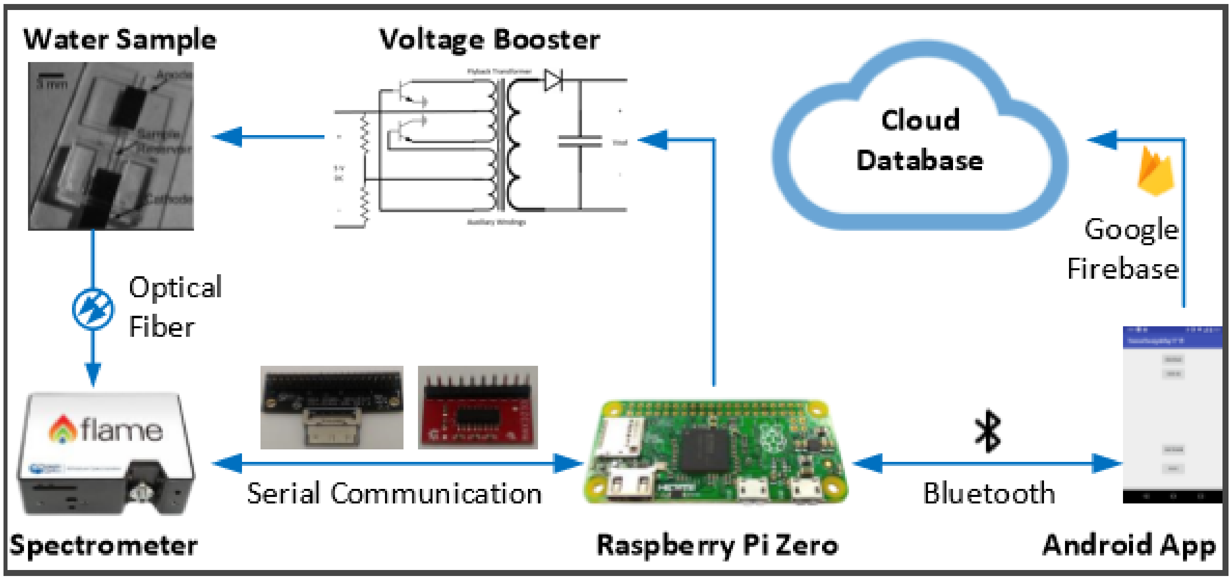
Portable Nutrient Data Collection System Phase II

May1719
Client – Dr. Long Que
Advisor – Dr. Daji Qiao
Ben Theisen – Team Lead
Michael Rupert – Web Master
Ben Engebrecht – Communication Lead
Logan Boas – Communication Lead II
Zakk Belloma – Key Concept Holder
Ryan Young – Key Concept Holder II

Contents

Revised Project Design



1.1 PROJECT STATEMENT

This project deals with a portable system that will be used to analyze water in the field. The purpose of this is to determine whether there are certain nutrients in the water in certain locations. This system has two methods of testing the water. The first is sending a high voltage across a small reservoir of water that essentially vaporizes the water creating a light that goes into a fiber optic cable and into a spectrometer. The second method is shining a strong enough light through the water sample that then goes into a fiber optic cable connected to the spectrometer. After the data is produced by the spectrometer, it will be sent through our system and out to an app for the user to easily view the data.

1.2 PURPOSE

This project could be very beneficial in several capacities, most notably in agriculture and environmental sciences. Agriculture could use this system to ensure that the water they are using is free of any harmful chemicals. This could help farmers gain better yields from crops, use their resources more efficiently (not harming crops with bad water and having to replace them), and ultimately have better financial results from their business. Environmental studies would be able to use this system to analyze potential pollutions in hard to reach areas with other equipment. With this portable system, they would be able to keep a watch on problem areas. Also, using this system, with the analysis of the spectrometer these scientists would know exactly what is causing problems in the environment. This means that they should be able to find a solution quickly. Society would benefit greatly from both uses of this project. There would be more food coming from farms and that food would be free of the harmful chemicals that would be detected by the system. Environmentalists finding pollution in water sources and dealing with the issues would lead to a cleaner environment and creating a cleaner place for people to live.

