University of Arkansas – CSCE Department

Capstone I – Final Proposal – Fall 2021

Aquaponics Monitoring

William Mendoza, Calder West, Hunter Yarbrough, Payton Smith

Abstract

Have you ever wanted to effortlessly maintain the life of your plants and fish on the go? Today, there exists no all-in-one aquaponics monitoring system that provides convenient, accurate information on the go. The purpose of our project is to find a solution to this problem by developing a mobile android application that detects a multitude of different levels within aquaponic systems and alerts users when any of these measurements are outside of their desired range. This approach requires the use of an aquaponic tank in addition to the appropriate sensors: a light sensor, a hygrometer (to determine if the plants are receiving enough water), a pH sensor (to analyze water acidity), a thermometer (to read water temperature), and a water level sensor (to make sure the water levels in the tank are appropriate). The sensors are installed in the aquaponic tank based on their intended purpose, whereby a wire will connect each sensor to individual ports on an Arduino WiFi Rev2. Over WiFi, the Arduino will transmit all measurements collected from the aquaponic system as data to Google Firebase, the host of our server and database. After all the data collected from the Arduino has been converted into JSON objects and stored into Google Firebase, Firebase will transfer this information to the application. The application will receive these attributes and display them in a way that is easily understandable for users. In addition, users are immediately notified of any issues within the aquaponic system so that they may tend to the system's needs as soon as possible. This application will seamlessly integrate into the busy day-to-day lives of consumers, as it has one function: to monitor the information it is collecting and notify the user when nutrient levels in the aquaponic system have exceeded or fallen below their limitations. Now plant parents, new and experienced alike, can live worry-free knowing that they have an around-the-clock technology-powered system in place that will alert them when things go wrong.

1.0 Problem

The practice of cultivating plants is an environmentally friendly hobby to pursue. Not only can it be therapeutic to the hobbyist, but it also encourages them to spend more time outdoors. There is even the added potential of personal or financial benefit by growing your own organic food or selling it at a farmers' market.

There exists no convenient way of monitoring an aquaponics system from a remote location. Leaving plants and fish unmonitored can become problematic if the aquaponic system does not contain the proper nutrients it requires to maintain a healthy living environment. Without proper monitoring equipment, the fish and plants could perish or grow at a slower rate. However, maintaining a garden, or an aquaponics system, could be very difficult for beginners. Beyond simply water and sunlight levels, more delicate plants and fish require strict levels for pH and water temperature. There is currently no simple, all-in-one system in place that can monitor these levels and notify owners when specified boundaries are outside of their range. Because time is such an issue people can become overwhelmed and turn away from aquaponics.

2.0 Objective

The objective of this project is to develop a monitoring system that makes maintaining an aquaponic system simple for the average home gardener. This system should do the time-consuming work of measuring the water levels, plant moisture, pH levels, water temperature, and light levels in an aquaponic system and automate it, sending the data back to the user and notifying them if anything seems abnormal in their system. The data will be readily available on the user's smartphone so they may review it when necessary and have complete transparency over their aquaponic system wherever they are. As a result, aquaponics will become much more approachable to those that are picking up the hobby for the first time or simply did not have the time commitment to learn and practice traditional aquaponics monitoring.

3.0 Background

3.1 Key Concepts

There exists no easy, all-in-one solution for an aquaponics monitoring system that tracks factors like water levels, plant moisture, pH levels, water temperature, and light. There has been a substantial increase in the number of households that have and grow small crops/herbs. Tracking the health of plants can be cumbersome and non-transparent. Plants can take time to maintain, and they require consistent nurturing so that they can persevere and survive. Without the necessary technology to monitor the plants and fish, they could be malnourished or affected by water levels, plant moisture, pH levels, water temperature, and light. A mini aquaponic tank will be used to maintain a healthy living environment for the plants and the fish. The aquaponic ecosystem creates sleek, low-maintenance, self-watering planters for growing different types of plants. The aquaponics tank will come with a self-cleaning mechanism controlled by a pump allowing air to flow to the fish and food to float to the plants. The tank is substantially big enough to fit a small number of plants that could grow conveniently in a household.

