

Amplify APP

Niko Hildenbrand

Ian Chadwick

Hank Zhong

Joe Biernacki

Project Links:

https://drive.google.com/file/d/1FpHksqqSj4S3_XtXdJJapdoQOSJj6R1/view?usp=sharing

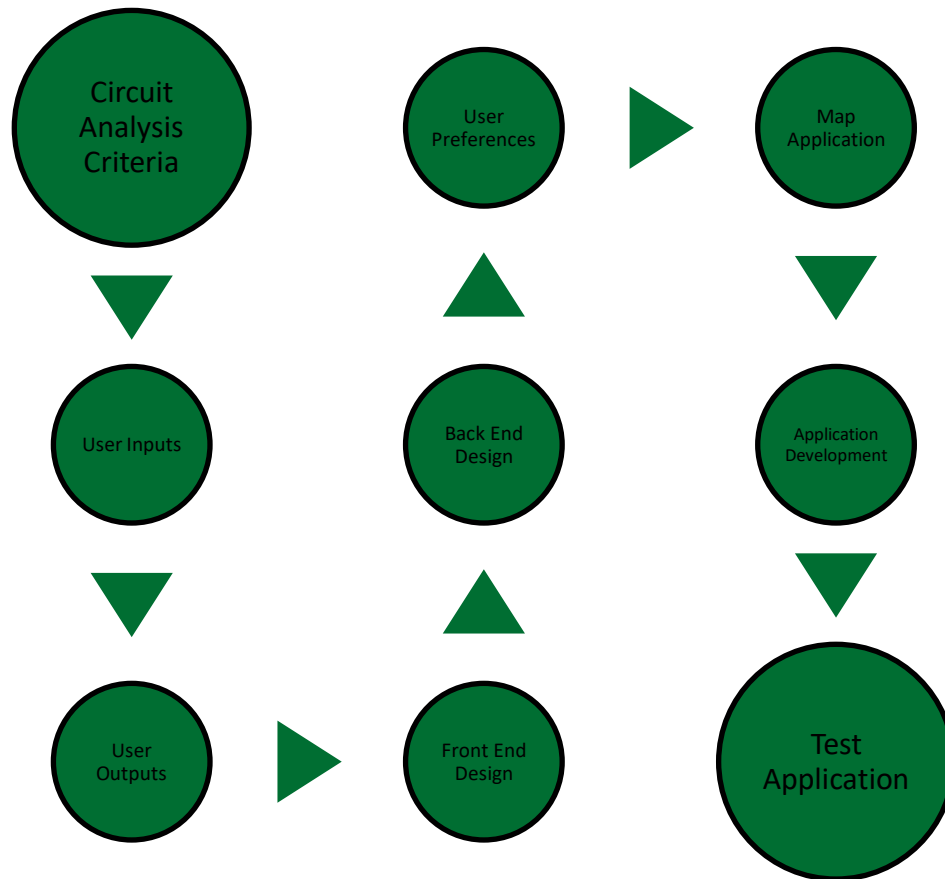
<https://github.com/ianjchadwick/AmplifyApp>

