

CircuitSolver Requirements

Version 1.2

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
- Modify circuit parts in case picture is incorrectly identified (e.g. add wire manually if it is missing between components)
- Canvas for the user to draw their own circuit and add their own components without taking a picture (in case the user doesn't want to draw it on paper)

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 algorithm constraints, limiting our ability to detect corners and components 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, for the first version of the app, we can only recognize almost straight lines, either horizontal or vertical.

The camera can focus on a target using the built in camera, as well, you can use the flash if necessary. 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 edited image into the app.

The application includes a combination of OpenCV algorithms to detect lines, both horizontal and vertical, as well as components. The app currently supports using resistors, inductors, capacitors, and DC voltage sources. The app can be enhanced to include support for AC circuits, as well as more complex components.

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 provides us with sufficient functions to be able to pull out the exact positions of the corners and the components.
- We assume that the parameters used for clustering and post-processing can be deduced with high precision from the camera resolution.
- We assume that a Tensorflow neural network can be generated with sample images, and then correctly identify components.
- We assume that unit tests and successive runs of the program on android devices and/or emulators will be enough to debug our program correctly.
- We assume that ngspice will give the correct output if we satisfy NGSpice's preconditions.

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
1.2	Final submission, all Use Cases updated	Dirk Haupt	11/19/16

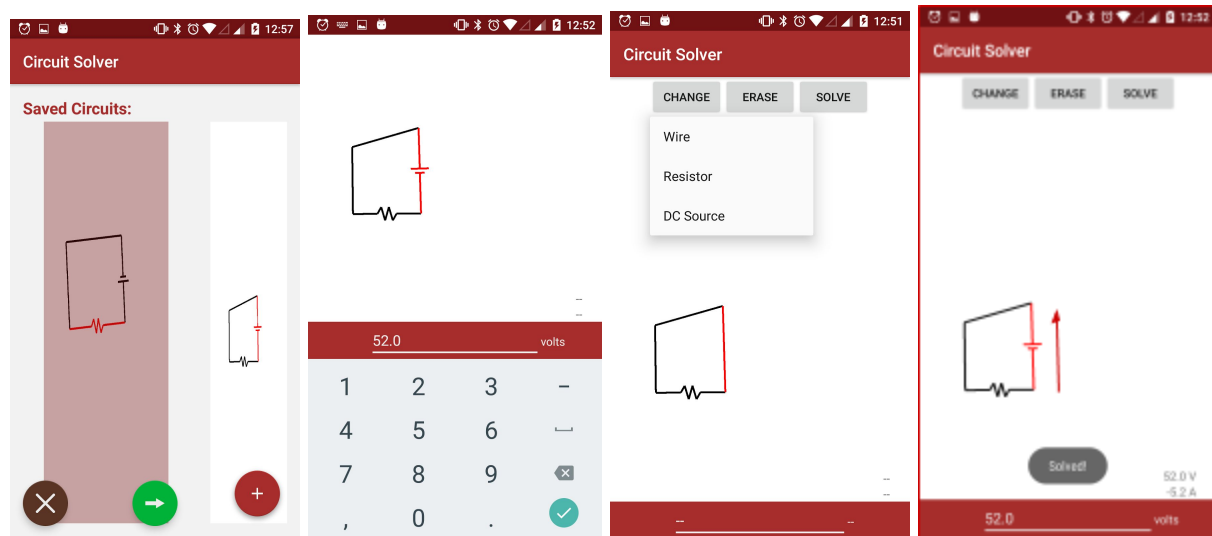
1.	Brief Description:	User has a saved circuit saved on his device that he/she wants to edit and re-solve.
2.	Preconditions:	User has a saved circuit saved on his device and the Home Page open on CircuitSolver
3.	Business Trigger:	User previously used the app to solve a circuit, but now wants to re-solve the same circuit with different value(s) or/and different

component(s). He goes to the home screen and taps his circuit and presses the green arrow button

