Aqua Eco: A Smart Aquarium

Edward Richards, Lisandro Osorio, Matthew Klein, Evan Kurnia

DEPT. OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, UNIVERSITY OF CENTRAL FLORIDA, ORLANDO, FLORIDA, 32816-2450

***Abstract* — The focus of this paper is to explain the design process and functionality of AquaEco—a fish monitoring system. The AquaEco monitoring system aims to solve the problem that many first-time aquarium owners face—high mortality rates. AquaEco includes sensors to monitor the water and its quality along with providing control to a fish feeder and lighting system. The system is controlled via an android application where the user can see in real-time the changing temperature, pH and clarity of the water. The user is also able to get notifications directly to their device if the parameters exceed healthy values.**

***Index Terms* — Wireless Sensor Networks, Sensor Systems and applications, Event detection, Wireless Communication, Condition Monitoring.**

I. INTRODUCTION

Our main goal was to develop a device that can provide real-time information about a consumer's aquarium. This device utilizes various sensors along with motors and LED lights. Lighting, feeding and monitoring are our core aims for this solution. Our solution titled AquaEco has features that reduce the labor and supervision required of the consumer as much as possible. AquaEco stands for Aquarium Ecosystem and aims to give people a device that is designed for first time aquarium users in mind. AquaEco is a great way for a household to begin their first steps into owning a full aquarium. An aquarium is a great way for people to learn responsibility by owning a pet that is low maintenance yet can still provide a household with an enjoyable and memorable experience if done correctly. Fish can live many years and can be a staple part of a household if taken care of properly. The issue arises when fish are subjected to bad water conditions. First-time aquarium owners might not be able to know when to change the water or know if it is currently safe for fish[1]. This lack of knowledge can often lead to many first-time fish owners' fish dying easily. In turn, these people are far less likely to try and keep fish again. AquaEco helps manage the aquarium by providing beginner tools that will help diagnose water condition and alert the user when the water needs changing or care. We aim to alleviate the challenge of this while introducing the concept of IoT into the household.

AquaEco features automatic lighting that will be synced to an internal clock to allow the tank's lights to turn on and turn off at the same time every day, mimicking the natural cycle of night and day. The user is given the option to change these times or manually control when the lights will turn off or on in the app. It also features an automatic fish feeder, which can be controlled via the android application. Like the lighting, the user can specify what time the feeder should release food each day. Lastly, we will be providing three sensors that will monitor the aquarium: pH, Temperature, and Turbidity. These readings, too, can be monitored via the android application, and the user will be notified when the temperature or pH of the tank is not within range. The application features a virtual tank builder in which the user will select from a given list of fish. After the user creates their tank, the application then monitors for the given temperature and pH ranges that this specific tank needs. These features will allow the end user to control and monitor their smart aquarium from anywhere in the world, and at any time, relieving the need for them to be physically present for feeding and lighting. This can be useful in many situations, such as when a user must be away from their home for a few days, or they cannot be home at the time when they would normally feed their fish. The application will be the bridge between all these components and the user. With our society's increasing dependency on technology to lessen the load of organizing our daily life, we feel that a solution such as our own should be very useful to a wide variety of people. We understand that there are individual products like temperature sensors and pH strips available for users to buy in many pet markets; however, we feel that by introducing the application along with one complete device that we will be able to differentiate ourselves in the pet market.

II. OVERVIEW OF SYSTEM

As shown below our aim was to have a system that meets these needs in terms of communication and interfacing. The power supply provides the necessary current and voltage to power the main components including the sensors, Wi-Fi module and feeder. The WiFi Module supplies the information to the server via data transfer protocols like http. This diagram's main purpose is to give an overview of the hardware components and how they are laid out within AquaEco.