<https://youtu.be/19TK7hHoozk>

Overall Software Architecture & Component Description

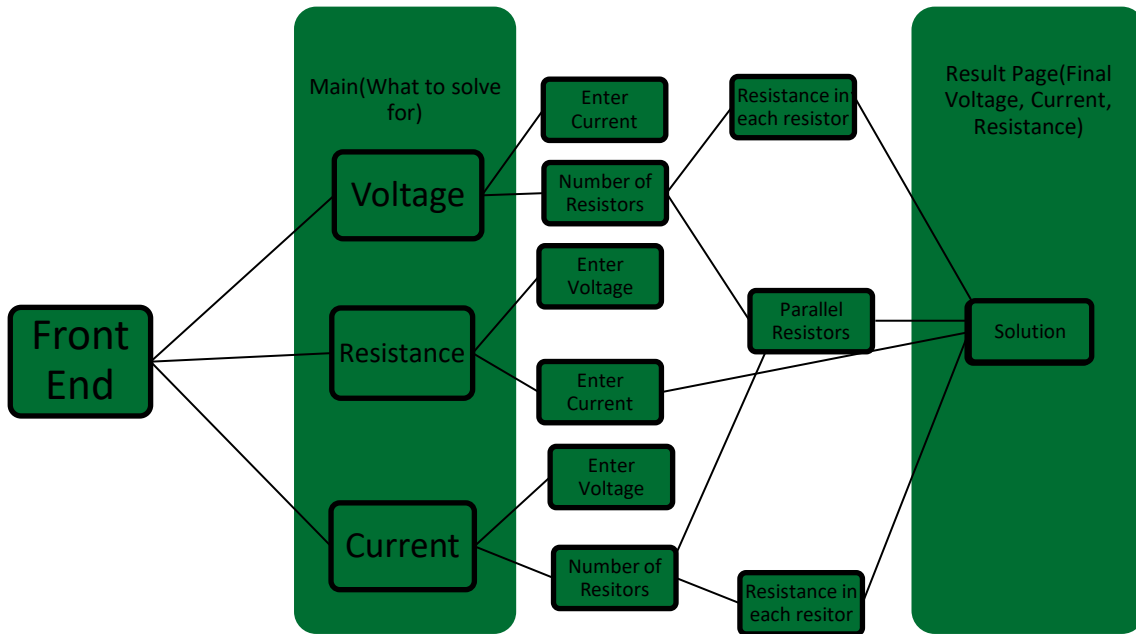
The overall Software Architecture and description of the components of the application has been divided into two different stages. The first stage is the overall Project Structure and components needed. This will describe, through the use of a flow chart, the steps taken to arrive at the development of the application. The second stage will describe the actual application development itself. This will utilize a diagram for the GUI along with a description of how classes work with each other to allow the application to work. Ultimately, the objectives and approach when designing the software architecture needed to create a successful application are outlined in this section.