4. **Basic Flow:**

5. User chooses the previous circuit he/she wants to edit.
6. User taps the green arrow button
7. Application displays the Circuit screen
8. User taps on component he/she wants to edit
9. The component turns red, the "Add" button changes into an "Change" button, the displayed value in the textbox at the bottom of the screen switches to the value saved for the selected component and the displayed value's units changes to match the component (ohms for a resistor, volts for a voltage source etc.)
10. User taps on the "Change" button
11. The application displays a dropdown list of circuit component options
12. The user taps on the desired component
13. Application replaces the red selected component with the component from the list and the list disappears from view. The displayed value of the component changes to its default value and changes its units to match the new component
14. User taps on the displayed value textbox.
15. Application displays the numpad for user to enter new value
16. User enters new value in textbox and presses the tickmark
17. Application displays updated value
18. User taps on the component they want to see updated values for as a result of the change they just made
19. The previously selected component turns black again and the tapped component turns red, the displayed value in the textbox at the bottom of the screen switches to the value saved for the selected component and the displayed value's units changes to match the component
20. User taps "Solve"
21. Application displays solved values for the selected circuit component in the bottom right and a "solved!" notification as well as the current through the selected component via an arrow as well as a small arrow on/nearby the component displaying current direction
22. **Post Condition:** After the basic flow is complete, the circuit will be fully solved and the user will be able to view all the values and current direction through each circuit component by tapping on them
23. **Alternate Flow: Only change numerical value**
24. When in Step 10, skip to Step 14 and continue Basic Flow
25. **Alternate Flow: Only change component**
26. When in Step 14, skip to Step 18 and continue Basic Flow
27. **Alternate Flow: Fixing unsolvable circuit**
28. When in step 20, after pressing "Solve" 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.
29. The user is prompted to either revert value back to the last working circuit configuration or to change individual component and component values manually without solving

30. If circuit is still unsolvable after the user makes manual changes and taps “Solve”, steps 28 & 29 will be repeated.
31. **Post Conditions:** The application displays the values and current direction of a solved circuit for each tapped circuit component
32. **Prototype Screen:**

