FISH GROWTH AND WATER QUALITY MONITORING SYSTEM FOR AN INTENSIVE AQUACULTURE SETUP USING RASPBERRY PI VIA LORAWAN IOT PROTOCOL

A Project Study Presented to the Faculty of

Electronics Engineering Department

College of Engineering

Technological University of the Philippines

In Partial Fulfilment of the Course Requirements for the Degree of

Bachelor of Science in Electronics Engineering

Submitted by:

De Pedro, Celline P.

Icamina, Jatt D.

Navarro, John Benjamin E.

Salvacion, Luigi James D.

Sobrevilla, Gian Carlo D.

Villanueva, Apolo A.

Adviser:

Lean Karlo S. Tolentino, ECE, AAE, M.Sc.

August 2020

APPROVAL SHEET

This project study entitled "**FISH GROWTH AND WATER QUALITY MONITORING SYSTEM FOR AN INTENSIVE AQUACULTURE SETUP USING RASPBERRY PI VIA LORAWAN IOT PROTOCOL**", has been prepared and submitted by the following proponents:

De Pedro, Celline P.	Salvacion, Luigi James D.
Icamina, Jatt D.	Sobrevilla, Gian Carlo D.
Navarro, John Benjamin E.	Villanueva, Apolo A.

In partial fulfillment of the requirements for the degree of **Bachelor of Science in Electronics Engineering** is hereby recommended for approval.

E-signature of LEAN KARLO S. TOLENTINO signed July 25, 2020 @ 2:07 PM

ENGR. LEAN KARLO S. TOLENTINO
Project Adviser

ENGR. TIMOTHY M. AMADO
Panel Member

ENGR. MARIA VICTORIA C. PADILLA
Panel Member

ENGR. GILFRED ALLEN M. MADRIGAL
Panel Member

Accepted and approved in partial fulfillment of the course requirements for the degree of **Bachelor of Science in Electronics Engineering.**

ENGR. EDMON O. FERNANDEZ
Head, ECE Department

ENGR. BENEDICTO N. FORTALEZA
Dean, College of Engineering

ACKNOWLEDGEMENT

The proponents would like to extend their boundless gratitude and appreciation to all the people who helped them during the completion of the project study.

To their families, for providing endless love and support. Their prayers and motivating words acted as the strength of the group, lifting their determination and confidence. To Icamina Family, for providing not only a home but emotional and financial support in all possible ways, particularly during difficult times throughout the project.

To their closest friends, Lorin C. Barnido Jr., and Engr. Daniel D. Cabrales, for providing them a helping hand and for the continuing support and encouragements.

To their adviser, Engr. Lean Karlo S. Tolentino, for encouraging valuable ideas and suggestions, and for extending consultation hours to complete this study and make it a success. To Engr. Nilo M. Arago, for his support and guidance not only to the proponents but to all the research groups involved in the project.

To the Electronics Engineering (ECE) Faculty members, particularly to the panel members, Engr. Timothy M. Amado, Engr. Gilfred Allen M. Madrigal and Engr. Maria Victoria C. Padilla, for their objective and constructive critiques during defense and insightful suggestions that led to the improvement of the project.

The group would like to thank Barangay Pulang Lupa Uno, Las Pinas City staffs, principally Hon. Guadalupe Alfonso Rosales, for allowing the proponents to deploy the project and conduct data gathering within the barangay hall perimeters.

This study would not be possible without the grace of Almighty God. The proponents are most thankful to Him for all the wisdom, knowledge, strength and encouragement He has given them all during the making of this project. All the honor and praise of this study are offered to God above.

ABSTRACT

Aquaculture faces many difficulties and one of its issues is the quality of fishes being lower which have a major effect in aquaculture industry. In this study, there will be two setups for the intensive aquaculture system – controlled and conventional setup. The controlled setup will be integrated with weight prediction and water correction and the latter will be the traditional method of cultivating and measuring fish. The proponents developed a fish growth and water quality monitoring system to help fish farmers in monitoring the growth of the fish without manually weighing it, reducing the effort and time consumed by the farmers and less stressful to the fishes. The automated system optimizes the yield of Nile Tilapia by monitoring and automatically correcting vital water quality parameters. Moreover, the system is equipped with weight prediction system for fish growth monitoring. The system identifies the weight of the fish through image processing and predictive analysis. The website application presents all the output data: (1) weight (2) water parameter levels. LoRaWAN will provide the remote access for the wireless transmission and reception of data.

By avoiding stressors to fishes, such as capturing and handling to manually measure its weight and monitor its growth, the data gathered shows beneficent results to the rate of fish growth. The automated aquaculture system yields a higher growth and survival rate of Nile Tilapia compared with the conventional aquaculture setup. The growth rate of the fishes in terms of weight improves by 47.88%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

THE PROBLEM AND ITS SETTING

This chapter presents the background of the study, the statement of the problem, the objectives, the significance, the scope and the limitations of the study.

1.1 Introduction

Aquaculture, also known as aquafarming, is the farming of aquatic life, including fish, shellfish and aquatic plants. There are two main subsets of aquaculture. Mainly marine and freshwater aquaculture. Marine aquaculture refers to the cultivation of aquatic species that naturally live in the ocean, while freshwater aquaculture refers to the cultivation of species that live in lakes, streams or river. Aquaculture is regarded as the most promising source of protein food in incoming years because it is a more controllable and manageable production system. Nile Tilapia and Milkfish are one of the major fishes now produced especially in the Philippines. In the other countries, groupers, sea bass, rabbitfish, red snappers, carps, and catfishes are grown by some farmers. Crustaceans are also being cultivated in aquaculture like tiger shrimps while white shrimps and mud crabs also have great potential (Aypa, 1995).