Stage 1



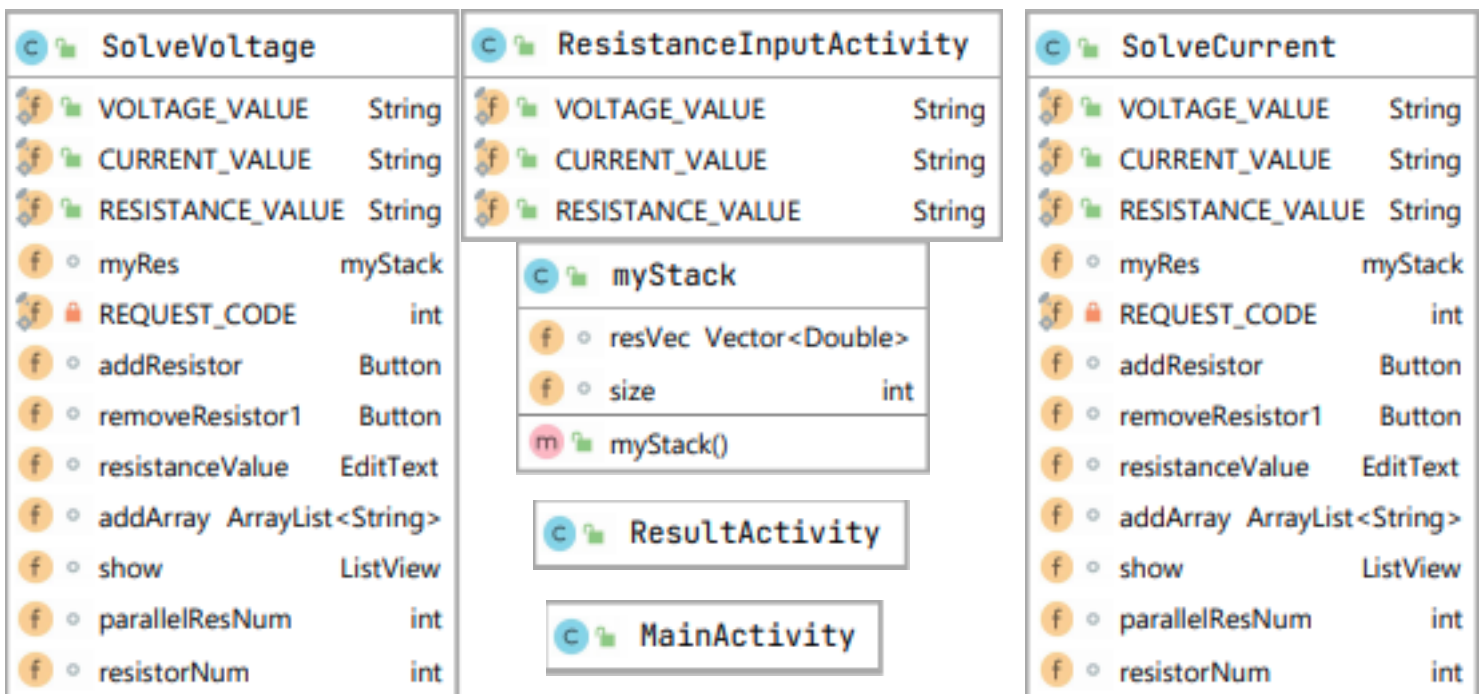
Circuit Analysis Criteria was the first step in determining how to tackle the design and implementation of this application. The balance between usability and complexity to arrive at the desired outcome was weighed heavily as a very user friendly experience was a top priority. User inputs were determined to be voltage, current and resistance on the initial page shown, with the assumption that the user wants to determine one of these values. After this page users are given the ability to add resistors, and determine if they want to calculate them when they are in series or parallel, each adding on to each other. With this information taken in, user outputs were determined to match initial user inputs, only having the missing value computed and outputted as well. The next step of Front End design was taking into account these inputs and outputs and planning a GUI that emulates this clearly. The Back End plan is to design a code in java to support the front end and the inputs and outputs planned for the project. The completion of these steps are vital to the project and can then enable the user preferences task where we implement user friendly options, such as drop down menus and enhanced visuals. All of these previous steps lead to mapping of how we are going to design this application. With this outline, Android Studio is used to implement. The last part of the Project structure is to test the application for functionality. The design will be crisp and clear and present the user with the exact outputs they desire.