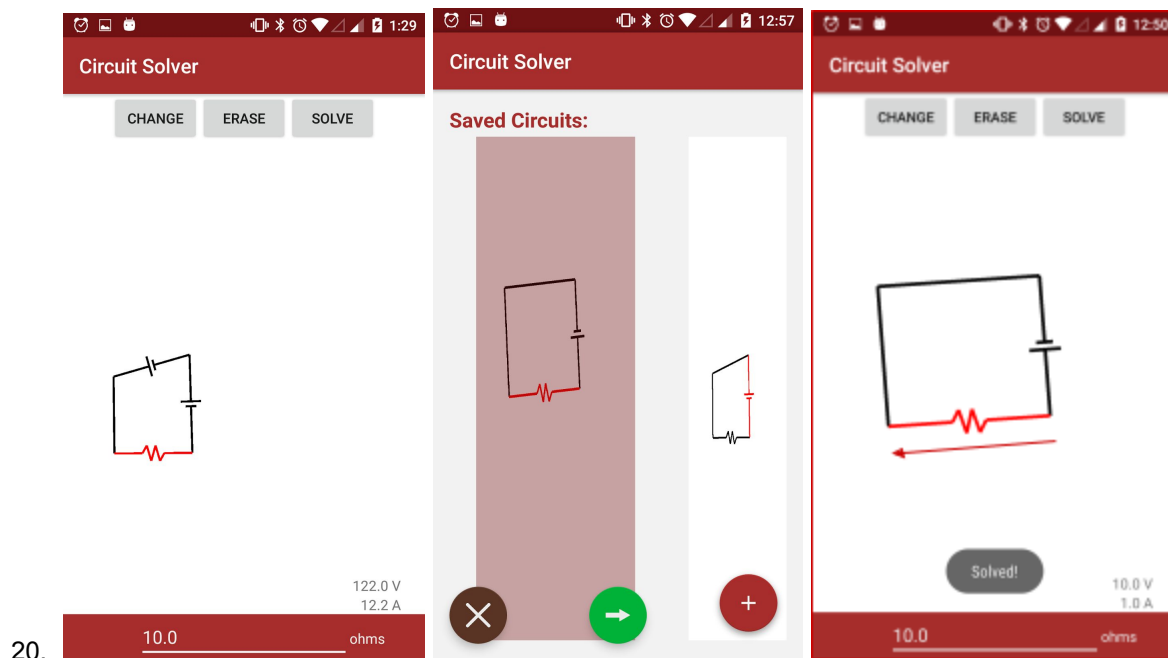


CircuitSolver Use Case Document

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

1. **Brief Description:** After the application solves the circuit, the user uses another app and returns to check the values of various components
2. **Preconditions:** User has just returned to the Circuit page from another app and the Circuit Solver application was not closed (was placed on the list of running background apps)
3. **Business Trigger:** User previously used the app to solve a circuit, has since started using another application, but now wants to view the solved circuit again.
4. **Basic Flow:**
 5. The user returns to Circuit Solver from another application
 6. The circuit page with the circuit the user previously solved is brought back into view
 7. User taps on component he/she wants to view values for
 8. The component turns red, the “Add” button changes into a “Change” button, the displayed value in the textbox at the bottom of the screen switches to the value saved for the selected component, the displayed value’s units changes to match the component (ohms for a resistor, volts for a voltage source etc.) and the application displays values corresponding to circuit component in the bottom right that was previously computed
 9. User can repeat steps 7-9
10. **Alternate Flow: View values for a different circuit**
11. When in Step 7, the user presses the back button

12. The application brings the Home screen into view
13. The user chooses a previously solved circuit he/she wants to view.
14. The circuit page with the circuit the user previously solved is brought back into view
15. The user taps "Solve"
16. User taps on component he/she wants to view values for
17. skip to Step 8 and continue Basic Flow
18. **Post Conditions:** The resistance, voltage and/or current through the circuit components the user selected is displayed as they are selected
Circuit logic (Ohm's law etc) is maintained
19. **Prototype Screen:**



CircuitSolver Use Case Document

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

1. **Brief Description:** Solve a circuit from an uploaded or taken circuit picture
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 on the circuit page. 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. User pinches the screen to zoom in and out from the circuit to see whether he/she is satisfied with the virtual representation of his/her circuit
10. User taps on a circuit component
11. The component turns red, the "Add" button changes into an "Change" button, the displayed value in the textbox at the bottom of the screen switches to the value saved for the selected component and the displayed value's units changes to match the component (ohms for a resistor, volts for a voltage source etc.)
12. User taps on the displayed value textbox.
13. Application displays the numpad for user to enter new value
14. User enters new desired value for the component in textbox and presses the tickmark
15. Application displays updated value
16. Steps 10-15 are repeated for components until the user is satisfied with the value assigned to each circuit component
17. User taps "Solve"
18. Application displays solved values for the selected circuit component in the bottom right and a "solved!" notification as well as the current through the selected component via an arrow as well as a small arrow on/nearby the component displaying current direction
19. **Post Condition:**
 - The application displays the values and current direction of a solved circuit for each tapped circuit component
 - Circuit logic (Ohm's law etc) is maintained
20. **Alternate Flow: Fix component placement**
21. When in Step 12, the user instead taps the "Erase" button
22. The application deletes this component
23. Steps 21-22 may be repeated
24. The user taps "Add"
25. The application displays a dropdown list of circuit component options
26. The user taps on the desired component
27. A message appears informing the user of their selected component
28. The user draws the circuit component on screen by tapping to anchor the one end and then dragging the other end of the component to its desired location. This determines the direction of components for which it matters (e.g. voltage sources)
29. Steps 24-28 may be repeated
30. Return to Step 10 and continue Basic Flow
31. **Post Conditions:** Same as basic flow
32. **Alternate Flow: Edit component(s)**
33. When in Step 12, the user taps the "Change" button instead
34. The application displays a dropdown list of circuit component options
35. The user taps on the desired component
35. Application replaces the previously displayed component with the selected component.
36. Steps 33-35 may be repeated
37. Return to Step 10 and continue Basic Flow