The Philippines, a country where fisheries is a very important industry and aquaculture is well established has a total territorial water area of 2.2 million sq km. Aquaculture is an important sector in the Philippine fisheries and the most dynamic since the decline of marine fisheries starting 1976. The country’s archipelagic nature is only one of the reasons for its importance. The other reason is the Filipino's great liking for fish. As the saying goes, "No meal is complete without a fish" (FAO, Philippine Fisheries, n.d.).

Nowadays, aquaculture faces many difficulties and one of its issues is the quality of fishes being lower which have a major effect in the aquaculture industry. Capture, transport, and mishandling through catch and release program are some of the reasons of fish having a lower quality (Wolf, 2009). As a solution to the problem stated, the researchers propose to develop a fish growth and water quality monitoring system with automatic correction.

1.2 Background of the Study

According to the *Florida and Wildlife Conservation Commission*, the traditional measurement of obtaining the fish measurement is being handled, lay it flat and manually get the total length including its weight which takes much labor and time consuming on the farmers. Capture and handling are obvious stressors for captive fish. Evidence that these stimuli are intrinsically stressful is provided by experiments that have documented marked increases in blood cortisol and/or glucose levels in fish deliberate handling according to the *Morphologic Effects of the Stress Response in Fish* (Wolf, 2009). Monitoring the growth of farmed fish is a more convenient way of measuring fish. Digital image analysis technique offers the potential for estimating key dimensions of free-swimming fish, from which the fish mass can be estimated according to the *Estimating Dimensions of FreeSwimming Fish Using 3D Point Distribution Model* (Tillett, McFarlene, & Lines, 2000). It targets an automatic vision-based system for estimating of free-swimming fishes. Estimation of fish size and weight is very essential to fish farming industry which provides key data with which feeding, grading, and harvesting operations controlled. In addition, temperature, dissolved oxygen, and pH level are the most critical water quality factors to be considered and monitored simultaneously according to *Smart Aquaponics in a Temperature-Controlled Greenhouse with Plant Growth Monitoring System Using Raspberry Pi via Android IoT Application* (Amora et al., 2017).

This study shall be composed of a stereo-vision camera setup that can monitor and capture moments of the fishes which will be used in image processing, and six different sensors for detection of additional six different parameters; pH value, water temperature, light intensity, air temperature, oxidation-reduction potential (ORP), and turbidity for the experimental setup. It also provides a conventional setup for culturing Tilapia for comparison between the controlled setup.

1.3 Statement of the Problem

The proponents of the project study sought answers to the following questions:

1. Why do we need to measure the growth of the fish?
2. What are the parameters of water quality needed to be maintained?
3. Why do we need to use image processing instead of the conventional way of measuring fish growth?

1.4 Objective of the Study

1.4.1 General Objective

This study aims to develop a smart aquaculture and fish growth monitoring system using image processing.

1.4.2 Specific Objective

The study aims to meet the following goals needed for the development of a fish growth monitoring system using image processing via website.

1. To develop a fish growth monitoring system through image processing using OpenCV Python for image detection and segmentation.
2. To develop a system that monitors the six different water parameters which automatically corrects the water quality in addition to the fish growth monitoring system.
3. To create a system that automatically sends data of the parameters and the fish growth monitoring system to the cloud.
4. To develop a web application that will consolidate and display information for the end-users using various web development languages.

1.5 Significance of the Study

The study eliminates the traditional way of determining fish growth that lessens the effort of the farmers in determining the growth of the fish and minimizes the possibilities of the fish to have a lower quality. The system will focus on monitoring the fish growth without making any contact or handling on the fish with the addition of monitoring and correcting the quality of water simultaneously in real-time. The implementation of the fish growth monitoring system benefits the fish farmers for the development of urban/backyard/small-scaled aquaculture and by having a cultivated fish with a higher quality. Having a higher quality fish leads to a better impact in the aquaculture industry. This study also helps other researchers conducting studies about fish. Considering the invasive conventional way of measuring fish growth affects the fish and could possibly fatal. The image processing will at least eliminate one of the factors that affects the growth and behavior of the fish. This new method of measuring fish growth also highlights the use of internet by creating a web application for online monitoring of the whole system.

1.6 Scope and Limitation

This study will focus on the development of fish-growth monitoring system through stereo-vision image processing using a website which includes the automatic water correcting system having six water parameters readings, mainly, potential hydrogen value (pH), water temperature, salinity, oxidation-reduction potential (ORP), dissolved oxygen (DO), and turbidity. The aerator will be used to increase the amount of dissolved oxygen.

The study is restricted in culturing one species namely Nile Tilapia in an Intensive Fresh Water Aquaculture system. The weight of the fish is only measured averagely with respect to its length. In addition, the water quality will be monitored only on its parameters; mainly, pH value, water temperature, salinity, oxidation-reduction potential (ORP), dissolved oxygen (DO), and turbidity.

1.7 Definition of Terms

1. Aerator – used to oxidize soluble iron and manganese to insoluble precipitates (Suez Water Technologies, n.d.)

2. Blood Cortisol – hormone in the body released when under stress (WebMD, n.d.)

3. Buffer Device – regulates pH (Thermo Scientific, n.d.)

4. Cloud – type of computing that relies on shared computing resources rather than having local servers or personal devices to handle applications (Webopedia, n.d.)

5. Dissolved Oxygen – refers to the level of free, non-compound oxygen present in water or other liquids. (Fundamental of Environmental Measurements, n.d.)

6. Internet of Things – concept of basically connecting any device with an on and off switch to the Internet (and/or to each other) (Forbes, n.d.)

