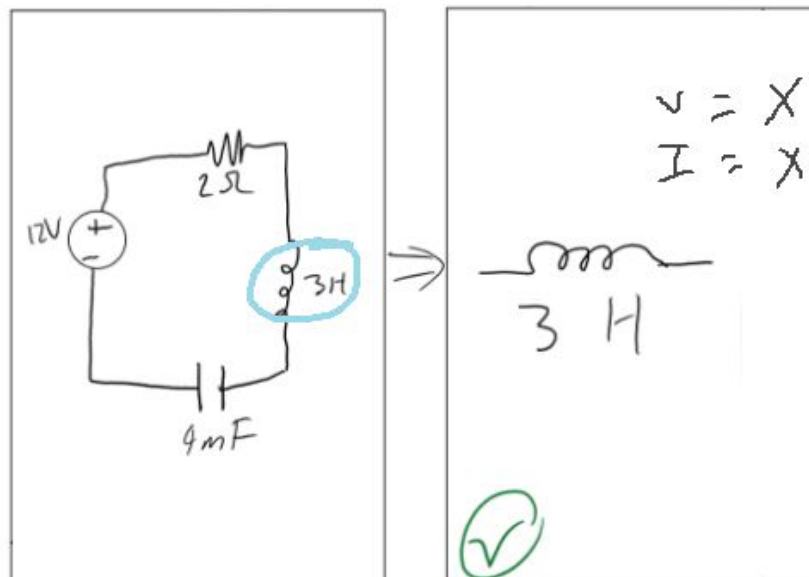
13. Application component's displayed value becomes a textbox for user to enter new value
14. User enters new value in textbox
15. Application displays updated value
16. User taps "Done"
17. Application displays updated circuit (same as View Circuit screen)
18. **Post Condition:** After the basic flow is complete, the circuit will be fully analyzed and the user will be able to view all the values of electrical components by tapping on them
19. **Alternate Flow: Only change numerical value**
20. When in Step 9, skip to Step 12 and continue Basic Flow
21. **Alternate Flow: Fixing unsolvable circuit**
22. When in step 16, after pressing "Done" and the circuit has become unsolvable, the user will be shown an error saying that the recent change made caused the circuit to be unsolvable.
23. The user is prompted to either revert value back to the last valid value or change the value of the component.
24. If circuit is still unsolvable, steps 21 & 22 will be repeated.
25. **Post Conditions:** If circuit stays unsolvable, application displays analyzed circuit (i.e. it reverts back to the same as View Circuit Screen)
26. **Prototype Screen:**



CircuitSolver Use Case Document

USE CASE 2: View component values of a previous analyzed circuit

1. **Brief Description:** After the application solves the circuit, the user checks the values of various components
2. **Preconditions:** User has a View Circuit page open on CircuitSolver
3. **Business Trigger:** User previously used the app to analyze circuit, but and wants to view the analyzed circuit again.
4. **Basic Flow:**
 5. The user chooses the previous circuit he/she wants to view
 6. Application displays the Circuit in full-screen mode
 7. The user taps on a circuit element (resistor, battery, light bulb or wire segment)
 8. Application enlarges circuit element
 9. Application displays values corresponding to circuit element (resistance, calculated voltage, calculated current)
 10. User can repeat steps 7-9
11. **Post Condition:** The resistance, voltage and current through the circuit components the user selected is displayed as they are selected
Ohm's law is held
12. **Prototype Screen:**