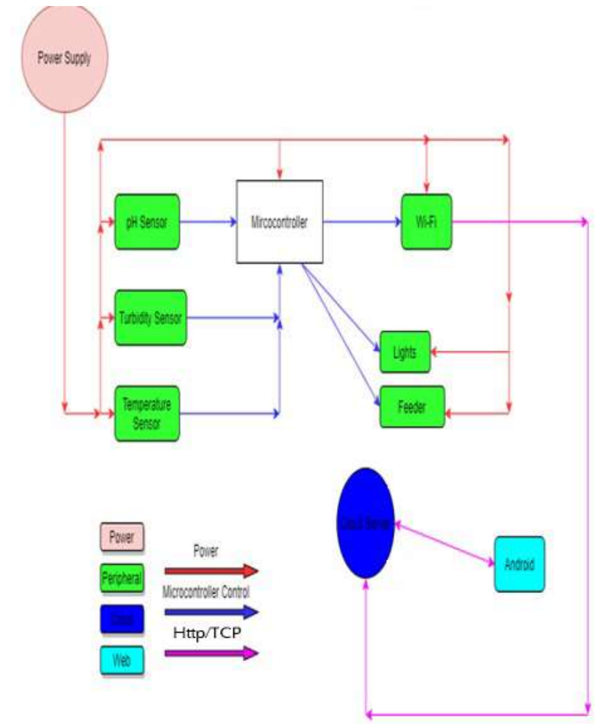


Fig. 1. Hardware components layout and connections.

A. ATMEL ATMega328P

The ATMEL ATMega328P microcontroller is a 8-bit microcontroller which runs based on RISC architecture for fast execution and boasts low power consumption through active operation and six types of sleep modes, which increases the efficiency of the AquaEco especially over a long period of time. It contains multiple methods of device communication through UART, SPI, and I2C as well, and has flash memory, EEPROM, and SRAM for data storage and programmability. One major advantage of the Atmega328P is that it supports ISP through an SPI interface, which allows the microcontroller to be reprogrammed while already installed inside of a system instead of having to be taken out to be programmed before installing it again. This allows for a much easier process for sending data into the microcontroller without needing to dismantle the system. The biggest advantage of the Atmega328p has to be with documentation and support found all over the web. Because this processor is found in many Arduino and other similar development boards, there is open source code available that taps into the many features this processor has. This board has been tested to work with a variety of modules and other devices which means that development time is much shorter than with any other microcontroller. These chips are easily flashed and could run code that is used with arduino products.

B. ESP8266

The ESP8266 module can be found for under four dollars through manufacturers and has become a recent pioneer in IoT applications. Its low cost makes it easy to incorporate it into many applications. It features 802.11 b/g/n giving it a slight edge in connectivity to other modules and has access to TCP and UDP modes while running on the TTL serial interface. It also has the same security features as other leading modules, and, most importantly, can be reprogrammed using the Arduino IDE. We ultimately chose this module to use as it had the features we wanted while also being extremely low in cost.

C. Gravity: Analog pH Sensor

The Gravity: Analog pH Sensor operates via an analog signal with a terminating BNC connector. One of the biggest reasons we chose this sensor was because of its PH meter analog to digital converter board. This board takes in the analog reading and properly converts it to a range of 0-5 volts. The pH sensor works in measuring the hydrogen ion concentration index also known as pH which is found finding the voltage difference between solutions and the base that was measured. Calibration is an important step in making sure that the proper readings are being displayed. We can do this to the sensor by Using the library supplied to properly map the voltages into a proper pH reading. In this case the supplied DfRobot_ph.h library for the arduino ide has a built-in calibration process that helps calibrate the sensor and store the values onto the Atmega EEPROM. This sensor is wired up to the same 5 V Vcc supplied by the Atmega.

D. Turbidity Sensor

Turbidity is the measure of opaqueness found in a liquid. The measurement is usually taken in terms of NTU. The NTU increases if there is an increase in suspended particles in the liquid. We chose this measurement as feces, food and algae can all cause the water to have more suspended particles. Higher turbidity can cause increased stress levels and cause fish to have issue finding food [2]. We can use this reading to display to the user whether it is time to change the fish water. The turbidity sensor supplied by DF robot works by measuring the light transmittance and scattering rate within a medium. This sensor like many of the other sensors in the system uses a 5-volt power source. This sensor releases values from 0 to 4.5 volts and works by taking this voltage and converting to a complementary NTU value which can go from 0 to 3000. These voltages and NTU levels correspond via the equation

$$(1)\quad NTU = -11204x^2 + 5742.3x - 4352.9$$

which is derived from taking the voltage of 4.2 volts as the baseline 0 NTU and having the minimum 2.5 volts correspond to 3000. To properly read the values from the device we will use an A to D converter also supplied by DFrobot. The turbidity sensor, however, is simply not accurate enough to display readings of NTU. Instead we calibrated the sensor to produce results in three steps. Clear, murky, and dark. We mapped the values that it produced from NTU to these words and displayed this to the user. This way this sensor is still used for its original intended purpose.