7. ORP – stands for 'Oxidation-reduction Potential'. Measurement that indicates the degree to which a substance is capable of oxidizing or reducing another substance (AQUA, n.d.)

8. Parameter – an independent_variable used to express the coordinates of a variable point and functions of them (Meriam-Webster, n.d.)

9. pH Level – measurement of acidity of alkalinity of water soluble substances. pH stands for 'potential of Hydrogen' (AQUA, n.d.)

10. Stressors – a stimulus that causes stress (Meriam-Webster, n.d.)

11. Turbidity – is a measurement of the degree to which the water loses its transparency due to the presence of suspended particulates (Lenntech, n.d.)

12. Salinity – saltiness or dissolved inorganic salt content of body of water (Coastalwiki, n.d.)

CHAPTER 2

REVIEW OF RELATED LITERATURE

2.1 Conceptual Literature

2.1.1 Aquaculture

Aquaculture is the farming of fish, mollusks, crustaceans and other aquatic organisms like aquatic plants in different environments like Freshwater, Brackish water and Marine Areas. An Aquaculture system can either be Extensive, Semi-Intensive and Intensive for the hatching, rearing, and harvesting of these aquaculture products. Food and environmental conditions are controlled for most aquaculture to increase the production aquatic organisms. (Food and Agriculture Organization, 2018)

Increasing demands that challenges the aquaculture industry has been a strong driving force for further research and development of farming techniques and methods to increase the level of fish production and produce a globally competitive high-quality aquaculture product.

2.1.1.1 Tilapia

Tilapia is the second most significant fish cultured in the Philippines next to milkfish or Bangus in the Philippines. (Fisheries Policy and Economics Division, 2008) Among other aqua cultured fishes, Tilapia is the number one most consumed freshwater fish in the Philippines. (Food and Nutrition Research Institute, 2008) The Philippines produced 316, 536 metric tons of tilapia in 2013 with a value of 669.8 million USD making the Philippines the 3rd largest tilapia producer around Asia and the 4th largest around the world. Nile Tilapia (Oreochromis niloticus) is the second most important cultured fish in the country. (Guerrero, 2013)

Figure 2.1 Aqua-cultured Tilapia

(Is tilapia Brazil's secret weapon in aquaculture growth strategy, 2016)

2.1.1.2 Temperature

Fish species are temperature-dependent. Fishes like tilapia, catfish and bass are warm water species with a preferred temperature that ranges from 65 to 85°F (18-29°C). The desired temperature for tilapia ranges from 81-85°F (27-29°C) for an optimum growth. When the temperature ranges below 70°F (21°C), Tilapia growth decreases and can stop reproduction. Temperatures below 50°F (10°C) is fatal to tilapia. (Rossana, 2016). Stress and mortality from handling increase below 65°F during the harvesting of Tilapia

2.1.1.3 Dissolved Oxygen

DO or dissolved oxygen is essential for respiration and decomposition. Dissolved Oxygen comes from atmospheric oxygen and photosynthesis from aquatic plants. Reduced DO can be detected if the fish don't feed well or come to the surface to breathe. (Towers, 2015) Tilapia can tolerate up to less than .3mg/L that may be fatal for other farmed fish, but growth was not further improved if additional aeration kept DO concentrations above 2.0 to 2.5 mg/L. Although tilapia can survive acute low DO concentrations for several hours, tilapia ponds should be managed to maintain DO concentrations above 1 mg/L. Metabol-ism, growth and, possibly, disease resistance are depressed when DO falls below this level for prolonged periods. (Popma & Masser, 1999)

2.1.1.4 Salinity

The Blue and Nile tilapias can reproduce in salinities up to 10 to 15 ppt but perform better at salinities below 5 ppt. Fry numbers decline substantially at 10 ppt salinity. Although Tilapia are considered freshwater fish, they can grow in an elevated level of salinity and can tolerate a brackish water environment. (Popma & Masser, 1999)

2.1.1.5 pH Level

The pH range or acidity level of water should be maintained between a pH level of 5 to 10, but a range of 6 to 9 can give Tilapia an optimal growth. Soluble carbonate or any bicarbonate source can be used to balance the level of acidity of the water as these are easy to obtain, extremely soluble and safe to handle. (Towers, 2015)

2.1.1.6 Oxidation-Reduction Potential

ORP or Oxidation-Reduction Potential sensor works by measuring the amount of dissolve oxygen present in the water. With higher ORP level, water has greater ability to destroy foreign contaminants including microbes and other carbon-based contaminants. Various levels of ORP corresponds to different applications. The ideal value for the level of ORP for aquaculture ranges from 150mV to 250 mV. (Environment Health, 2016)

2.1.1.7 Turbidity

Turbidity refers to the overall appearance of water and is considered an important parameter for water quality. Presence of foreign substances such as foams and scums provide clues to the characteristics of water quality. Clear water promotes light penetration for underwater organisms. (Boyd & Tucker, 1998)

2.1.2 Sensors

Sensor is a device whose purpose is to detect events and changes in its environment and send information to another electronic device which produces an output as a recognition in quantity detected. It converts physical parameters into electrical signals. (Agarwal, 2015)

2.1.2.1 Industrial Analog pH Sensor

This is an industrial grade analog pH sensor which are used in a variety of environmental monitoring system such as water monitoring for fishponds or aquaponics. This type of sensor requires quick response and exceptional thermal stability. It can also be submerged under water for relative amount of time before re-calibrating. The sensor's electrode is made of sensitive glass membrane with low impedance making it very accurate.