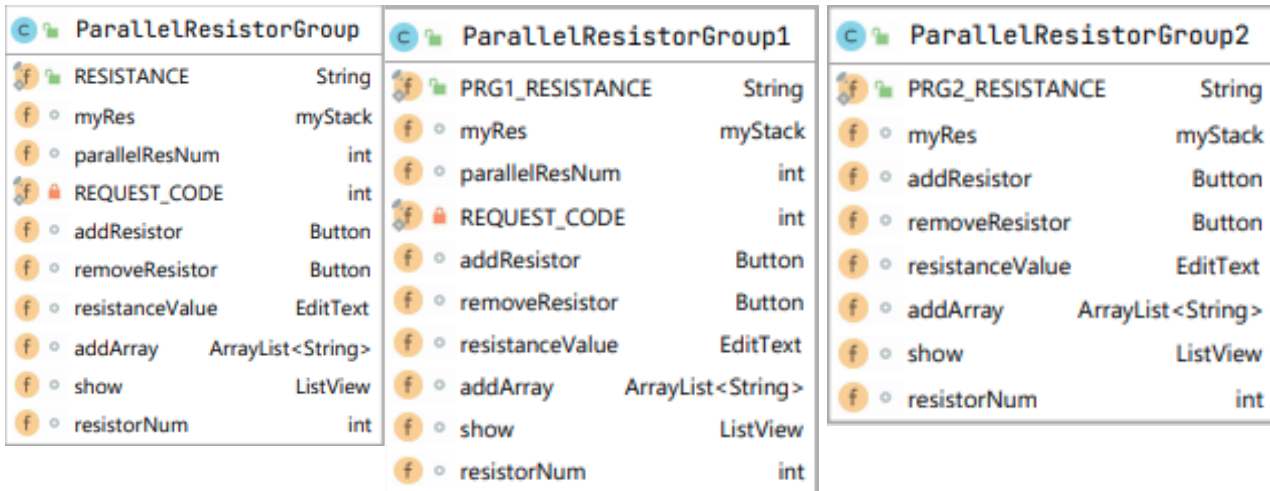
Stage 2



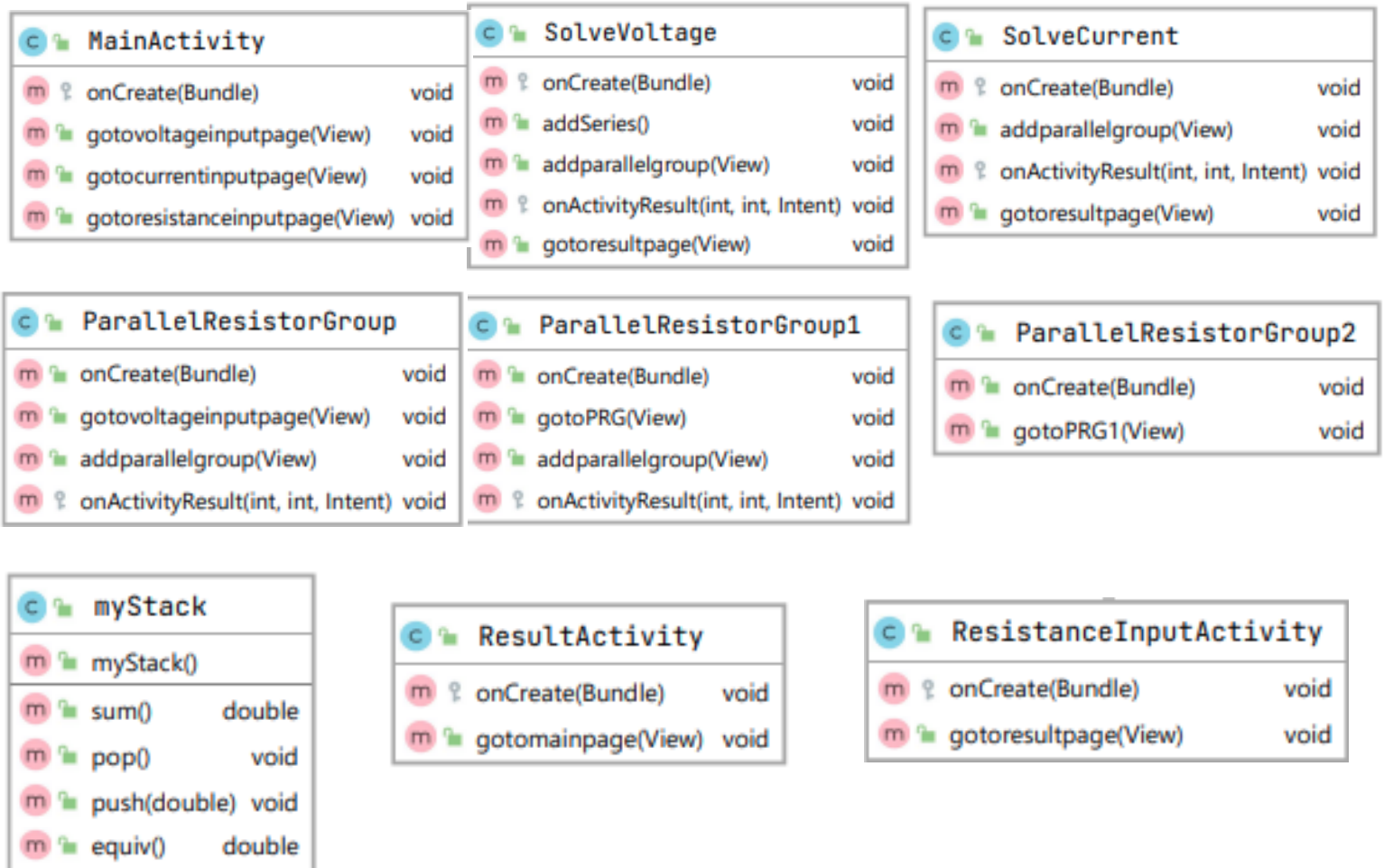
The diagram above shows the front end GUI component, its layout, and the logical process behind the design. The idea behind this design was to easily integrate a complicated process of circuit solving for a user friendly experience. The app prompts the user for which component of a circuit they would like to solve for. This includes either voltage, current or resistance. Depending on this choice, the user is shown a new screen giving them the option to enter values for the two components that are known. If solving for voltage or current, the user is given the option to add resistors in parallel or in series. The complexity of this application comes in to play if multiple resistors are added, with each either being in parallel or in series with one of the previous resistors. Once all data entry has been completed, a results page is given with the values of all three components and a diagram of a circuit with those values included. The option to start over and re run the application is given at the bottom of this page.