2.1 SYSTEM SPECIFICATIONS

The system needs to be reasonably portable. It can be assumed that a user will have a bag of some sort to carry the system in the field. Still, form factor is one of the deciding factors that we have used in selecting what components to use. Another assumption is that the user will have access to power to charge the system after a day of use. Currently the app is being developed for Android systems, so it will be assumed that the user will have some sort of android device capable of Bluetooth connection with a system version of 4.0 or more recent.

2.1.1 NON-FUNCTIONAL

-Interface should be simple and have a tutorial

Tutorial should be able to walk users through the system for them to use it correctly.

-Entire system needs to be portable

Users will need the system in a field and should be able to carry easily.

-System should last for 50 trials on one battery life

Users in the field should be able to perform multiple tests on one battery charge.

-Reasonable wireless range

The user should be able to have their phone with the app

-Data acquisition in less than 5 seconds

The entire process from start to display the data on the app should take less than 5 seconds. This will depend mostly upon how fast the spectrometer performs its function.

-One battery charge should last for a full day in the field

The battery should be able to last for a full day, or until the user performs 50 tests.

2.1.2 FUNCTIONAL

-Weatherproof/Weather resistant

This system is going to be used in the field around water and potential weather. The system will need to be able to deal with

-Wireless connectivity/Data transfer

The system needs to be able to wirelessly transfer data to the app that we are developing.

-Store data from each sample in a database

Storing data from each sample in a database will allow for the users to compare data from different samples taken in separate locations or the same location over a period.

-Test multiple samples

The system should be able to test multiple samples and should be able to store the data from each sample separately.

-Display data in human readable fashion

The app should display the data from the samples in a way that the user can read.

Raspberry Pi Zero – Our selected microcontroller. Raspberry Pi's can run full Debian Linux. The Pi Zero has a 1GHz, Single-core CPU, 512MB RAM, and 40 GPIO headers4. Dimensions of the Pi Zero are 65mm by 30mm, so it is a very small microcontroller and this will help with portability. The serial communication software running on the Raspberry Pi will not be multithreaded, so a single-core CPU is sufficient for our needs. The entire board costs $5 on its own. GPIO pins, specifically the Tx and Rx pins will be used for serial communication.

Voltage Booster

The voltage booster we selected is a version of a flyback converter circuit. That boost 5 V DC input voltage to a range up to 400 kV DC. The output voltage is controlled by the length of the air gap is between the positive lead and the path to the negative lead. A PWM from the raspberry pi switches the voltage booster on and off as needed to minimize wear and tear on the voltage booster and power consumption. The

Micro-Discharge

The team in phase one designed this device to test the sample of water by arcing the boosted voltage through the water sample between an anode and cathode. The light that is produced due to this arc is detected by an optical fiber and the fiber carries the collected data to the spectrometer to be analyzed. The light of the arc is important because the light emitted by the electricity passing through each specific element occurs at a specific frequency.

Serial Communication

We are using our Raspberry Pi connected to an Ocean Optics Spectrometer along with the Pi4J library to test serial communication. These devices and library are necessary for us to perform the serial communication and collect data from our spectrometer. We also plan to use the OceanView Software, which is an Ocean Optics Software that allows us to view data from the spectrometer to verify our collected data is correct.

Bluetooth communication

To send data from the system to the android app, we will be using Bluetooth. This allows us to have a wireless connection between the system and app which provides flexibility to the user. This communication will be achieved with a Bluetooth dongle.

Firebase

This is our solution to the database demand of this project. The system will be used predominantly in the field. This means that there may not be any sort of connection to the internet during testing. Even though there may not be a connection to the internet, the app will still store sample data in a local database. Once an internet connection is established, the sample data from the local database can be uploaded to the cloud using the Firebase API.