The pH sensor has an adapter board that can be connected to a micro-controller such as Arduino. The small electrical signal coming from the sensor will be converted by the adapter board into an easy to read analog voltage output which varies linearly from 0 to 14 pH. (MakerLab Electronics, 2015)

Figure 2.2 Industrial Analog pH Sensor

(MakerLab Electronics, 2015)

2.1.2.2 Water Temperature Sensor

DS18B20 is a digital thermometer ideal for underwater applications. It is pre-wired waterproof temperature sensor provides 9-bit to 12-bit Celsius temperature measurements that communicates through a digital wire bus. It is used for thermostatic controls and thermally sensitive systems that can withstand from -55°C to 125°C (-67°F to +257°F). (Maxim Integrated, 2015)

Figure 2.3 DS18B20 Temperature Sensor

(MakerLab Electronics, 2016)

2.1.2.3 ORP Sensor

ORP Analog Meter V1.0 measures the ability of the ability of oxidation and reduction of aqueous solutions that follows a logarithmic curve and requires adjustments in terms of calibration. ORP detection is a reliable method of measuring water quality for water quality testing, marine science, hydroponic gardening and aquaculture. It can measure from -2000mV to 2000mV with a response time less than 20secs and has an accuracy of ±10mV (25°C) at a suitable temperature range of 5°C to 70°C.

Figure 2.4 ORP Analog Meter V1.0

(The Pi Hut, 2018)

2.1.2.4 Analog Turbidity Sensor

Analog Turbidity Sensor Module detects particles present in the water by measuring the amount of light that pass through the water which is directly proportional to the amount of total suspended soils (TTS) in the water to identify the quality of the water. It has a response time less than 500ms with an insulation resistance of 100M (minutes). It provides a digital output of High- and Low-level signal where the threshold value can be adjusted using the potentiometer. (MakerLab Electronics, 2016)

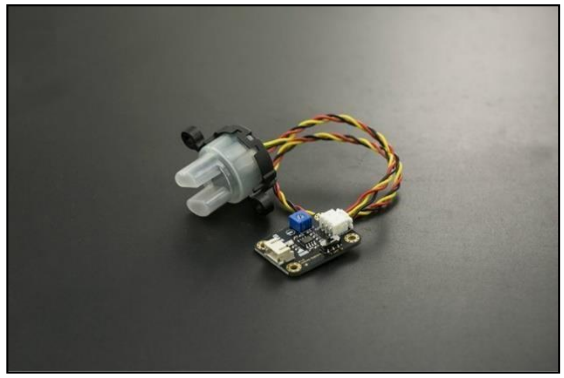


Figure 2.5 Analog Turbidity Sensor Module Gravity Series

(MakerLab Electronics, 2016)

2.1.2.5 Dissolved Oxygen Test Kit

Atlas Scientific Dissolved Oxygen Test Kit has everything you need in terms of accurate full-range DO readings in an environmental monitoring like hydroponics and fish keeping. It has an operating voltage of 3.3V to 5V with full-range DO readings from 0.01 to +35.99 mg/L with an accuracy reading of ±0.2. It is compatible to any microprocessors that supports UART or I2C protocol. (Elec Design Works, 2015)

Figure 2.6 DF Robot Dissolved Oxygen Test Kit

(DF Robot, 2015)

2.1.2.6 Electrical Conductivity Meter

Electrical Conductivity Meter measures the number of electrolytes present in the water. The concentration of the electrolyte varies for different aqueous solution. This sensor is used to measure the salinity of different water systems and is also used for soil. It has an operating voltage of 5V with a measuring range of 1ms/cm to 20ms/cm with an accuracy of ±10% at an operating temperature range of 5°C to 40°C. (DFRobot, 2015)

Figure 2.7 Analog Electrical Conductivity Meter

(DFRobot, 2015)

2.1.3 Raspberry Pi 3 Model B

Raspberry Pi 3B features a 1.2GHz 64-bit quad-core ARMv8 CPU with 802.11n Wireless LAN and Low Energy Bluetooth 4.1 making it suitable for IOT development. It also has 1GB RAM, 4 USB ports, 40GPIO pins, Full HDMI port and Ethernet port excellent for hackers and makers because of its small footprint, low power consumption and low price. (DFRobot, 2018) It also has an upgraded power source that goes up to 2.5A to power powerful devices on USB ports. (MakerLab Electronics, 2016)

Figure 2.8 Raspberry Pi 3 Model B

(MakerLab Electronics,2016)

2.1.4 LoRaWAN

LoRaWAN or Long-Range Wide Area Network is a common protocol for smart assets and IOT devices to communicate with one another. LoRawan is ideal for sensor networks, security systems, smart metering and industrial controls. It has a range of 2km to 5km in an urban environment and up to 15km in a suburban environment and can surely deliver two-way communication at data rates from 0.3kbps to 50kbps (LoRa-Aliiance.org, 2018) LoRa RFM95 can achieve a sensitivity of -148dBm and a power output of up to +20dBm.

Figure 2.9 LoRa RFM95 Shield 915MHz

(Circuit Rocks, 2015)

2.1.5 Arduino Mega 2560

Arduino Mega 2560 is a microcontroller that features 54 digital input/output pins, 16 analog inputs, 4 UARTs (hardware serial port), a USB port, a 16MHz crystal oscillator, an ICSP header, a power jack, and a reset button. (MakerLab Electronics, 2016)

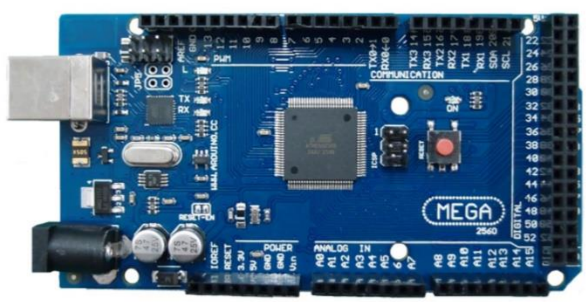


Figure 2.10 Arduino Mega 2560

(MakerLab Electronics, 2016)