An Arduino board will act as our monitor that maintains the health of the plant by tracking factors like water levels, plant moisture, pH levels, water temperature, and light. This is done using a variety of insertable probes. The Arduino would send information to a built HTTP server that parses the incoming requests and extracts the values. The server program is called Google Firebase, which is used as an intermediate communication medium for IoT devices, such as the Arduino. Google Firebase allows information to be stored in JSON, so JSON will be the server language of choice since it is easy to control and manipulate data. This will allow the application to send information back to the user so that the user has a clear understanding of the quality of their plant-based on visual factors such as water levels, plant moisture, pH levels, water temperature, and light. The user will be able to sign in or sign up on a mobile application that will receive all the data pulled from the Arduino software and stored in Google Firebase. This information will consist of water levels, plant moisture, pH levels, water temperature, and light. The mobile application will send notifications in case the plant moisture is not wet enough, the water levels are low, the pH levels are out of a specified boundary, the plant is not getting enough light, and the water temperature is safe for the fish. The user will also be able to manage multiple aquaponics systems within one account, these systems will be stored and linked to the user's account within Google Firebase.

Probes are essential for monitoring a healthy environment for the fish and the plants. The most important factors include water levels, plant moisture, pH levels, water temperature, and light. The Arduino will support these necessary probes and the probes will be programmed to work correctly with the board using Arduino software. The water level probe will ensure that the water level of the tank is never too low for the fish or the plants. If the water level becomes too low, there will be an alert sent to the user’s mobile application. The plant moisture will ensure that the plant is getting enough water. If the plant is not getting enough water the user will not only be alerted but will be told there may be a problem with the pump. Regarding pH levels, this is one of the most important factors since certain fish will require different pH levels to survive. The pH level monitor will alert the user if it goes out of the boundaries the user specified through the mobile application that is safe for the fish. Water temperature and light will also be necessary to sustain healthy fish and plant life, therefore these probes will be programmed so that the user is alerted if these factors are out of specified boundaries. The user will set a period that the plant should be getting light; if the light sensor does not recognize light during these specified boundaries, then the user will be alerted that the plant needs more light.

3.2 Related Work

[1] Simple Arduino Controlled Aquaponic System has similar components to our aquaponics monitoring system and can monitor nutrient levels; however, it does not include a mobile

application that sends data from the Arduino to the user. This project instead focuses on self-sustainability rather than user sustainability. A simple Arduino Controlled Aquaponic System takes time to reach an equilibrium. Because the system must build up a culture of organisms to reach stability, it means this project requires 4 full weeks before it is ready to function properly. Before this time, the system claims it can lose fish and plants. The system must be frequently checked during the early stages of the system's life. Aquarium-bought chemical indicators are required to monitor the concentration of nutrients in the system. The focus of this project is on maintaining the health of the ecosystem rather than making the experience more accessible to users. Additionally, there is no way to alert the user when any of these measurements are out of bounds.

[2] The Smart Aquaponics with Dashboard is an aquaponics system with two tubs. The tub in the bottom has water and fish in it. The tub at the top of the system simply contains the plants that the owner wants to grow in the system. Plants can be in pots or planted directly into the tub at the top of this system. The excess water flows out of the tub or pot and enters the fish tank. The water in the tank will already be oxygen-rich, so there is no need for an external aerator for this fish tank. This system also cleans the water in the fish tank automatically while running so the user is not required to regularly clean the fish tank. There are a lot of similarities between our hardware and this project's hardware. We will be using similar sensors as the Smart Aquaponics with Dashboard including the plant moisture sensor, temperature sensor, and Arduino Uno. This project mainly differs in the aspect that it requires to be wired into the internet rather than being a wireless interface. The data is also sent to a server that provides a dashboard of information on a web page rather than a mobile application. There also is not any way to notify the user in case any of the sensors or probes malfunction or go out of a specified boundary.

[3] The AquaSprouts Garden product is similar to what we want in size but not quite as intuitive due to it being self-cleaning. We want to create an app that allows the user to optimize all the factors that go into nurturing a plant. We also want to view our aquaponics data in real-time rather than trusting a self-sustaining machine. The AquaSprouts Garden is special because it brings the principles of aquaponic farming into a simple living space for the user. The water circulates from the aquarium through the plant bed, and fish waste is recycled to provide vital nutrients for growing plants, leaving the water clean and clear without the need for additional filters or frequent water changes. This is a well-designed project that is already being marketed on Amazon. In terms of the hardware and physical layout of how the plants and fish are combined into one system, our project does not drastically differ. There is no simple way of monitoring this system. The system simply monitors itself, which can limit the type of plants and fish that you may want to maintain in an aquaponics system. Our system will have a pump that works the same way but will also include a list of probes so that the user can monitor all the

different factors such as water levels, plant moisture, pH levels, water temperature, and light through a mobile application.