Smartphone app

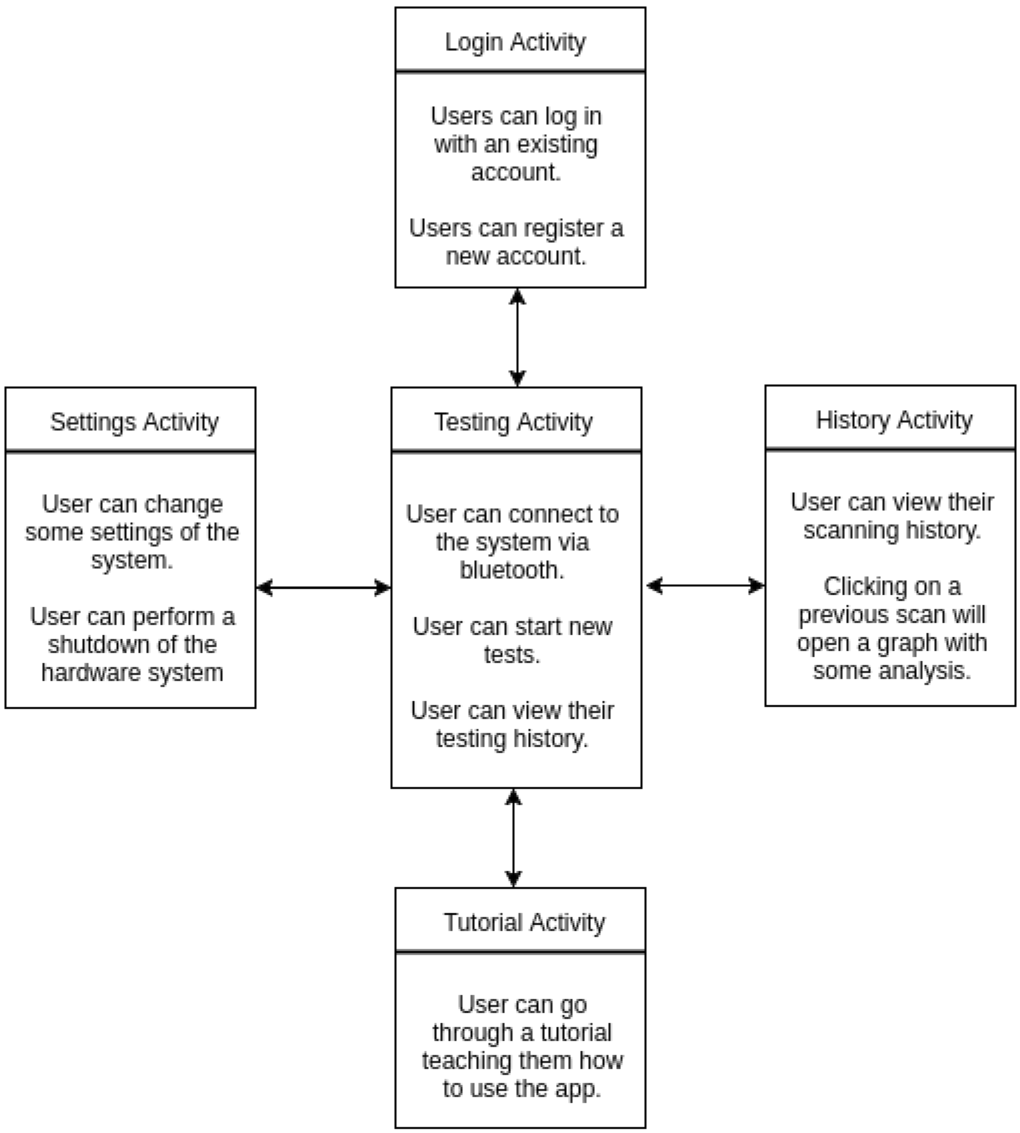
An app allows users to view data right away in the field. This can be useful in many ways. It could increase efficiency of the device by allowing users to know if they are testing in the right place. The app will have the ability to control the spectrometer in the system through Bluetooth and serial communication, store data from samples in a local database, and also upload that data to the cloud when desired. Uploading data to the cloud will allow users to analyze the data further and with more powerful systems than a simple smartphone app, leading to a better result from the system.

Power

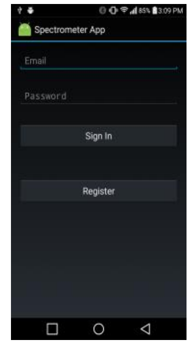
Our solution for powering this device needs to be relatively portable, considering that one of the functional design specifications for this project is that it needs to provide portable analysis. Thus, a 13,000mAh battery with two USB discharge ports was chosen. This provides us with two 5V power supplies with a total of 3A. We measured the Raspberry Pi and spectrometer and discovered that they both draw around 350mA. This mean that this low-cost battery pack can power the device for a total of 37.14 continuous hours, which is more than enough for this design. The voltage booster draws 2 A while operating, but it is only on for 0.05 seconds. This means the voltage booster's power consumption is much smaller than the raspberry pi's power consumption.