2.1.6 Python

Python created by Guido van Rossum is a high-level programming language. Python syntax is designed to be readable and straightforward which makes it an ideal teaching language. It runs on every major operating systems and platforms. Python is mostly used for automation of web browser and GUI's. As a highly effective code generator, it can write different applications that manipulate its own function. (Yegulalp, 2017)

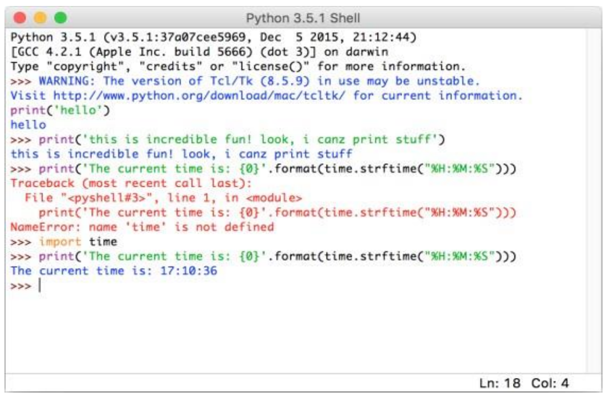


Figure 2.11 Python Programming Language

(Grant, 2016)

2.1.7 Stereo – Vision Camera

A Stereo – Vision Camera is a sort of camera with at least two focal points with a different picture sensor or film outline for every focal point. This permits the camera to simulate human binocular vision, and consequently enables it to catch threedimensional pictures, a procedure known as stereo – vision photography.

Figure 2.12 Stereo – Vision Camera

(Junk Mail Classifieds, 2017)

2.1.8 Solar Photovoltaic System

2.1.8.1 Solar Photovoltaic Panel

Solar cells, also called as photovoltaic cells converts sunlight directly into electricity. It generates electricity by absorbing sunlight as a source of energy. As the wafer from the solar photovoltaic panels gets bombarded by sunlight, photons in the sunlight knocks off excess electrons to make a voltage difference between sides that can produce up 0.5V. (Solar Direct, 2016).

Figure 2.13 Solar Photovoltaic Panel

(Bundu Power Co., 2018)

2.1.8.2 Solar Charge Controller

Excess charging occurs for most Solar Photovoltaic Panels connected to batteries. An overcharged battery may overheat and may become damaged. Charge controllers are usually connected between the solar panel and battery. Charge controller stops electricity to flow as the battery indicated full. (Wade, 2003)

Figure 2.14 Solar Charge Controller

(New Energy Co., 2015)

2.1.9 Buck Converter

Buck Converter also refers to step-down transformer for DC-DC conversions. It steps down the voltage as it steps-up the current from its input supply towards it output or the load. (Mammano, 2001)

Figure 2.15 DC-DC Buck Converter LM2596S

(MakerLab Electronics, 2015)

2.1.10 Inverter

Inverter is an electrical device that converts DC voltage to an AC voltage where the DC input is usually lower while the AC output is equal to the supply of grids equal to 220V. Inverters are usually built as a standalone equipment for solar powers and other renewable energy resources. It also works as a backup power supply from batteries that can be charged separately. (Sunpower UK, 2013)

2.1.11 Relay and Relay Driver Circuits

Relay is an actuator, an electromechanical device that uses electromagnet that to operate a pair of movable contacts in open position to closed position. Relays takes small amount of power to operate the relay coil. Transistors can be used for fast DC switching ON-OFF and requires separate switch circuit to operate. (Keith, 2015)

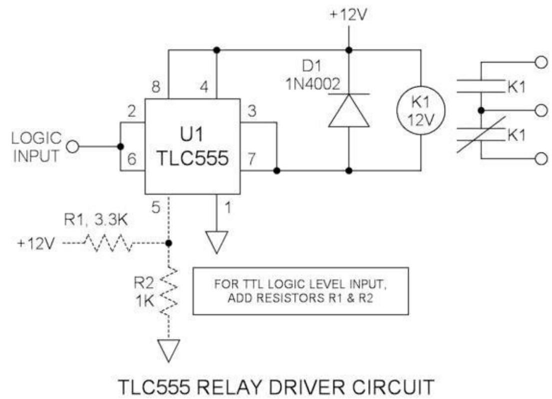


Figure 2.16 Relay Driver Circuit

(Keith, 2015)

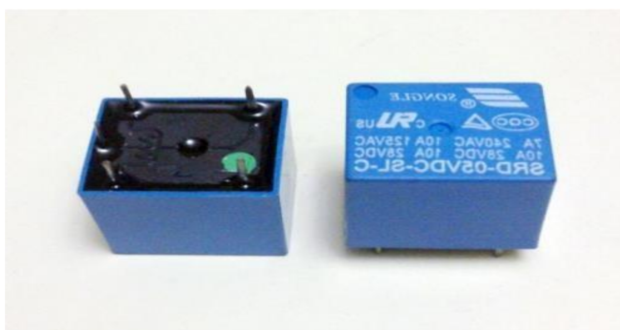


Figure 2.17 Electromechanical Relay

(Keith, 2015)

2.1.12 Internet of Things

IoT or Internet of Things refers to the connection of different physical devices that are now connected to the internet that enables sharing and collection of data. It adds digital intelligence to the device and help merge physical world merge with the digital world. (Ranger, 2018)

2.1.12.1 Cloud

Cloud or Cloud Computing is the storage and accessing of data and programs over the internet instead of personal storage drives. Cloud is considered a metaphor of internet that represents a gigantic server-farm infrastructure. (Griffith, 2015) A cloud can be a wide area network (WAN) like the public internet or a private global network. (PC Mag, 2016)