4.0 Design

4.1 Requirements and/or Use Cases and/or Design Goals

Requirement: Provide an aquaponics monitoring system that uses an Arduino wired to a pH level sensor, water level sensor, thermometer, light sensor, plant moisture sensor that collects data every ten minutes.
Use Case: Retrieves data from sensors or probes to send off to the server.

Requirement: The Arduino must be able to connect and stay connected to WiFi.
Use Case: Send collected data wirelessly to a Google Firebase database and update the user. If there is no data received from the Arduino, then the server should alert the user that there may be a problem with the sensors, internet, or software.

Requirement: The Arduino should send an alert if a probe seems to be malfunctioning.
Use Case: Sends the issue to the server so that the app can alert the user, allowing them to address the issue as soon as possible.

Requirement: The android application monitors the data it receives from each of the sensors to determine whether there is an issue with the aquaponic system it is monitoring.
Use Case: Allows the user to upkeep their system without having to physically check on it regularly.

Requirement: The Arduino needs to ping the server to ensure it is connected to the internet.
Use Case: The Arduino needs to stay connected to the internet so that it can send information from the Arduino software.
Design: The Arduino will pull data from the probes or sensors, this data will be sent to Google Firebase, then the Android application will pull the data from the Google Firebase and alert the user if any of the values are missing or out of the specified range.

Requirement: The mobile application should refresh the data periodically as well as whenever the user requests a refresh.
Use Case: Ensure the app is kept up to date so it can alert the user if an undesired value is collected.

Requirement: The mobile application will allow the user to set specified boundaries regarding pH levels, temperature, and light at a specified time during the day.

Use Case: Ensure that the application alerts the user if any of the specified boundaries are out of range.

Requirement: The Arduino must stay connected to a power source.
Use Case: The Arduino must maintain power to monitor the aquaponics system, if power is lost and no data is received in Google Firebase the user will be pinged with a notification.

Requirement: The aquaponics tank needs to include a pump and a tub on top of the tank.
Use Case: The pump will need to filter the water, the tank will host the water, and the tub will host the plant or plants.

Requirement: The app will support user signup using a username, password, and ID, and login with username and password.
Use Case: This tethers the user's system's info to one account, allowing the user to access their monitoring data from any supported device.

Requirement: The app will support adding, removing, and switching between unique Arduino systems.
Use Case: Allows the user to have multiple Arduino monitoring systems tied to the same account.

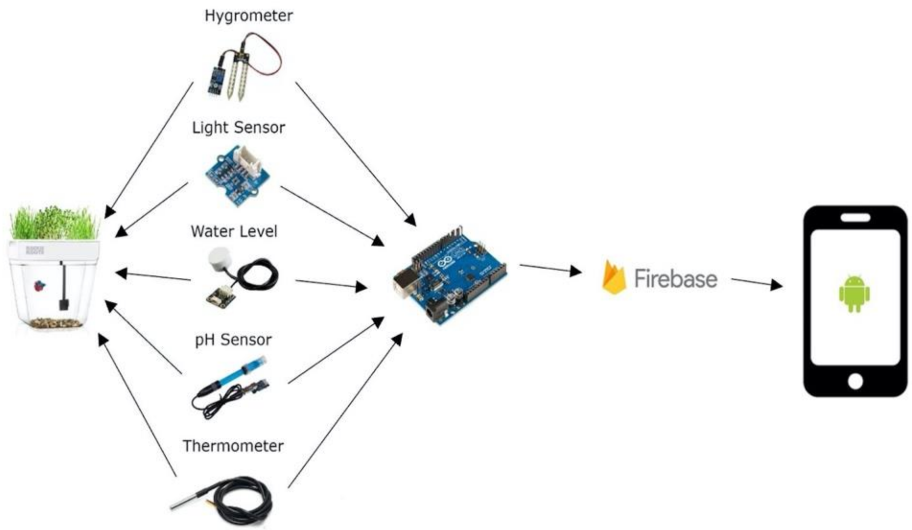
Requirement: The system should support multiple distinct clients out of one server, and it should correctly map the data from a system to its corresponding user account.
Use Case: Allows multiple people to receive/use info from our monitoring app.