Implementation Details

Application Flowchart



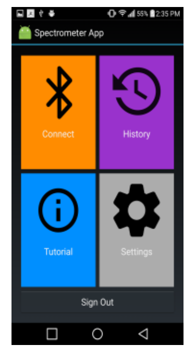
Login Activity:



Introduction: This activity allows a user to log in with an existing account, or register a new account. Using these accounts, we can keep track of a user's scanning history.

Details: User authentication is done using java and the Google Firebase Authentication API. Some validation methods are in place to ensure that a user has entered both an email and password in, and that it is a valid email before attempting to log the user in, or register a new account.

Testing Activity:

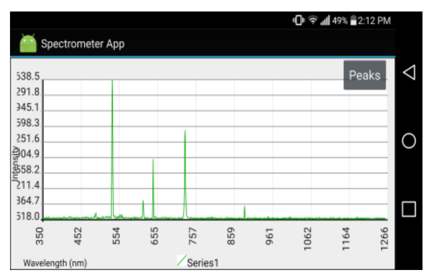


Introduction: This is the main functional activity of the application. It allows a user to create a Bluetooth connection with the system. And once that connection is made, perform scans on water samples. In addition, buttons to access the other activities and sign out are on this page.

Details: This page is coded in java utilizing android application APIs and components to keep the activity updated while scans are performing. This is so that the user does not think that the app has frozen while a scan is underway.

History Activity:

Introduction: This activity gets the scans that the current user has taken. It will sort those scans by the time taken. Clicking on a certain scan will open the graph for that scan and tell the user what wavelengths had peaks detected.



Details: This activity uses java and Firebase Database API to allow the user to see their scans. For the graphing, android graphing libraries are used. The data is analyzed when the page is opened to detect peaks on the graph. This was a decision made to speed up the scanning process.

Settings Activity:

Introduction: This activity allows the user to change some settings of the hardware. Specifically, the integration time and wavelength that the spectrometer returns. Additionally, a shutdown button is implemented to allow for a safer shutdown of the physical system.

Details: This activity utilizes java and Instream and Outstream objects from the Bluetooth connection to transfer data. The shutdown button runs a safe command line shutdown command on the raspberry pi. This helps prevent data corruption on the Raspberry Pi's operating system.

Tutorial Activity:

Introduction: The tutorial shows users each button and explains the functionality of the button. Additionally, it explains what each activity is used for.
Details: The tutorial is created using Material Showcase. This tool allows for a building of a clean and concise tutorial.

Web viewer:

Introduction: The web viewer is a web application used for viewing all the scans in the database. This database is created using Firebase which is a cloud database that also allows for the storage of local data when there is no internet connection.
Details: Created in react.js, the web viewer will live update when scans are uploaded to the database. Additionally, scans can be deleted from the database using this web application. This allows a user's to keep their database clean of unwanted scans.

Hardware System:

Introduction: The hardware system is the center of this project. It achieves communication between the Android application and the spectrometer using Bluetooth and serial communication. Additionally, the hardware controls the voltage booster which is used to get the actual data from a water sample.
Details: The hardware system is comprised of a Raspberry Pi, Max3232, Ocean Optics breakout board, LEDs, portable battery, and voltage booster. The Raspberry Pi runs a full Linux operating system and on startup runs a server code program that we wrote. This program accepts connections from the application and facilitates data transmission between the components. Using the GPIO pins, we light LEDs depending on the status of the system, and activate the voltage booster for enough time to get a good scan read by the spectrometer.

Testing process and testing results

For testing our project, we started by testing the components separately. Once we felt comfortable with the performance of each component we began to connect the components and test that they still worked. This process was re-iterated until each component was added into our system.