Figure 2.18 Cloud Network

(PNG Icons, 2017)

2.1.12.2 Web Development

Websites are set of different webpages located, prepared and maintained by the same server as a collection of information or data. (Mifflin, 2016) These webpages include different multimedia contents and can be accessed via a public Internet Protocol (IP) network. Web development refers to developing websites for hosting. It includes web design, web content development and other different tasks. Web developments ranges from creating plain text pages to complex web-based applications. (Techopedia, 2018)

2.2 Related Studies

For the study conducted by Amora, et al., they innovated the conventional aquaponics by developing a plant growth monitoring system that uses image processing through Raspberry Pi to measure the plant surface area. They also include the monitoring of pH level and temperature of the water in the system with real-time acquiring of data detected by the light intensity sensor and air temperature sensor. Normal status is also achievable as correcting devices are present to restore the system parameters to its standard level. In this study, the data reports are being transmitted effectively between the system and an Android unit with the presence of the Android application through IOT. (Amora, Bartolata, Sarucam, Sobrepeña, & Sombol, 2018)

For a successful management of fishponds, understanding of water quality is needed which includes several parameters such as temperature, dissolved oxygen (DO), turbidity, ph level, etc. *Effects of water physico-chemical parameters on tilapia (Oreochromis niloticus) growth in earthen ponds in Teso North Sub-County, Busia County*, a study conducted by Makori, et al., assessed the effects of earthen pond water physicochemical parameters on the growth of Nile Tilapia under semi-intensive aquaculture system. Based on previously high harvest, a control pond was purposively selected, and another five ponds were chosen through the use of Systematic sampling. Using a multimeter probe, physico-chemical parameters of each pond are measured. Sixty fish randomly obtained from the four months' duration of the experiment. These fishes were 10 mm mesh size, measured and weighed. The mean range of physico-chemical parameters are the following: dissolved oxygen 4.86-10.53 mg/l, pH level 6.1-8.3, temperature 24-26 °C, conductivity 35–87 μS/cm and ammonia 0.01–0.3 mg/l. As the temperature and dissolved oxygen (DO) increases, the growth rate of tilapia also increases. The Overall Specific Growth Rate ranges between 1.8% (0.1692 g/day) and 3.8% (1.9 g/day). (Makori, Abuom, Kapiyo, Anyona, & Dida, 2017)

Fish counting and tracking is important for conservation purposes especially for fishing industries. There are existing automatic fish counters that uses resistivity, light beams and sonar principles. The study conducted by Shevchenko (2017) in a research entitled, *Fish Detection for Species Recognition*, he distinguishes fish from other objects in terms of its size and texture by using the BS approach. His system detects moving objects and concluded that the approach can be successfully applied for the fish detection. He used three BS algorithms, Adaptive GMM, KDE and ViBe, to determine what is suitable for the main problem of the study which is the quality of the video. As a conclusion to their study, an average between detection and noise resistance is shown by the KDE, Adaptive GMM is highly motion sensitive while ViBe is very noise-resistant. (Shevchenko, 2017)

Based from the study entitled**,** *Fish Image Segmentation Using Salp Swarm Algorithm* by Ibrahim (2018), image segmentation which is very essential in fish recognition given the circumstances present in its living environment. The

Salp Swarm Algorithm was used for its superiority on other algorithms which belongs on the same category. Salp Swarm Algorithm or SSA is a novel optimization algorithm for solving optimization problems with single and multiple objectives, the main inspiration of SSA is the swarming behavior of salps when navigating and foraging in oceans. Other than SSA they used other segmentation processes and methods like the "Otsu's Method" which is used in thresholding. In simple words, they utilized a segmentation method for real-world fish images based on SSA. (Ibrahim, Ahmed, Hussein, & Hassanien, 2016)

The study conducted by Adam et al., provide a useful data and information in the management as well as conservation of this food fish. Also, gaps in the knowledge regarding this species can be filled through the results accumulated. Oreochromis niloticus (Nile Tilapia) manifests isometric growth pattern whenever their local environment is favorable for their growth. The outcome of this study concurs with the length-weight relationship result obtained by the workers who studied this particular species in the northern part of Jebel Aulia dam Reservoir. (Adam & Khalid, 2016)

Aquaculture is considered as an essential component of achieving food security and economic development. Improvement in the environmental control, decrease in the catastrophic losses and production cost, and enhancement of product quality are attainable in the automation of aquaculture system. Simbye and Yang designed a wireless sensor monitoring and control system for aquaculture. The water quality parameters of temperature, dissolved oxygen content, pH value, and water level can be detected by their system in real-time and will be transmitted to the base station host computer through ZigBee wireless communication standard. (Simbeye & Yang, 2014)

In the study entitled *Design and Implementation of Water Quality Monitoring for Eel Fish Aquaculture* by Salim et al. (2016), the proponents focus on providing a water quality monitoring by using a single Raspberry Pi 3. They developed a monitoring system that can process multiple sensors at once without using a microcontroller as their bridge of data acquisition. Raspberry Pi 3 and Python program is combined for sensor acquisition program and program viewer which comes in two methods, real-time viewer and historical data viewer. Their system can also monitor the changes in the level of dissolved oxygen, temperature, and acidity due to the effect of microbubble aeration. (Salim, Haiyunnisa, & Alam, 2016)

Prototype and proof of concept implementation applied by the concept of IoT about a remote monitoring system for the aquaculture water quality is presented by Encinas, et al. Slow response time in the water quality monitoring are among the problems in an aquaculture system. The proposed system in this work monitors the quality of water based on wireless sensor networks (WSN) and on the Internet of Things (IoT). This system is accurate, versatile, low power consumption and low cost at the same time. (Encinas, Ruiz, Cortez, & Espinoza, 2017)