E. DS18B20 Digital Temperature Sensor

Our final temperature sensor selection was based on a variety of reasons, there were many benefits to having an external temperature sensor that is based on the one wire bus. This bus meant that connection to the Atmega was straight forward. The sensor interfaces via the digital input on the board, the other two wires red and black are for Vcc and for gnd. One important thing to be aware of is that we used a pull up resistor of 4.7k ohms so that we guaranteed a high state for this temperature sensor.

This temperature sensor can operate on 5V which we can supply via the 5v output found on the board. With these components the sensor operates through via the library DallasTemperature which is a third-party library available to download via the internet. This library was explained earlier but is the main reason we have rapid development times. The readings are done in Celsius and through conversion can be turned into Fahrenheit readings. These results were compared with a laser temperature reading and were found to be accurate. This temperature sensor is able to work in water—which is the main goal ultimately, as it will stay inside the fish tank for monitoring.

F. WS2812B

One of the main reasons we wanted to include lighting in our system were the many benefits it provided a wide variety of fish and aquatic life like plants.[3] In this regard we choose the best option for our system that was also able to be controlled by our micro controller and set the proper time controls. LED lighting provided a low power, low heat option that would not negatively affect our fish. The WB2812B is an addressable led strip. This means that using an external library known as FastLed we can individually control the LED's using their position. The led strip operates using 5v which matches the rest of our system. Each LED draws approximately 50 mA. As these LED's are very bright and draw a lot of current, we decided to limit these to only four in our final system.

G. Servo 9G Motor

The motor that will be in charge of powering our fish feeder is the servo 9g motor. This motor is small and compact which will be used for our small fish tank feeding one fish. This part can be changed to a larger servo or even a stepper motor if we decide to make a much larger fish feeder as the servo fits our need in the small area sector. To connect the motor, we used the powered digital inputs the Atmega has and connected it to the 5V power supply. Using the libraries, we were able to use the servo to operate a sweeping motion that pulls back and then forwards on a wooden block to dispense the food.

IV. HARDWARE DESIGN

There were many considerations for the design of the PCB based on the needs and selection of components of the system. A PCB design and schematic software called EAGLE was used to build the schematic of the PCB, decide the layout of the components, route the connections, and export files to be sent to PCB fabrication companies for production. Although extensive research was conducted when designing the first prototype of the PCB, it was found that some components still had to be added or replaced. A second version of the circuit board was created with new additions and was tested to work for the final implementation.

Multiple devices are included in the AquaEco, and all of them are connected to the microcontroller, the ATmega328. Due to the multiple peripheral devices that had to be connected to the microcontroller, through holes were selected as the connection points of the PCB because they provided sturdy, stable connections that were resistant to physical, electrical, and thermal stress. This was especially necessary during the prototyping stage of this project, in which wiring had to be adjusted multiple times and the stability of the system was not yet known. These holes are also located around the perimeter of the circuit board and allowed for an organized location in which the devices were wired and soldered to. Keeping the through holes around the outside of the PCB instead of having mounting areas for the sensors directly on the board also allowed for flexible placement of components, which was also a useful property when finalizing the positioning and case of the AquaEco.

For the connections of the system, the schematics, datasheets, and specifications given by the manufacturers of the components were referenced to determine where the through holes should have traces to. For the pH, temperature, and turbidity sensors of the AquaEco, each sensor required a connection to a 5 volt Vcc, ground, and a data pin. The Vcc and ground wire for all sensors are connected to the Vcc and ground pin respectively of the microcontroller, the data of the temperature sensor is connected to a digital pin, and the turbidity and pH sensors are connected to analog pins.

The LED strip; while also sharing a three-wire connection of Vcc, ground, and data line; had to be connected differently than the sensors. LED strips are a component that can consume a large amount of power, especially when a particularly large length of LED strip is used. If the LED strip was connected to the ATmega328 directly similar to the sensors, it would suddenly draw a significant amount of current and drop voltage levels within the system when turned on, which could end up damaging the LED strip, microcontroller, or entire circuit. As a result, the Vcc and ground pins of the LED strip are wired directly to the power supply and have a 1000 μF capacitor connected across them. By having it connected directly to the power supply, instead of drawing a large amount of current through the ATmega (which could damage it) it would safely receive it from the source. The addition of the capacitor also acts as a buffer for the current and would smoothen out sudden changes in voltage. Furthermore, a 470 Ω resistor is connected in series between the data pin and a digital pin of the microcontroller for the same reason; to limit the current going into the strip.