Starting out, serial communication was our primary focus. For this testing we had the spectrometer communicating to our laptops through rs232 serial communication. We tested that the communication was working by sending commands to the spectrometer and monitoring whether we were getting expected results back. Through this testing process we had to do a lot of rewiring until the communication was working as needed. The result lead us to successfully communicate to the spectrometer.

Next, we worked on making serial communication work through our raspberry pi. This involved creating python scripts to send commands to the pi and read the results back. Through this testing process the wiring was less of an issue, and debugging the python scripts to a functional state was the objective. After debugging we successfully had serial communication working from the pi to the spectrometer.

With serial communication working as expected, we moved forward by making Bluetooth communication work between the raspberry pi and our android devices. This involved creating server scripts on the raspberry pi to wait for a Bluetooth connection to be established. Once we had the connection working we tested that data could be sent between the two devices by sending mock data and mock commands. This process again involved debugging our code until things worked.

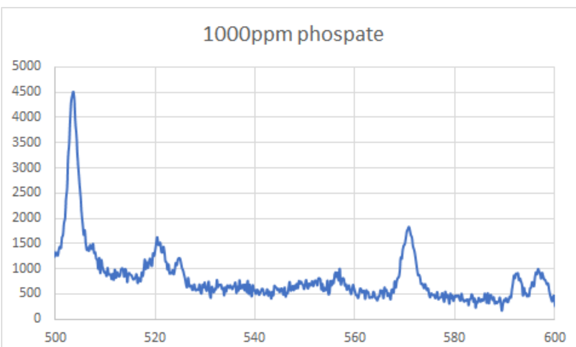
With both Bluetooth and serial communication working separately, we combined both components to establish full system communication. This phase involved adding commands to our server scripts that when recognized would request data from the spectrometer and send the results back out through Bluetooth. To make sure this worked we tested that commands were being received correctly, data was being collected correctly, and data was being sent back to the phone correctly. After debugging and multiple trial runs we felt comfortable with the system communication.

Once the system was working, we next made the system run headless. In other words, we made the server scripts on the raspberry pi run on bootup. We also added controls to shut down the raspberry pi from the application. To test this, we incorporated LED lights that indicated the state of the server code being run. We also redid the tests used before to make sure that results were coming back to the phone as expected. This made our system usable without needing to boot up the server manually.

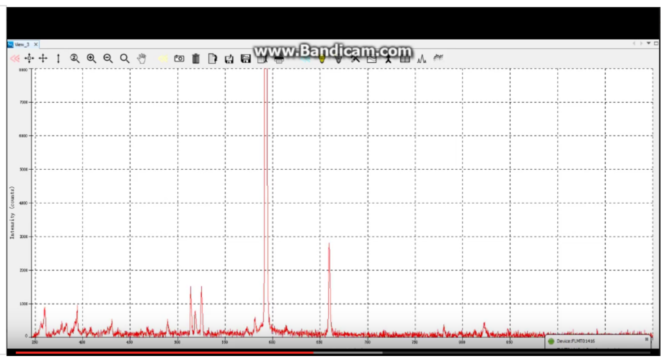
With the system communication working headless, we moved forward by saving our data to our firebase database and making the data viewable through a graph both on the phone and on a web viewer. Testing this involved taking scans from the spectrometer and viewing the output of the scans in both the database and on our graphs. We further tested this by comparing our results with results made available from a application provided with the spectrometer. The results confirmed that our communication was working and the data was being collected and displayed correctly.

The last component we needed to incorporate was the voltage booster. We tested this by setting up a controller circuit and recording the voltage produced when the voltage booster was triggered. These tests proved that we could control the voltage booster and the voltage booster was outputting a sufficient voltage for our spark. With the voltage booster working, we added in its controls to the raspberry pi through the GPIO pins. We made sure we could trigger the voltage booster by activating and deactivating the GPIO pins and then incorporated the voltage booster into our scan command for the spectrometer. Our final tests involved using the voltage booster across water samples and collecting the results through our system. These results were compared with the expected results made from using the spectrometer software to confirm that everything was working.

From our testing, we confirmed that each part could work independently, and then confirmed that the parts could work together. Further testing was done to confirm that they system as a whole worked and was producing the expected results. The expected results were confirmed by comparing them with results produced by software available with our spectrometer. These tests proved that the entire system performs as it should.