38. **Post Conditions:** Same as basic flow

40. **Alternate Flow: Upload a picture**

41. When in line 5 the user selects the upload/take picture button and selects load picture

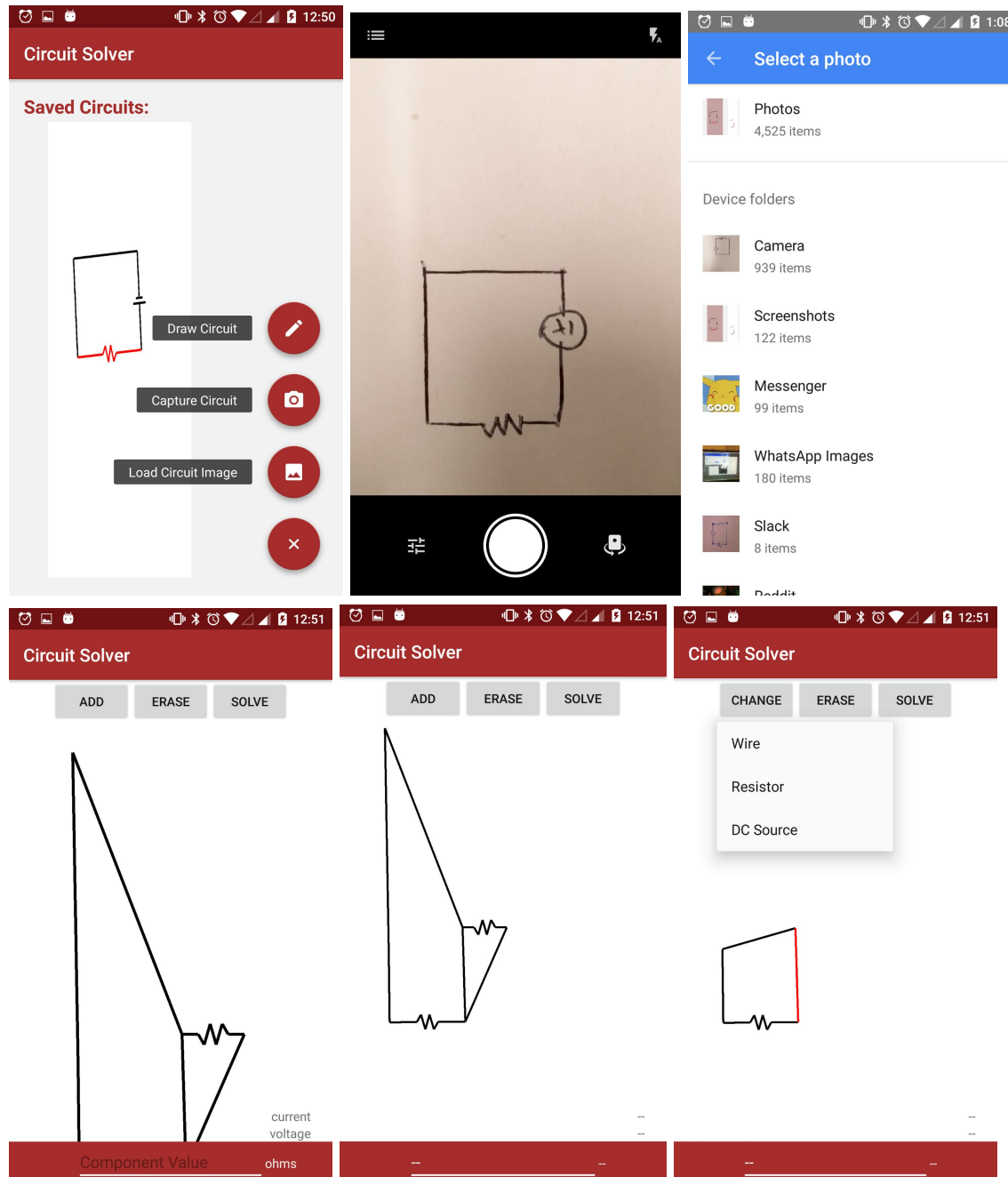
42. Application opens local user gallery