The power supply of the system was also another major consideration. During the initial design phase of the PCB, it was not yet known whether the instrumentation kit that was provided by UCF for this project would be sufficient to power the entire circuit. To prepare for the situation in which an alternative power supply would have to be implemented, two through holes for a Vcc and ground were added to the PCB with traces throughout the board to provide the 5 V that most of the devices of the AquaEco needed to operate. If a different power source was to be utilized, then switching out the current supply to a new one would only require the wires from the old source to be removed and replaced with those of the new one. During testing of the initial PCB of the circuit, it was found that the instrumentation kit did not provide enough current to power the system reliably at only 700 mA. Problems such as the WiFi module occasionally turning on and off and the LED strip not fully turning on the number of LEDs that the microcontroller was programmed to toggle were seen. Furthermore, the voltage of the system at various locations were monitored with a voltmeter and found to be much less than the desired level. Due to the connections of the PCBs power supply being through holes however, the PCB was able to be disconnected and switched to an alternative supply easily; a 5 V, 2 A wall wart that was split open and connected via the internal Vcc and ground wires. This provided the system with a sufficient current and is included in the final implementation of the project.

The device of the AquaEco that didn't run on 5 V was the WiFi module, the ESP8266. Instead, the module was powered by 3.3V. To supply this, a voltage regulator was added to the PCB, which would convert a 5 V into a 3.3 V output. The first version of the voltage regulator was designed using an online tool provided by Texas Instruments called WEBENCH Power Designer. However, when the ESP8266 WiFi module chip and its 4 pins were connected onto the circuit as well as most of the other peripheral devices, the voltage of the output of the regulator was tested to stabilize around 2.4 V instead, much less than what was needed to power the main 3v3 header on the ESP-8266. As such we decided to add another Voltage regulator to power only the 3v3 pin out, while using this one to power the IO pins. A second voltage regulator implementation was then chosen with the LD1117V33, a low drop voltage regulator that would take an input of between 4.3 and 15 V and reduce the output down to a fixed
3.3 V. Along with featuring a through hole connection that made implementation of the regulator simple, the overall design of the second voltage regulator setup was simple and

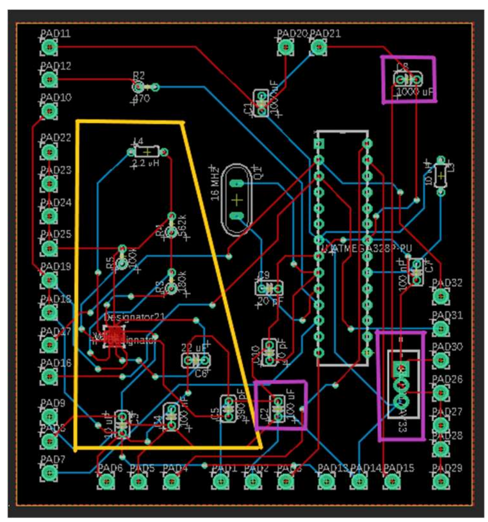


Fig. 2. PCB view of the first voltage regulator (yellow) and second voltage regulator (purple)

provided a compact and cheaper solution that was easier to troubleshoot. The only additions needed to the LD1117 were capacitors connecting the input and output pins of the regulator to ground, which would stabilize the signal from the LD1117.

When testing the regulator on the second design of the PCB, only the LD1117 setup had to be implemented onto the board; although the mounting for the first voltage regulator was left on the board if it were to be needed to power the IO pins of the ESP-8266, the second voltage regulator was able to provide the necessary current and voltage to power all four pins of the ESP-8266. During testing, it was found that the output voltage of the LD1117 regulator did provide the necessary 3.3 V, even when all devices of the PCB were connected. Alongside having through holes connections to provide the 3.3V, EN, RST, and IO0 pins of the ESP8266 with 3.3 V, through holes were also added for its ground, TX, and RX pins connecting to the microcontroller.