UML Attributes





UML Methods



Application Overview

Amplify is an app for Android that performs a simple circuit analysis. Users will be able to enter information about their circuit in to the application and the app will perform the circuit analysis and solve for the unknown values. Amplify is an app that is vital to help solve for voltage, resistance and current. We are targeting an audience of mainly electrical engineers or electricians. It is a simple application that shows the relationship between the three major numerical components of a circuit and can be utilized by these groups of people. In addition, it can be manifested for use by students of all ages in either physics or circuit classes to get an idea of how these components relate to each other. The big difference between this app and other circuit analysis applications on the market is that the interface of Amplify is extremely user friendly. As we began looking at other similar applications, it was noted that the reviews continued to vent frustrations with the complexity of the interface. This refers to both the ability to input data and the over complexity of a getting an output. Why make it complicated for users? We want our users to be able to enter the values they know and get a clear and crisp output that makes sense. This is not about jumping through hoops, it is about arming users with data immediately.

Task Distribution

Niko Hildenbrand:

- Set up the timeline, milestones, and objectives.
- Interfaced with all team members to maintain progress during development, including calling and organizing meetings.
- Made sure that all required work was completed and turned in on time.
- Tested all code for efficiency and ease of use in Android Studio.
- Made all documentation for timelines, objectives, and meetings.
- Created and completed the Project Report
- Created Project video
- Created Statement of Work for all members to agree on

Joe Biernacki:

- Checked all design specifications for system-level problems and performed system-level integrated testing.
- Commented and made readability of the code easy.
- Tested all code for efficiency and ease of use in Android Studio.
- Back end software architecture description.
- Figured out what data should be passed between the GUI and the backend code (C/C++/Java) and implementing the routines that will do it.
- Tested routines to demonstrate the proper passage of data.
- Converted C++ code to Java for ease of integration into Android Studio
- Ensured efficiency, robustness and basic securities.

Ian Chadwick:

- Checked all design specifications for system-level problems and performing system-level integrated testing.
- Commented and made readability of the code easy.
- Tested all code for efficiency and ease of use in Android Studio.
- Back end software architecture description.
- Figured out what data should be passed between the GUI and the backend code (C/C++/Java) and implementing the routines that will do it.
- Tested routines to demonstrate the proper passage of data.
- Converted C++ code to Java for ease of integration into Android Studio

- Ensured efficiency, robustness and basic securities.
- Demoed app in final class to Professor

Hank Zhong:

- Created a crisp and intuitive design that made the learning curve for the user extremely minimal.
- Thoroughly tested front end code and provide comments/code for easy testing.
- Tested all code for efficiency and ease of use in Android Studio.
- Designed both a mock up GUI and final GUI that allows users to easily use the application
- Created clear and complete error messages if input was not correct.
- Front end software architecture description.
- Create a robust design that does not crash when utilized.
- Created a visually appealing application that utilizes icons and backdrops to improve the users experience.