Our System



Results from Provided Software

Appendix I: How to operate the system

Our system is designed to be as simple as possible to use by the end user. There are two systems that the user will have to understand how to use and set up, The enclosure and the Android app. Section 1 will go over setting up the enclosure, and section 2 will cover using the android app to operate the system

Section 1: Enclosure

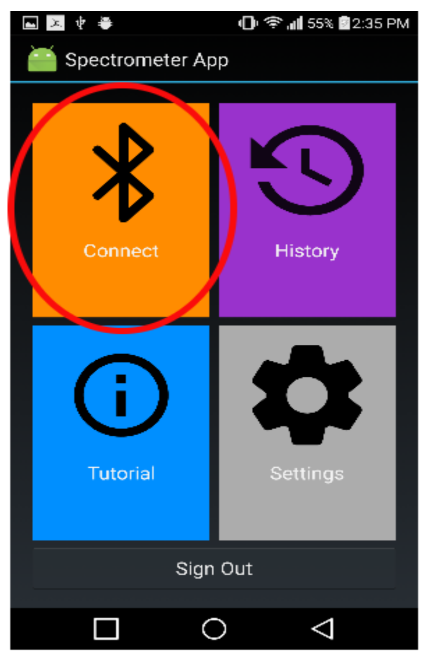
. (The LED array refers to the green, yellow and red LEDs connected to the Raspberry Pi)

1. Ensure the systems battery is charged, if it is not charge the system using a standard phone charger by plugging it into the mini usb port on the enclosure.
2. Plug the breakout board into the appropriate slot of the spectrometer. This will give the spectrometer power, and complete the serial communication connection.
3. Power the system on by flipping the power switch to the on position, and wait for the yellow LED mounted on the enclosure to power on. This ensures the system is ready to operate. Once the app is connected to the hardware via Bluetooth, the green LED will be lit. And once a scan is running, the red LED is on.
4. Load a water sample that you would like to test into the tray.
5. The enclosure is now ready to begin the testing procedure, proceed to section 2
6. Turn the system off when done scanning by first performing the shutdown command from the settings page in the app, and once all of the LEDs are off, flipping the power switch to the off position.