4.2 High Level Architecture

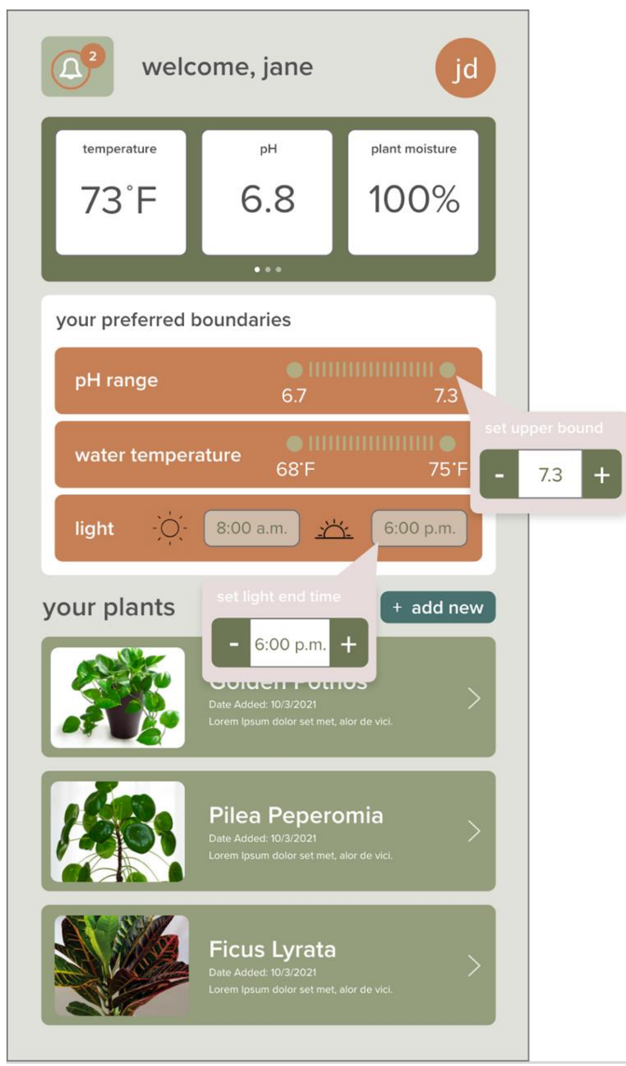
Our project monitors small-scale aquaponics systems. This system will comprise of two components, one is a hardware component that consists of an Arduino and its connected sensors, and the other is the software aspect, which is an android app that receives data from the Arduino through a Google Firebase server/database.

The hardware component will consist of a light sensor to measure the intensity of light, a hygrometer to measure the amount of water the plant is receiving, a pH sensor to measure the acidity of the water, a thermometer to measure temperature levels, and a water level sensor, and an Arduino UNO Wi-Fi Rev2. The hygrometer, pH sensor, thermometer, and water level sensor will be placed in the water, and the light sensor will be connected to the top of the system. Each of the sensors will then be connected to an Arduino UNO Wi-Fi Rev2, which will connect to Wi-Fi and send the collected data to a Google Firebase server using HTTP. The Arduino will take a reading on each sensor and send the collected data to the server every ten minutes. This number is arbitrary and can be changed, but since the system should not fluctuate rapidly because we are using an inexpensive server, there is no need to collect and store data continuously. The Arduino will be given an ID number specified by us, which will allow the user to connect their account to their Arduino system. The Arduino will be hardcoded to connect to Wi-Fi, but functionality to use the app to connect to the Arduino using Bluetooth and set up the Wi-Fi could be added if time allows.

The server and app comprise the second component of this project. The server will be hosted using Google Firebase and will receive data from the Arduino, store the data as JSON objects, and send it to the app. The app is an android app. The frontend of the app will be written in XML and the backend will be written in Java. The app will open with a login page that asks for a username and password and has a button to a signup page. The signup page will have fields for a username, password, and Arduino ID. This will connect the username and password to the specific Arduino system that the user wants to use. Once the user is connected to an Arduino system, the user will be brought to a home page. The home page will display the data received from the Arduino system and will allow the user to modify the range boundaries for pH level, water temperature, time receiving light, and water level height. The app will use the ranges to compare against the received sensor data and send notifications to the user when the received data is above or below the range. Since the user can configure the ranges for each of the sensors, the system will be able to be used for a variety of plants and fish. The app will have navigation to another page that displays each of the five data points recorded from the past month in a graph. The app will have a profile page that allows the user to log out and add, remove, or switch Arduino systems. Switching Arduino systems will allow the user to view the data of a different Arduino system. The data that is displayed on the app will be updated every ten minutes. There will be a refresh button on the home page that updates the current data by sending a message to the Arduino telling it to take a reading and upload the data to the server. The JSON objects stored on the server will come in two types. The first will have fields for username, password, and an array of Arduino IDs that are connected to that account. The second will contain a field for each of the five data types collected. The server will also be able to tell when it is no longer receiving data from the Arduino and will send a message to the app that the Arduino has lost connection. A diagram of how each part is connected is included below along with a screenshot of the layout of the app.