Effort Breakdown

Niko 25%

Ian 25%

Hank 25%

Joe 25%

Timeline of Meeting Progress

11/13/20 – 5pm

Discussed design ideas and decided on project. The project was determined to be Amplify. The user defines the following parameters when prompted: The battery voltage, value of resistors, number of resistors and is each resistor in series or in parallel to any of the previous resistors. The user will be able to click a button to add each additional resistor. This app should be able to solve any circuit with just resistors. Display visually the final values of current, voltage and resistance in one uniform final output page, with the option to return to the beginning and re start without having to close the app.

By next meeting:

- Run by the TAs the idea of the project and get feedback.
- Map out idea of design
- Decide on final inputs and outputs for users and the layout of this
- Have a working awareness of how this app should work and run
- Decide who takes what tasks for the project and assign roles

11/22/20 – 11am

Images below are planned for implementation into Android Studio. Went over exactly what needs to be designed within Android studio and how this would look for the user. Discussed the idea of buttons for users to choose what should be implemented in their circuit design and how to implement this in Java.

By next meeting:

- Gain familiarity with Java and programming in Android Studio.
- Begin implementation of classes and functions and determine how that code will work together.
- Determine how buttons should look in GUI
- Have a running list of all meetings and milestones

+ Add a voltage source

↳ V source voltage (Volts)
 // user-defined
 // only 1
 // Creates a voltage source object

+ Add a series resistor

↳ R₁ value Ohms

+ Create a group of parallel resistors

↳ R₂ value Ohms

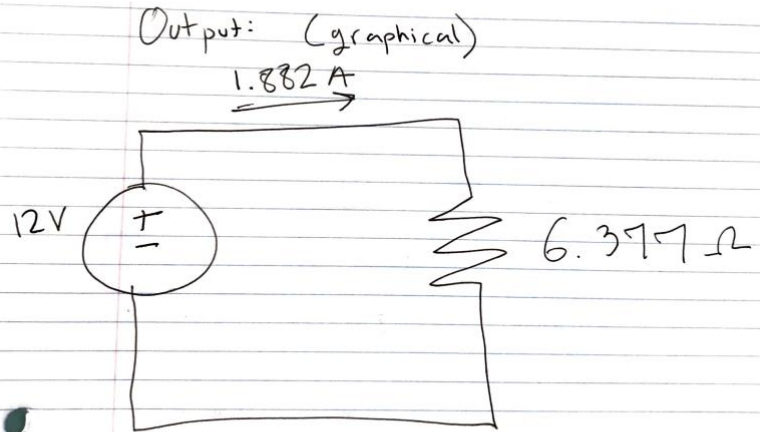
↳ Dropdown

↳ shows list of resistors you've already created

// creates a "group of resistors in parallel"

+ Add a current source

↳ I source current (Amperes)



11/25/20 - 10am

Went over a mock-up of a designed GPU for the project. Discussed multiple options for user inputs and how to display this in app. Going to look into way of creating more inputs when a user inputs a number such as number of resistors. The idea of a dropdown box was mentioned and how to implement this in this GUI. Assigned milestones to have completed by next meeting. Ian and Joe will meet up outside together as they are splitting both tech and spec roles.