For the main micro controller wiring we needed to have the necessary clock signal to not only drive the many components but also interface properly with the arduino uno bootloader that we were planning to flash. Because the arduino uno bootloader required a 16Mhz external crystal. We decided to also include this crystal in the second design of the circuit board. The crystal allows for a more accurate timing of program communications, which is highly beneficial to our system. Also, the external oscillator is much faster than the 8 MHz internal clock of the ATmega, which allows for more commands to be executed faster and further improves the communication between the user's application and the AquaEco. Two 20 pF decoupling capacitors are also connected on both ends of the oscillator to ground to remove unwanted noise with the clock signal. A low-pass filter is also included in the right side of the PCB connecting the Vcc of the ATmega328 to the AVcc pin, which reduces the noise between the two connections.

IV. SPECIFICATIONS

Our system specifications are shown below in Table 1, the numbers with asterisks (*) are specifications that can be demoed to potential clients and show the speed and accuracy of our system. Gathering the requirements of a system is always a very important part of the project. The requirements can be of many kinds and can take various forms, below are the main marketing components of our system that must be met to meet the demands of the market and of the user, this end goal is necessary to establish the need of features.

TABLE 1
HARDWARE SPECIFICATIONS:

Specification type	Description
Hardware 1	The system's *dimensions* should not exceed 16''12'' 6''
Hardware 2	Sensors should provide accurate readings within 1:30 of environmental change
Hardware 3	The fish feeder should have an operating speed of at least .12 seconds with a stall torque of at least 17.5oz /in.
Hardware 4	The fish feeder will handle depositing food at least 1 time per 30 minutes.
Hardware 5	The microcontroller should use the Wireless LAN 802.11b/g standard to communicate to a server
Hardware 6	Ph sensor, turbidity, and temperature sensors should be accurate to within 7 percent.

TABLE 2

SOFTWARE SPECIFICATIONS:

Specification type	Description
Software 1	Alerts should be received when thresholds set through the app are reached within 1 minute.
Software 2	Application should not have loading screens exceeding 2 seconds
Software 3	Set up time should not exceed 2 minutes
Software 4	Easy to use GUI that will display information to users

V. USER MANUAL

This entire system will be easy to set up and will create a strong first-time experience while also providing those already used to creating aquariums capabilities to enhance and lessen the work needed every day.

Step 1. Attach Aqua Eco device to aquarium:

The Aqua Eco device was designed to easily attach to any aquarium. Using the two hooks the device can simply be attached by sliding the system down the side of the aquarium wall. The user should also note that the sensors should be substantially submerged under water

Step 2. Pair device:

First users will need to download the android application Aqua Eco. On the applications homepage users will see a set up new device option. They will select this option which will bring the user to an easy to follow wifi setup menu. After completing all tasks, the Aqua Eco system will be connected to the application.

Step 3. Configure device:

The Aqua Eco system features the exciting ability to input data on the type of environment the system should expect. For example, users will be asked what type of fish and how many will be living in the aquarium. Once selected the system will determine the best living conditions for the fish and alert the user whenever the aquarium's environment becomes dangerous for the fish.

Step 4. Observe function

The user should observe the different sensors outputs and confirm the correctness of those values by testing with an external source. Once the values of the Aqua Eco system are confirmed it is time to introduce fish to the environment. The application will alert the user whenever the different characteristics fall below or above the safe value.

VI. SOFTWARE DESIGN

The next sections will cover the interaction in terms of software in relation with the Atmega-328p, ESP-8266 WiFi module, the server, and the Android application.

A. Atmega-328p

The first aspect of our system begins with the Atmega-328p microcontroller. It is responsible for reading temperature, turbidity, and pH values. It also controls the servo motor for the fish feeder and the led's found in the led strip. The Atmega-328p was first flashed with the Arduino Uno bootloader, this gave us more options when it comes to putting firmware onto the microcontroller in more convenient ways. The Arduino IDE is a great option that includes built in serial monitoring for debugging purposes and has a wide variety of libraries that can be used. The components in the system have extensive documentation relating to systems in place that are using the Arduino Bootloader and as such made the decision to include the bootloader much easier. Using a variety of already pre-established libraries like DallasTemperature, OneWire, EEPROM, FastLED, and Servo, we were able to directly interact and communicate to the Atmega-328p's analog and digital pins. The values that were read from the sensors were communicated to the Esp-8266 via UART, through the settings shown below:

• Baud Rate: 9600 • Data: 8 bit • Parity: None • Stop Bits: 1 bit • Flow Control: none

This communication was done by connecting the Atmega-328p with the Esp-8266 through the Tx and Rx pins of both micro controllers as shown in figure V.