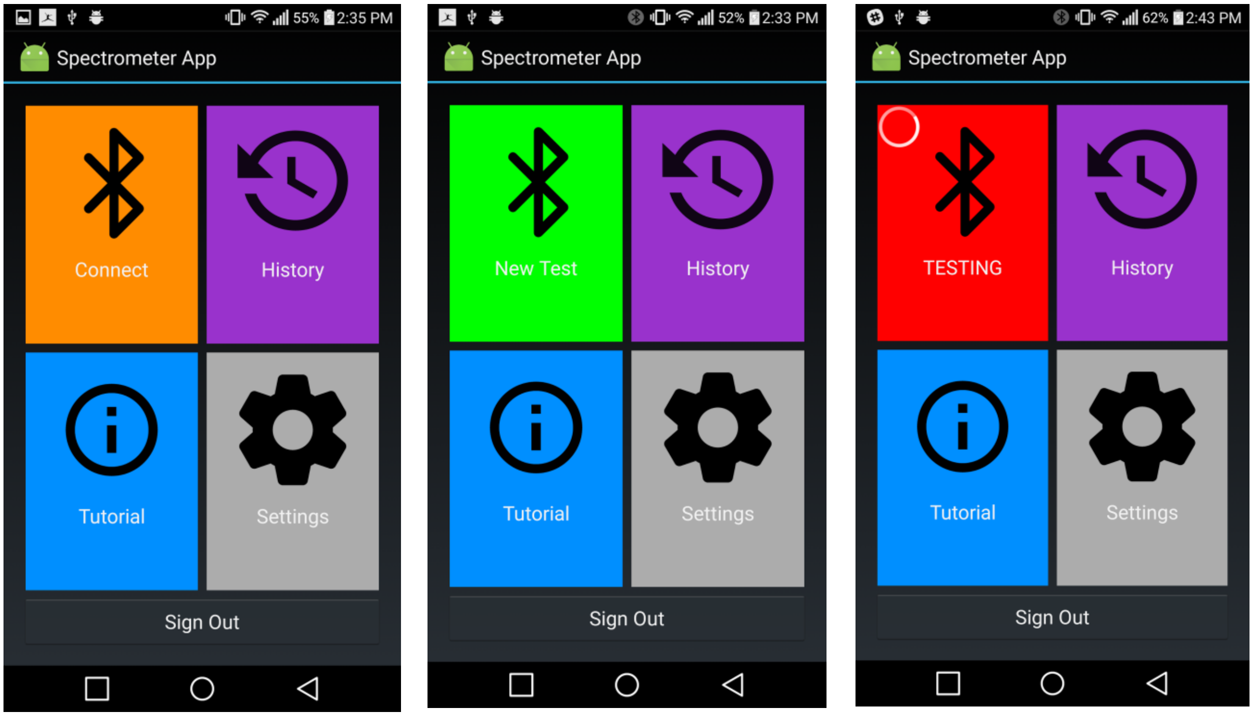
Section 2: App

1. Open the app on your android phone
2. Sign in when prompted, or register a new account.
   a. The app will not function if you do not sign in or register a new account
3. To start and process scans, Bluetooth functionality should be active. The app will prompt you to do so if it is not.
4. Wait for the yellow LED to become active on the enclosure
5. Press the yellow connect button to establish a Bluetooth connection to the enclosure. The button is shown in the image to the right.

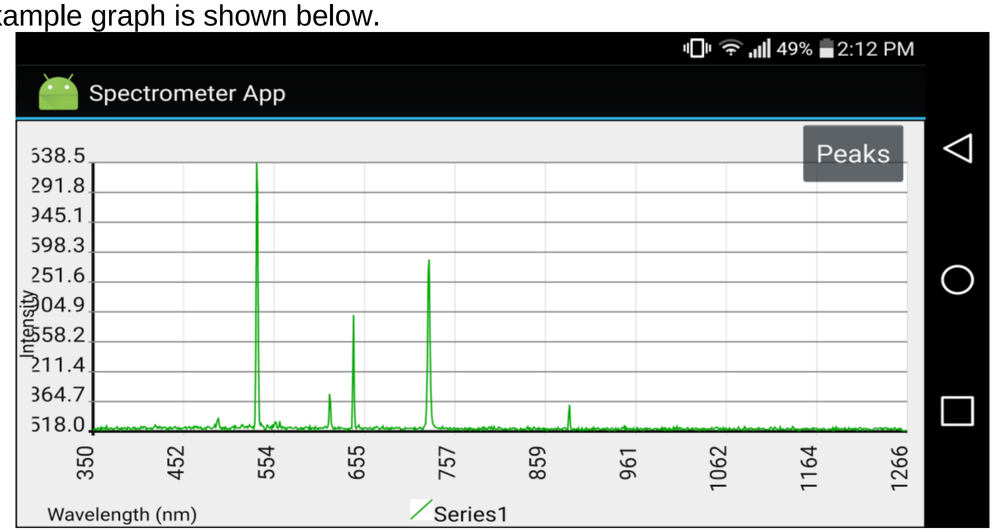
6. Once connected the connect button will turn green, and when clicked it will run a scan and add the data to your user history.



7. When a scan is in progress the connect button will turn red, do not disturb the enclosure during this process. Each state of the button is shown below



8. After the process is complete the connect button will turn green and you can run another scan whenever you would like.
9. After scans are complete you can click the blue history button to view graphs of previous scans, this view of the graph also has a button that will display the peaks of the graph. An example graph is shown below.



10. If you want a more detailed view of the graph you can also use the web viewer to view the graph.
11. Once you have completed running all scans, click the grey settings button to access the settings view. From here click the shutdown button to power down the Raspberry Pi.

Appendix II: Alternate Designs

Light Source with Chemical Treatment

Changes from our current design:

At the very beginning of this project, our client presented us with two options for scanning water samples with our spectrometer. The primary method is what you see in our current design. Alternatively, the option was presented to introduce a chemical mixture to the water sample, and shine a bright light through it for the spectrometer to read.

Issues discovered:

There were several technical issues with this design, with the main issue being that the light source was large and required a large power source, effectively making it impossible to use that light source in a portable fashion. It was expressed to us that the high voltage arc technique was the method to use.