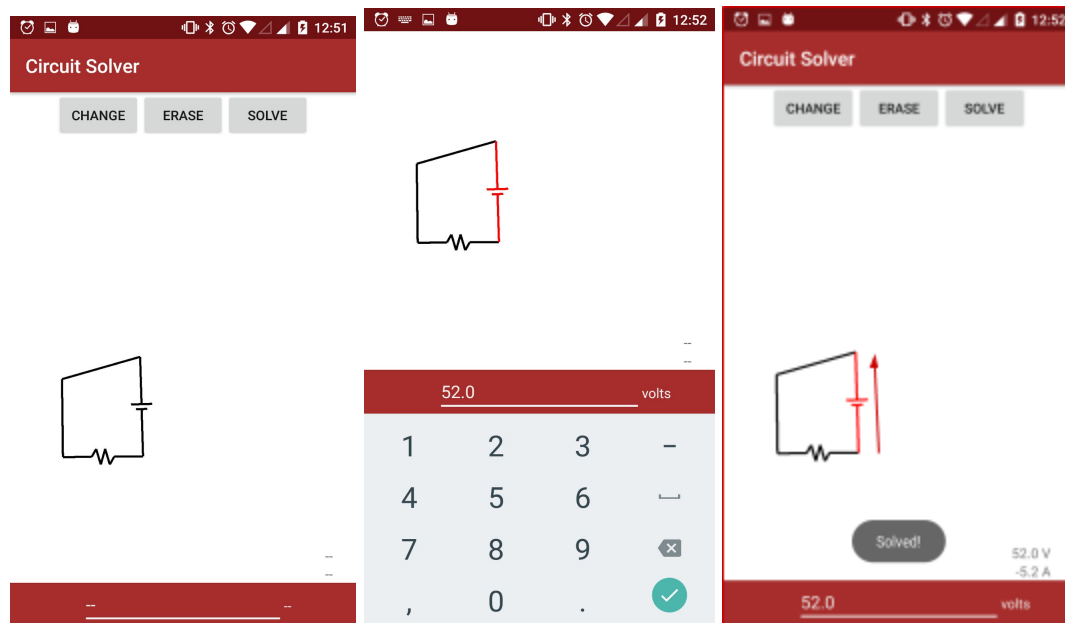
43. User selects the desired circuit image saved on their device