B. Esp-8266

The ESP-8266's main purpose in our system is to relay the information to a cloud hosted server. The information includes the sensor readings. times, and flags for enabling timer controls for the fish feeder and led. The Arduino Ide also features flashing onto these chips directly and as such made the programming of these chips much easier. Another goal of the communication link was to enable the end user to easily set up an AquaEco device onto their home network. As this is not natively supported, we used an external library known as WiFiManager to create a temporary network and have the user select the AquaEco device and establish this connection on their Mobile phone or computer. The Esp-8266 then uses a combination of MySql libraries and Http get requests to obtain information from the server and send information to the server. Additionally, it is in charge of parsing the data along to the Atmega microcontroller to avoid filling of unnecessary information on our serial bus.

C. Cloud Based Server

We used Heroku to host both our Database and Api in the cloud. This enabled us to use our application anywhere without having to be connected to our local network. Since this was our original goal, it was vital to set up a constant connection through a web server from a reputable host. In terms of database, we choose MySql to be the management system. The main reason for choosing a database was to facilitate the communication between the android application and the Esp-8266. By creating this link, we could avoid having to use Bluetooth or local connections for communication. We began by creating a link between the Esp-8266's MAC Address and the cellular device MAC Address. Once linked, information is stored by pairing it back to the Esp-8266 public IP address. Once the information is stored, we retrieved it via an API created using the PHP language. This allowed us to use Get and Post requests in both our android application and from the Esp-8266. Another reason for the need of this API was due to speed, as direct connections to our database were much more costly and due to issues with the ESP libraries also required more tuning.

D. Android application

The final piece in our communications link was the android application. Its main goal was to provide the interface needed to give the user the tools to interact with the AquaEco device. There were three goals set up to be accomplished: the android application has to read the sensors, control the fish feeder and led lights, and provide notifications to the end user.

The android application was developed to be easy to follow and easy to navigate. This was done by using Android Studio, which is development software that eases the development of an android application. It provided us with features such as emulation, that could show our app being used in a variety of devices. It also provided us with a real time showcase of Ui elements via its xml viewer. This allowed us to use the basics of user interface to make a clean and easily readable application.

To interface with the other components the Android Application used okhttp3 external library to allow us to use Get and Post calls to our server. These were done asynchronously as to not impact the user experience. The calls on return would then parse the data correctly and show the user the pH and temperature. After careful consideration, we decided to hide the actual NTU value from our turbidity sensor to the user. This was due in part to the accuracy of our sensor. Instead we decided to use the Turbidity sensor as a means to communicate with the user about the water status in their tank. By setting ranges we are able to display to the user if the water is "Clear", "Murky" or "Dark" and provide a color changing droplet of water to signify this. This allows us to still use the sensor and forego its accuracy issues.

To allow communication with the fish feeder and led lights we designed a user interface that provided the user with two options. Immediate controls and time-based controls. One of the core aspects of our system is to make it easier for the end user to keep a stable Aquariums system. In this regard we felt it was necessary to give timing controls to the user so that every day the same setting would execute. The fish feeder executes at a certain time every day while the led lights are able to be set on a time range. If the user wants more options or more direct controls the immediate options are also given.

One big feature of our application are the curated notifications. In figure 3 we show this component. First the user begins by selecting the change aquarium button. The