Hygrometer
Light Sensor
Water Level
pH Sensor
Thermometer
Firebase



welcome, jane
jd
temperature
73˚F
pH
6.8
plant moisture
100%
your preferred boundaries
pH range
6.7
7.3
set upper bound
7.3
water temperature
68˚F
75˚F
light
8:00 a.m.
6:00 p.m.
set light end time
6:00 p.m.
your plants
+ add new
Date Added: 10/3/2021
Lorem ipsum dolor set met, alor de vici.
Pilea Peperomia
Date Added: 10/3/2021
Lorem ipsum dolor set met, alor de vici.
Ficus Lyrata
Date Added: 10/3/2021
Lorem ipsum dolor set met, alor de vici.

4.3 Risks

Risk	Risk Reduction
WiFi connection to the Arduino is disrupted	Implement code within the Arduino WiFi Rev2 to restart the WiFi connection after it is lost. If it is still lost after the restart, send the user a notification that the Arduino has lost connection to the internet.
Electricity and water interacting (fire hazard)	Create drip loops for any wire coming out of the water. In this case, that would be the hygrometer, pH sensor, and water level sensor. This will prevent water from reaching any electricity connectivity points.
Plant pathogens/disease	Regularly check, clean, and sanitize the aquaponic system. Any water contaminants will directly affect fish and all the plants.

4.4 Tasks

1. Initial research on the background of aquaponics and Arduinos.
    a. How plants and fish can live together in unison benefitting from one another in an environment
    b. How significant factors like water levels, plant moisture, pH levels, water temperature, and light play a significant role
    c. Make sure we have a clear understanding of how to wire the Arduino and connect it to the sensors in and on our aquaponic tank

2. Practice using basic Arduino commands, and program the Arduino and sensors so they function properly when connected. These sensors will include water levels, plant moisture, pH levels, water temperature, and light.

3. Test each sensor with simple test cases (before the sensors have been submerged in water or placed in the system) to ensure that none of our hardware is faulty and we have the essential components required for the aquaponics system. Understand how sensors return valid values to our Arduino program run inside of Arduino software.

4. Begin coding Arduino software in Python.
    a. Conduct research on Python libraries that work with the Arduino board, software and sensors
    b. Implement simple Python concepts that return values based on data

5. Test to see that code has been implemented properly by testing the Arduino-sensor system with the necessary sensors submerged in water.
    a. If coded improperly, revise code to ensure the sensor information is received and being read within the Arduino software
    b. Clean up any unnecessary code and make sure that it is easy to read

6. Start initial research and development of the server.
    a. Gain an understanding of how Google Firebase receives, and processes information sent by the Arduino
    b. Also, research how to use JSON commands on Google Firebase correctly so that the user will have permission to send data to and retrieve data from the application.
    c. Begin building an HTTP server that parses incoming requests and extracts values

7. Store JSON information so it is functional with our application.
    a. Store users' sign-in information as well as their collected data in a JSON file. Their sign-in information will be needed to give the user permission to view his or her data upon sign-in validation.

8. Gain knowledge regarding android mobile application development in Java. The mobile application will include information sent to the user from the database, including water levels, plant moisture, pH levels, water temperature, and light. This includes research regarding:
    a. Sending notifications to the user
    b. How to pull information from Google Firebase to our application
    c. How Java receives and manipulates data sent from Google Firebase

9. With the knowledge we have so far, code the Java application so that the data is being received and stored appropriately.
    a. Data that the application receives should be able to be compared to other values set within the application. For example, the pH levels must stay within a default specified range, so the app needs to be able to identify when a measurement is out of range.

10. Test and ensure code works properly by putting sensors in scenarios where their measurements should return a value that is out of range.
    a. If the sensors do not return out of range, continue revising the code until values return as expected.