44. Go to basic flow line 8

45. **Post Conditions:** Same as basic flow

39. **Prototype Screen:**



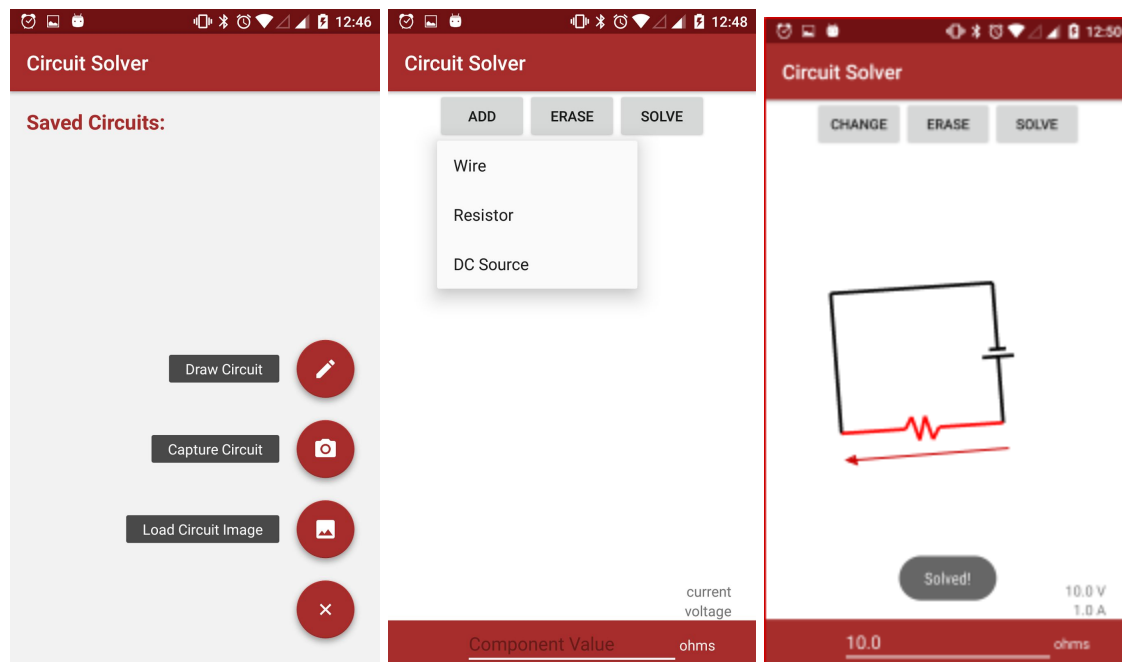


CircuitSolver Use Case Document

USE CASE 4: Create virtual circuit by directly drawing in the app

1. **Brief Description:** The circuit is drawn directly in the app
2. **Preconditions:** User is on the CircuitSolver home page
3. **Business Trigger:** The user doesn't have the circuit on paper and wants to create it directly in the app
4. **Basic Flow:**
 5. User selects the "Draw circuit" button
 6. Application opens a blank circuit screen where a circuit can be drawn
 7. User pinches the screen to zoom in and out from the circuit to position the view adequately so that the component can be placed correctly
 8. The user taps "Add"
 9. The application displays a dropdown list of circuit component options
 10. The user taps on the desired component
 11. A message appears informing the user of their selected component
 12. The user draws the circuit component on screen by tapping to anchor the one end and then dragging the other end of the component to its desired location. This determines the direction of components for which it matters (e.g. voltage sources)
 13. Steps 7-12 may be repeated
14. **Post Condition:** The virtual circuit is drawn and is saved and can be solved or modified as desired later
15. **Alternate Flow: Edit component(s)**

16. When in line 12, the user makes a mistake while drawing a component (e.g disconnects components accidentally or overlaps components)
17. The user taps on component he/she wants to delete
18. The component turns red, the “Add” button changes into an “Change” button, the displayed value in the textbox at the bottom of the screen switches to the value saved for the selected component and the displayed value’s units changes to match the component (ohms for a resistor, volts for a voltage source etc.)
18. The user presses the “Erase” button
19. The application deletes the selected component and it is not visible any more on the screen
20. Steps 17-19 may be repeated for other offending components
21. Go to step 8 and continue basic flow.
22. **Post Conditions:** Same as basic flow
23. **Prototype Screen:**



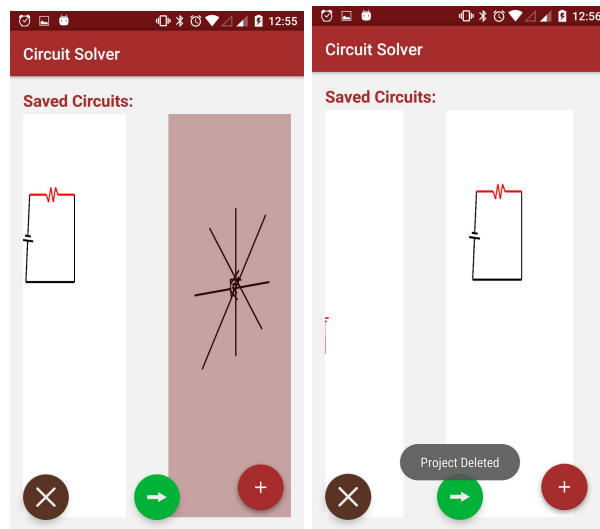
CircuitSolver Use Case Document

USE CASE 5: Delete saved circuit

1. **Brief Description:** Delete a saved circuit
2. **Preconditions:** User is on Home screen
3. **Business Trigger:** The user has a saved circuit which they want to delete
4. **Basic Flow:**
5. User taps on the saved circuit they want to delete
6. Application highlights the tapped circuit and displays an option to delete circuit at the bottom of the screen
7. User taps on the delete option