Takeaways:

There are many ways to get a spectrometer reading of our water sample, but this project focuses on the high voltage arc technique.

Arduino vs Raspberry Pi

Changes from our current design:

During the design process of our project's hardware, we knew that we would need a microcontroller of some sort to communicate with the spectrometer and relay that data to our phone app. Several options were considered. Some included the Raspberry Pi you see in our current design, an Arduino, a Teensy, and several other low power microcontrollers.

Issues discovered:

When we receive data from the spectrometer, we receive approximately 3000 pixels at 2 bytes each. These pixels must be processed individually before being sent to our cloud database. To do this quickly, we decided to go with the Raspberry Pi, which has a much more powerful processor. Some other perks of using a Pi include the fact that it runs a full Linux operating system, which allows us to use networking, SSH into it, as well as plug it into a monitor and debug things manually.

Takeaways:

In some applications, using a Raspberry Pi would be overkill, and could possibly be slower if handling simple data. In those cases, an Arduino would be perfect for the task.

Bluetooth vs Wi-fi communication

Changes from our current design:

When looking for a protocol for sending our data from our microcontroller, we examined several popular types. Bluetooth which you see in our final design, and WiFi Direct were our main options.

Issues discovered:

When attempting to design a system using Wifi Direct, we discovered that the protocol is fairly complex to get implemented, and Android did not support it directly. There were ways we could make it work, but it would have taken us much more time.

Takeaways:

Every application has unique needs for its requirements, and in our case Bluetooth was a better fit. Every protocol has pros and cons which must be considered.

Serial vs USB spectrometer communication

Changes from our current design:

When attempting to communicate with the spectrometer, we needed a method to send and receive commands to initiate scans and receive data. Our two options were USB and RS-232 Serial. In our current design, we use RS-232.

Issues discovered:

Implementing USB came with many technical complications, as well as a lack of USB ports on the small version of Raspberry Pi that we chose to use. This was another case of, we could have implemented it and it would have eventually worked fine or even better, but we were able to achieve results more quickly with serial communication.

Appendix III: Other Considerations

Section 1: Issues

1. There is a surprising amount of variety in serial communication standards. Our group spent some time learning about the difference between logic levels and connection pinouts. The main initial obstacle to getting data from the spectrometer was that the RS232 cable and breakout connector followed no published wiring specs. After we could determine the pinouts and obtain a level-shifter, we were able to make quick progress on the rest of it.
2. Our group ended up with hard-copies of almost every part's datasheet that was used for the project. This was simply because we had to annotate the datasheets so heavily that it became impractical to do it electronically. We came across everything from misleading specifications to downright false information during this project. Needless to say, if the datasheets were correct in the first place, implementation wouldn't have taken up so much of our time.
3. Along with the above point, we ended up frying some components due to this. One notable example was a pre-built voltage booster module that we initially used for testing. Page 1 of the datasheet for the FET on the board specified max of 15V and 5A. It started to smoke and blew up at 5V. Examining the datasheet afterwards, page 10 had a short section explaining that the maximum power dissipation was only 0.4W (0.53% of expected).
4. Using personal supplies for the project had the potential to cause some issues. One was a 10-year old tin of flux which had dried out to an almost peanut-butter-like consistency. Once we put it on the Raspberry Pi, it almost immediately stripped the outer layer of the solder mask from the PCB.

Section 2: Project by the Numbers

Parts destroyed: 4
Flux/Lead fumes inhaled: Unhealthy Amounts
Number of times that reversed polarity caused issues: 5
Man-hours spent debugging said polarity issues: 7
Number of times a floating ground connection caused issues: 3
Man-hours spent debugging said floating grounds: 5
Pretended to be part of a sophomore-level EE lab to use their equipment: 1
Fantastic looking scans deleted the day before a major presentation: 2
People tazed by touching voltage booster: 0
Number of times safety equipment was worn around voltage booster: 0
Fingers burnt by solder/soldering irons: 4
Man-hours spent during the last four days of the project: 72
People in TLA who asked what we were doing when testing voltage booster: 2
Number of times we responded with just "Senior Design!" and went back to work: 2