11. Create an alert for each sensor when it measures a value out of range. These notifications will need to be worked in parallel to ensure that our code remains consistent across the application and that all alerts are finished at the same time. This includes notifications for when:
    a. Plant roots are not receiving enough water
    b. pH levels have gone outside of their set boundaries
    c. Plants are not receiving light during the time the user has set for them to
    d. The water level in the tank has fallen below the minimum line
    e. The temperature in the aquaponic tank is not safe for the fish

12. Code the application so that the user is alerted on their mobile device whether the app is open or not.

13. Test the different factors being measured in the system in different conditions to ensure that the user is receiving notifications when measurements are out of bounds.

14. Begin researching XML.
    a. Gain an understanding of how to implement a user interface with XML in Android applications
    b. The user interface of the Android application will need to be simple to use while maintaining enough information for the user to gain insight into any problems within their aquaponics system

15. Start implementing the user interface of the application.
    a. The user interface will have visual factors added such as color and pictures that symbolize what the aquaponics system is monitoring. Multiple people will be working on the user interface in unison to ensure the user interface is appealing and simplistic to use
    b. Give users the ability to view the water level, plant root moisture, pH levels, water temperature, and light at any time

16. Begin developing code to monitor data over time.
    a. The historical data will need to be stored in JSON and worked on in unison while the historical data visualization for the user interface is being created.

17. Give users the ability to set their own boundaries within the app and implement this in the user interface. This includes the ability to set boundaries regarding:
    a. Water temperature
    b. pH levels
    c. Time of daylight needed to reach the plant

d. Water level height

18. Test the consistency of the aquaponics system and the mobile application in unison
    a. This will ensure that both our hardware and software work as expected.
    b. Make sure our application, server, and Arduino are running and functional at least 2 weeks in advance.

19. Modify and update the user interface
    a. Since the UI is not required to analyze the data of our factors, we will perfect it during this time. This will ensure that everything works as expected and there are no problems with the Arduino, software, or code.

4.5 Schedule

Tasks	Dates
1. Initial research on background in aquaponics *(1a-1b)* Begin researching in depth on Arduinos *(1c)*	1/06-1/13 (1 week)
2. Practice with basic Arduino commands 3. Test sensors with simple test cases	1/14-1/21 (1 week)
4. Research Arduino Python libraries *(4a)* Begin implementing initial Python code for Arduino software *(4b)*	1/24-2/03 (1.5 weeks)
5. Run multiple test cases with sensors submerged in water Revise any non-working or unnecessary code *(5a-5b)* 6. Initial research on Google Firebase *(6a-6b)* Begin developing HTTP server *(6c)*	2/04-2/14 (1.5 weeks)
7. Make a file to store users' sign-in information and collected data *(7a)* 8. Research Android mobile application development using Java *(8a-8c)*	2/15-2/22 (1 week)
9. Code the basic Java application so that data is being received and stored properly *(9a)* 10. Test sensors out of range to ensure code works properly *(10a)*	2/23-3/05 (1.5 weeks)
11. Create an alert for each sensor regarding when it measures a value out of range *(11a-11e)*	3/06-3/16 (1.5 weeks)
12. Code the application so that the user is alerted on their mobile device whether the app is open or not	3/17-3/27 (1.5 weeks)

13. Test the different factors being measured in the system in different conditions to ensure that the user is receiving notifications when measurements are out of bounds	
14. Begin researching how to implement a user interface with XML *(14a-14b)*	3/28-4/04 (1 week)
15. Start creating the user interface for the application *(15a-15b)* 16. Develop code to monitor data over time *(16a)* 17. Give users the ability to set their own boundaries in the app *(17a-17c)*	4/05-4/19 (2 weeks)
18. Modify and update user interface *(18a)*	4/20-5/04 (2 weeks)

4.6 Deliverables

- Design Document: The hardware will consist of an Arduino Uno Wi-Fi Rev2 that works on the Arduino software, probes or sensors that measure water levels, plant moisture, pH levels, water temperature, and light that work with the Arduino Uno Wi-Fi Rev2, an Aquaponics tank that includes a pump, a server hosted on Google Firebase, and a mobile application developed for Android using Java.
- Application code: The Arduino software will be developed in Python. The code will send the information received by the Arduino board to Google Firebase. Code for the Android application will be written in Java and made to retrieve JSON data from Google Firebase.
- Aquaponics tank: Plants will be placed on top of the tank, and a fish will be in the reservoir below the plants. The aquaponics tank will include a simple pump that pulls water from the tank upward to the plants and pushes oxygen created by the plants down to the fish. Sensors placed in (pH sensor, water level sensor, thermometer) and on (light sensor, hygrometer) the tank will be connected to an Arduino Wi-Fi Rev2.
- Google Firebase(server/database): Google Firebase is the server that receives and stores data as JSON objects sent from the Arduino. The server will send information from the Arduino to the mobile application. Google Firebase will also host user information and allow for multiple aquaponic systems to be linked to one account.