8. Application deletes the circuit's files and removes it from the display on the Home screen and a notification appears to notify the user of the deletion
9. **Post Condition:** The selected circuit is deleted from the application

10. Prototype Screen:

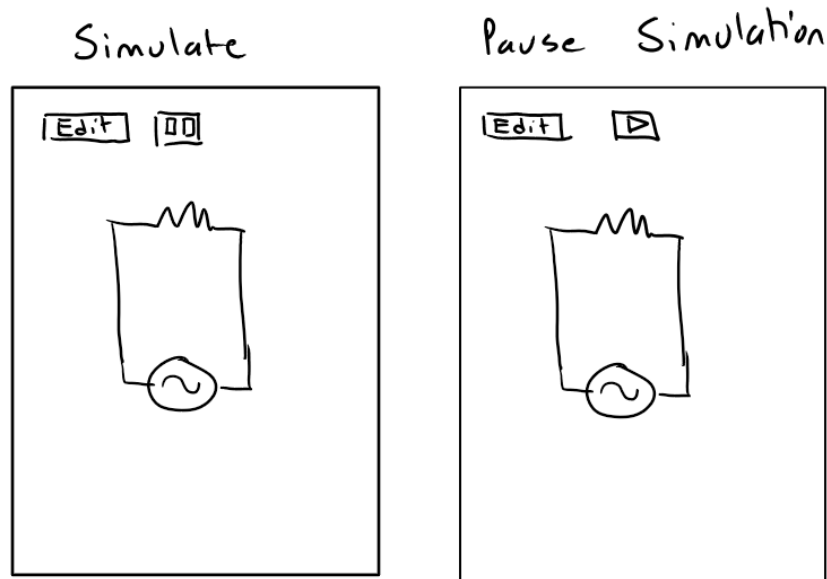


CircuitSolver Use Case Document

USE CASE 6: Circuit Simulation (not to be implemented in CPEN 321)

11. **Brief Description:** Perform AC Circuit analysis and simulation on circuit
12. **Preconditions:** User is on the Edit Circuit screen and has drawn a complete circuit
13. **Business Trigger:** The user wants to simulate their circuit over time
14. **Basic Flow:**
 15. User taps "Solve"
 16. The "Solve" button changes to an "Change" button, and application also displays a "Pause" button. In the backend, the system begins sending NgSPICE request periodically to retrieve analysis of circuit at different time steps.
 17. User selects a circuit element.
 18. Application displays the solved value of that circuit element in real time.
 19. User taps the "Pause" button.
 20. The "Pause" button turns into a "Play" button. Application stops periodically sending requests to NgSPICE. Analyzed voltages and currents are frozen.
 21. User taps "Change" button.
 22. Application displays regular Edit screen, as it was before the use case was triggered.
23. **Post Condition:** The application displays the values and current direction of a solved circuit for each tapped circuit component
19. **Alternate Flow: Edit component(s)**
 20. When in Step 11, the User taps the "Play" button again.
 21. Go to Step 7 in basic flow.
24. **Post Conditions:** Same as basic flow

25. Prototype Screen:



26. Note: This use case was not in the scope of our initial release of CircuitSolver for CPEN 321. However, the app was designed with this use case in mind. As NgSPICE is capable of analyzing circuits per unit step of time, it is a matter of modifying the UI and implementing code which periodically sends requests to NgSPICE in order to satisfy this use case.

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.