The topic presented by Chen et al., is an aquaculture based environmental monitoring system of wireless sensor networks for a fish farm Environment Simulation. A mobile device is used to monitor the fish farm Environmental Data accompanied by Instant memory and control over the various environmental data. A WIFI transmission module is equipped in the central system for the most popular Android mobile devices that are connected directly to improve the convenience of the overall system and timeliness. (Chen, Sung, & Lin, 2015)

Water Quality, tank environment, and also fish behavior must be monitored in fish farms. Monitoring the different farming processes can improve the use of resources and its sustainability and profitability at the same time. In the paper, *Design and Development of Low-Cost Sensors for Monitoring the Water Quality and Fish Behavior in Aquaculture during the Feeding Process,* Parra et. al., proposed a set of sensors for the monitoring of both water quality and fish behavior. It can monitor the water quality parameters, fish tank status, and the fish itself. It also has a smart algorithm that sends alarm whenever the parameter levels go beyond the safe value. (Parra, Sendra, Garcia, & Lloret, 2018)

In the work presented by Borges, et al., entitled, *Water Quality Monitoring in Aquaculture as a Remote Laboratory Proxy for Environmental Studies*, through a hardware interface, data from physical and chemical domain of the water are received by an application developed in LabVIEW. These data are recorded in an Microsoft Access Database can be locally queried. On the other hand, an IP camera allows the students to monitor the overall system in real time. The students can observe the real-time variations of a specific water quality parameter and other changes in the levels of other constituent by logging in to the system. This helps the students to familiarize the use of some new information technologies. (Borges, Mendes, Restivo, & Pereira, 2011)