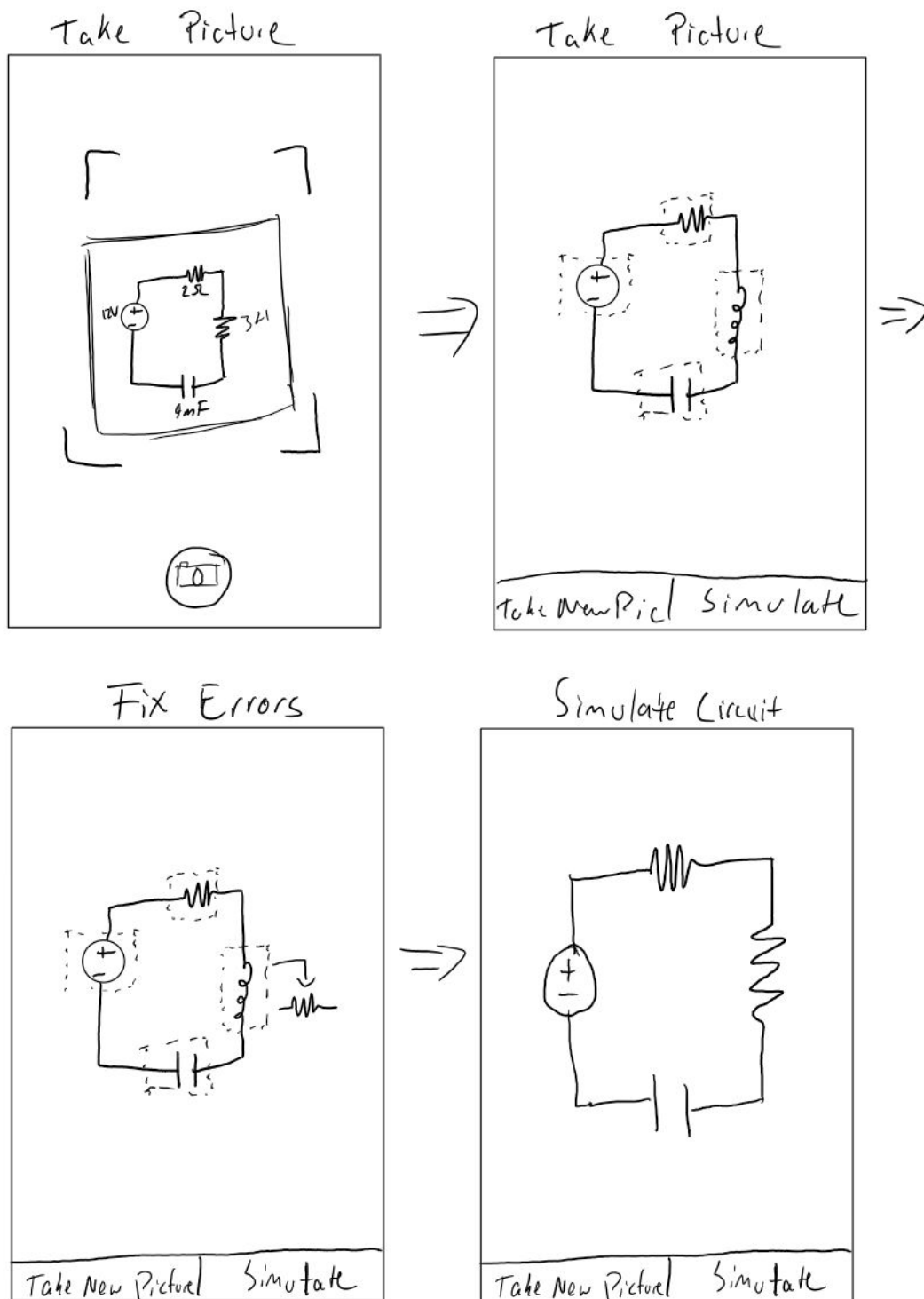


CircuitSolver Use Case Document

USE CASE 3: Solve a circuit from an uploaded or taken circuit picture

-
1. **Brief Description:** (Before a circuit can be saved, it must be solved first)
 2. **Preconditions:** User has a home page open on CircuitSolver
 3. **Business Trigger:** The user has an image of a circuit or wants to take a picture of a drawn circuit to analyze the voltage difference and current of different components in the circuit.
 4. **Basic Flow:**
 5. User selects the "take picture" button and selects take picture.
 6. Application opens a camera screen
 7. User takes the picture
 8. Application receives the image and produces a virtual circuit. For the virtual circuit each circuit component (other than the wires) is a tappable by the user which displays that component's values (as in Use Case 1 and 2).
 9. Application asks user if circuit components have been correctly identified
 10. The user taps "yes"
 11. The application saves the virtual circuit
 12. **Post Condition:** The virtual circuit is saved and it is known that the user is satisfied with the components the application identified
 14. **Alternate Flow: Edit component(s)**
 15. When in line 9, the user taps "no"
 13. The application displays the Edit Circuit screen
 14. The user taps on component he/she wants to edit
 15. The application displays enlarged component and it's corresponding value, as well as a sidebar displaying other component options
 16. User taps on one of the component options
 17. Application replaces the previously displayed component with the selected component.
 18. Steps 13-17 may be repeated
 19. The user presses the "Save" button
 20. Go to basic flow line 11
 21. **Post Conditions:** Same as basic flow
 16. **Alternate Flow: Upload a picture**
 17. When in line 5 the user selects the upload/take picture button and selects load picture
 18. Application opens local user gallery
 19. User selects the desired circuit image saved on their device
 20. Go to basic flow line 8
 21. **Post Conditions:** Same as basic flow
 22. **Business Rules:**

23. Prototype Screen:



Non-functional Requirements

Speed

The scan of the picture and recognition of the circuit has to be done in a decent amount of time (max 10 sec).

Image detection

The detection of the image and its transformation to a digital version shouldn't contain more than (30%) of errors.

Adaptability

The application has to work for any hand drawing made by different people, given the general shape of a circuit component.