By next meeting:

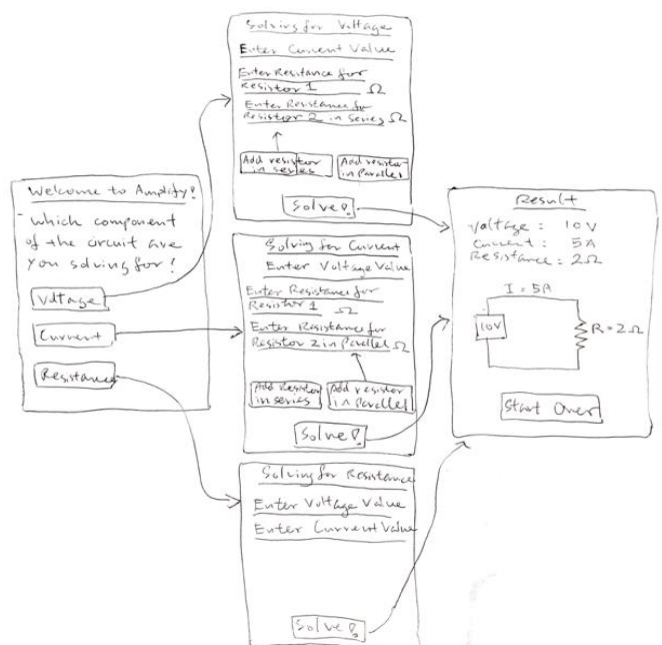
- Ian and Joe work on back end and create a class for inputs
- Hank work on front end of graphics.
- Niko look into integrating back end into Android Studio

11/29/20 - 7pm

Code for both the Front End and Back end has been written and GUI has started to come together. Ian, Niko and Hank will go to this week's office hours to check in with TA's. It was determined that Niko will make the project video at the end. Discussed GitHub code and the need to finish as soon as possible. Have to debug and clean up the code to create resistor classes. Discussed a mockup of the app interface shown on the side and determined how we can implement this into Android. In addition to this, Niko responded back to Professor Coskun's email saying Ian will demo the Project.

By next meeting:

- Clean up resistor classes and go over GitHub code to debug
- Begin work on putting together the whole project report
- Have each group member begin writing up small description of the component they are working on



11/30/20 – 5pm

Went over feedback received in TA checkpoint and how to efficiently make this work by the Thursday deadline that we set to have app up and running. Discussed how to simplify GUI and store the input received from the user. Talked about the most complicated design we could receive from the user and how that would be handled and if we want to make limitations on amount of resistors added and whether they can put each additional resistor in series or parallel with the previous resistor.

By next meeting:

- Convert current C++ code in java for back end design
- Figure out how to save activity state as is when user moves to another screen
- Figure out how to create an object in another activity when button is pressed in the current activity.
- Figure out how to store values from EditText views created by the 'add Resistor' button.
- Have at least 80% of Project report completed to go over with team and get feedback on things to change

12/4/20 – 10am

Ironed out all remaining details about how to get the app up and running. When the button to add a resistor is clicked, the add resistor function adds a string value which is then put into the array list. Discussed the current problems with the array lists that we have in the GUI. Talked through about how to give each resistor value a new ID to be able to have a new string ID object added into the array list. Used getID to do this. Discussed when we add additional resistors and how do we have all of those additional resistors then be added together and have one value put on previous screen. Also talked about adding delete buttons to additional resistors to be able to delete the object from the array list.

By next meeting:

- Have 95% of project report completed for group to approve
- Determine how to add user data into array list
- How to take the array list and put it into algorithm and get output back
- Based on the array list how do you re call that data once you move back levels in the GUI
- Implement a back button on each page

12/6/20 – 1pm

7th meeting- Met in a zoom meeting and stayed online for 5 hours while each member worked on their portion of the remaining project. After stripping down the application to make sure it runs like intended, configured the GUI to put needed data into correct areas. Finalized app and made sure things run smoothly for demo. Added the project report to GitHub to have all group members go over and approve this. If anything needs to be changed in it, those changes will be made before the deadline. Zipped all necessary files in preparation for final turn in.

By Monday Deadline:

- Update the README file on GitHub to explain how to run the app
- Finalize app for a crisp clean experience
- Finish youtube video
- Complete final draft of project report