5.0 Key Personnel

William Mendoza – Mendoza is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. He has completed Software Engineering, Programming Paradigms, and is currently enrolled in Embedded Systems. He has had some experience in the past working with an Arduino board. Mendoza will ensure that the physical aquaponics/plant system interfaces with the Arduino correctly to connect it to the application.

Robert Yarbrough – Yarbrough is a Senior Computer Science major in the Computer Science Department at the University of Arkansas. He has completed Programming Paradigms, Software

Engineering, Database Management, Computer Organization, and Programming Foundations I and II. Yarbrough has worked on personal Python projects that deal with Machine Learning and Database Management. Yarbrough will ensure that the application will work properly with the Arduino device and the User Interface is implemented correctly so that the hardware and software work in unison.

Payton Smith – Smith is a senior Computer Science major in the Computer Science and Computer Engineering Department at the University of Arkansas. She has completed Programming Foundations I and II, Software Engineering, Programming Paradigms, and Database Management. Smith has experience interning as a User Experience Designer, focusing specifically on the creation of app wireframes and ensuring that workflows are efficient from beginning to end. Smith will make sure that the user experience in the app is seamless and meets the goals set at the beginning of the project.

Calder West - West is a senior Computer Science major at the University of Arkansas. He has completed Software Engineering, Database Management, Computer Organization, Programming in C, Programming Paradigms. He has experience working on websites and android apps. West will ensure that the app and website are implemented correctly and can connect to the aquaponics system.

Dr. Matthew Patitz - Dr. Matthew Patitz received his B.S. and M.S. in Computer Science from Iowa State University, then worked at a start-up software company, SupportSoft, in Redwood City, CA as a software engineer and team lead from 2000-2005. In 2005 he returned to Iowa State University to pursue his Ph.D. in Computer Science, which he completed in 2010. From 2010 to 2012 he was an assistant professor of Computer Science at the University of Texas-Pan American. Since 2012, Matt has been in the Department of Computer Science and Computer Engineering at the University of Arkansas. He is currently an associate professor. His research interests are DNA computing, algorithmic self-assembly, and theoretical computer science in general. He has received research grants from the National Science Foundation, including a CAREER award in 2016, and published over 60 peer-reviewed conference and journal papers.

6.0 Facilities and Equipment

-An Arduino UNO WiFi Rev 2 will be required so that data is collected from each of our probes, and with the built-in WiFi capability, it will be able to send data to our server.

-A prebuilt hydroponic system will help us ensure our sensors work correctly and that the data makes it from our monitoring system to the app.

-A power supply that powers the Arduino UNO WiFi Rev 2 and includes a long enough cable to reach the tank.

-A plant light that provides light so that the plant can grow properly under the required conditions.

-A fish will help feed the plant's nutrients and help with testing the sensors in our aquaponics system.

-A plant will be required for the aquaponics system and will act as a test case for the plant moisture sensor.

Sensors (compatible with Arduino)

-The temperature sensor will be required to make sure that the water in the tank is warm or cold enough for the fish.

-The pH sensor will be used to ensure that the water's pH levels are safe enough for the fish.

-The water level sensor will ensure that the tank has enough water and there aren't any issues regarding leaks.

-The light sensor will recognize if there is sufficient light for the plant.

-The plant moisture sensor will recognize if the plant is getting enough water or not.

7.0 References

[1]https://www.instructables.com/Simple-Arduino-Controlled-Aquaponic-System/

[2] https://create.Arduino.cc/projecthub/neuron/smart-aquaponics-with-dashboard-4e8b23

[3]https://www.amazon.com/AquaSprouts-Garden/dp/B01B4ZRVR4/ref=pd_lpo_2?pd_rd_i=B01B4ZRVR4&psc=1

[4]Leithauser, David C. *How to Cheaply Monitor and Automate Your Aquaponic-Hydroponic Garden with Arduino/Genuino*. 2017.