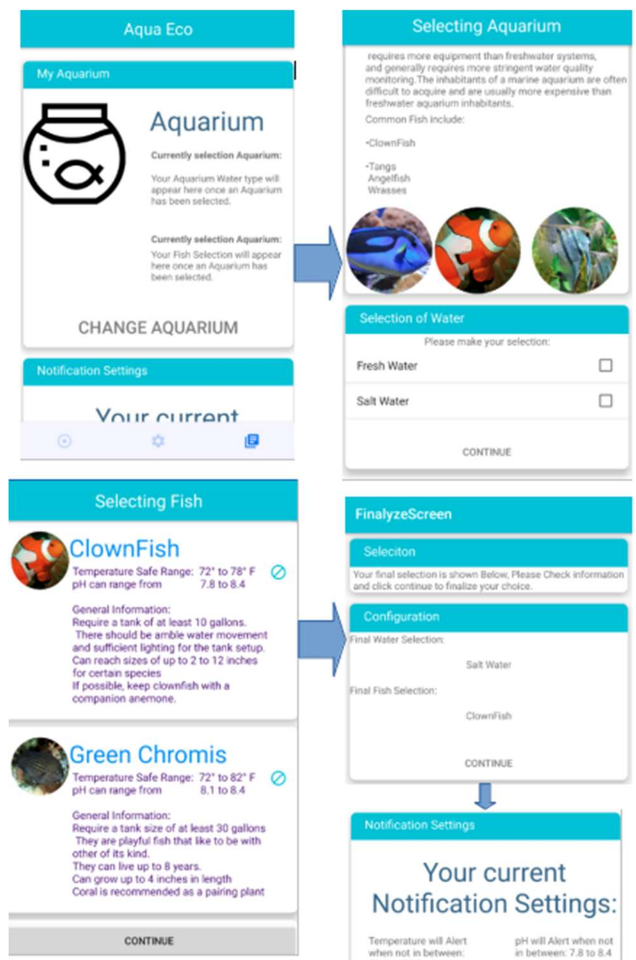


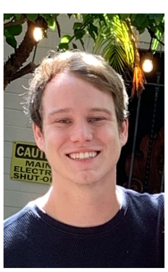
Fig. 3. Overview of Aquarium App creation

user is then prompted with information regarding both freshwater and saltwater options. After a user selects an option, they are presented with a list of seven fish that have been pre-selected to be compatible with each other. After the user scrolls and selects the fish they want included the application calculates the compatible temperature and pH living conditions of the selected fish. These temperatures and pH conditions are saved on the device and are used for background processing of notifications. Every minute the application requests data to our API using androids services and checks to make sure that the temperatures and pH levels are within the proper ranges. If not the user receives a notification as to alert them of this.

VII. Conclusion

The Aqua Eco: A Smart Aquarium gives the user the ability to experience all the joys of owning an aquarium while avoiding the headache that might arise by any solutions already on the market. By creating a wireless sensor network the Aqua Eco provides the user with the ability to detect any potentially dangerous environmental changes long before they become life threatening to the user's aquatic life. The system also allows for environmental controls such as LEDS which can help aid in creating the best environment for aquatic life. Lastly the system includes a fish feeder which can be scheduled so the average pet owner does not have to worry if they forget to feed their fish. With all these features combined the Aqua Eco: A Smart Aquarium revitalizes the excitement of aquatic pet ownership and brings the hobby into the 21st century.

VIII. The Engineers



Edward Richards - Edward Richards is a student at the university of central florida studying for his bachelors in electrical engineering. He was born in Davie, Florida and moved to Orlando after finishing High School. After graduation, Edward wishes to return to school to pursue his Masters Degree in Electrical Engineering.

Matthew Klein – Matthew Klein is a student at UCF studying for his Bachelors degree in Computer Engineering. He was born in Fort Walton Beach, Florida and moved to Orlando to go to UCF after High School. Post-graduation, Matthew will be working for a government contractor as a Software Engineer Intern, and hopes to eventually pursue a Masters at UCF.

Evan Kurnia - Evan Kurnia is a student at UCF studying for his bachelors in Electrical Engineering. He was born in Hollywood, Florida, and has had an interest in electronics and engineering since a young age. When he graduates, he hopes to find a job and gain industry experience before considering obtaining his Master's degree.

Lisandro Osorio- Lisandro Osorio is a student at UCF studying for his bachelors in Computer Engineering. He was born in Orlando,Fl and began studying at UCF after highschool. He wishes to find a software engineering job after graduation to obtain industry experience.

Acknowledgement

The authors wish to acknowledge the assistance and support of the University of Central Florida Electrical Engineering and Computer Science departments. Along with special acknowledgements to Dr. Samuel Richie and Dr. Lei Wei.

References

[1] Angela, T. (2013, March 01). Spotlight Goldfish: Proper Aquarium Environment. https://www.saintfrancis.org/2013/03/01/spotlight-goldfish-proper-aquarium-environmen

[2] Hecht, T., & Lingen, C. (2017). Turbidity-induced changes in feeding strategies of fish in estuaries. https://www.ajol.info/index.php/az/article/view/154200

[3] Shirlie, S. (2019, February 13). LED lighting for your Freshwater Aquarium. https://www.thesprucepets.com/led-lighting-1381056