CHAPTER 3

METHODOLOGY

3.1 Research Design

3.1.1 Input – Process – Output Diagram

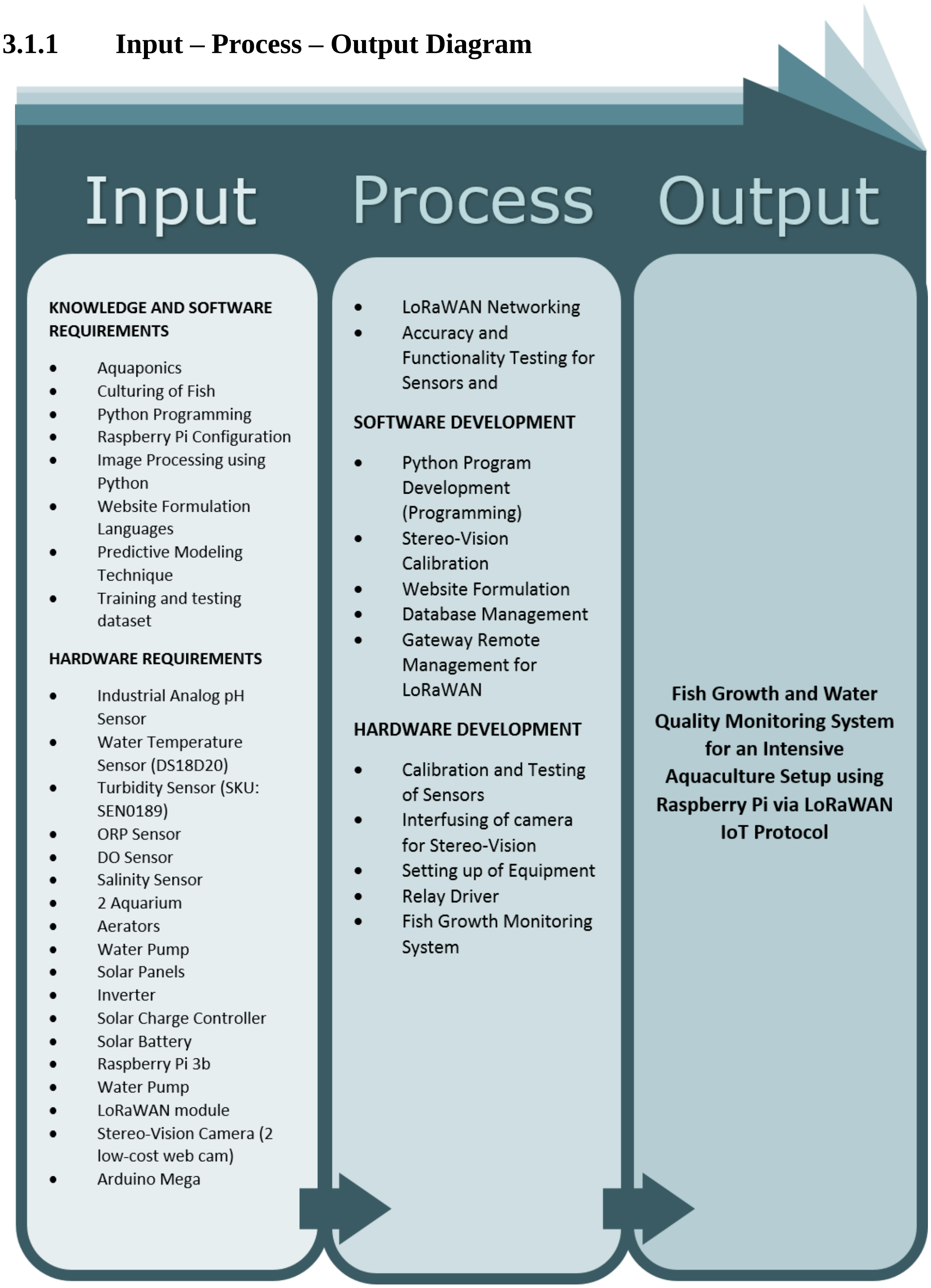


Figure 3.1 Input-Process-Output Diagram

Shown in the figure 3.1 is the conceptual framework of the study wherein the input block includes all the information needed to proceed and the requirements for both hardware and software domains. The software requirements focus mainly on the proficiency in Python programming that involves data and image processing. Hardware requirements, on the other hand, composed of all the peripheral devices that is needed to form the aquaculture. With that said, the input of the project will come from the reference electrode of the sensors which are connected to the Raspberry Pi 3b via Arduino Mega microcontroller. The Raspberry Pi contains the calibration setups for the sensor outputs, as well as the algorithm for image processing and the predictive model to predict fish weight which is based from the data acquired beforehand intended as testing data through conventional method of weighing fishes in an aquaculture. All under the software development section of the process block. For the locus of hardware development, setting up of equipment and calibration of sensors comes first before the habitation of fishes.

The processed data will be sent to the cloud via LoRaWAN remote gateway and will be available for download in the website together with the qualitative real-time water parameter levels that will indicate whether the water is still safe for the fishes.

3.1.2 Research Process

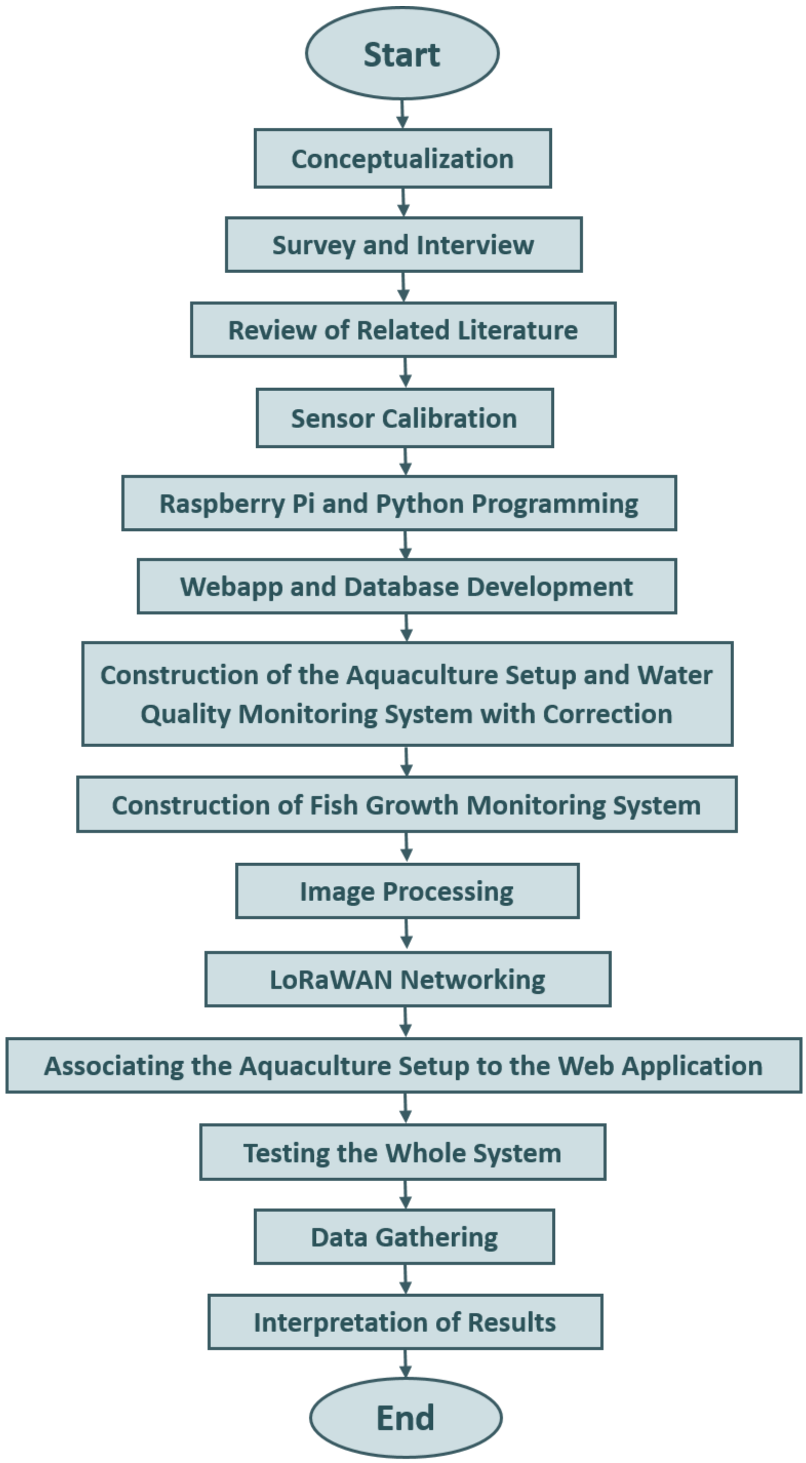


Figure 3.2 Flowchart

Figure 3.2 shows the flow chart on how the research process chronologically occurred throughout the study. The process started from conceptualization on how the aquaculture came up as the main focus of the study, followed by conducting a survey on who are deeply concerned in this area. The researchers then sought for projects that have existed in the past and established what was needed to be done or improved. The sensor calibration and python programming were done before the actual construction of the setup. This effectively gave more time for the machine learning in the later stage of the project. After all the necessary programming was done, the researchers set out to construct the actual aquaculture system. The steps that follows is about assembling the whole setup and connecting everything necessary for deployment. The interpretation of results was done after a full test of the whole aquaculture system and data gathering.

3.1.3 Block Diagram

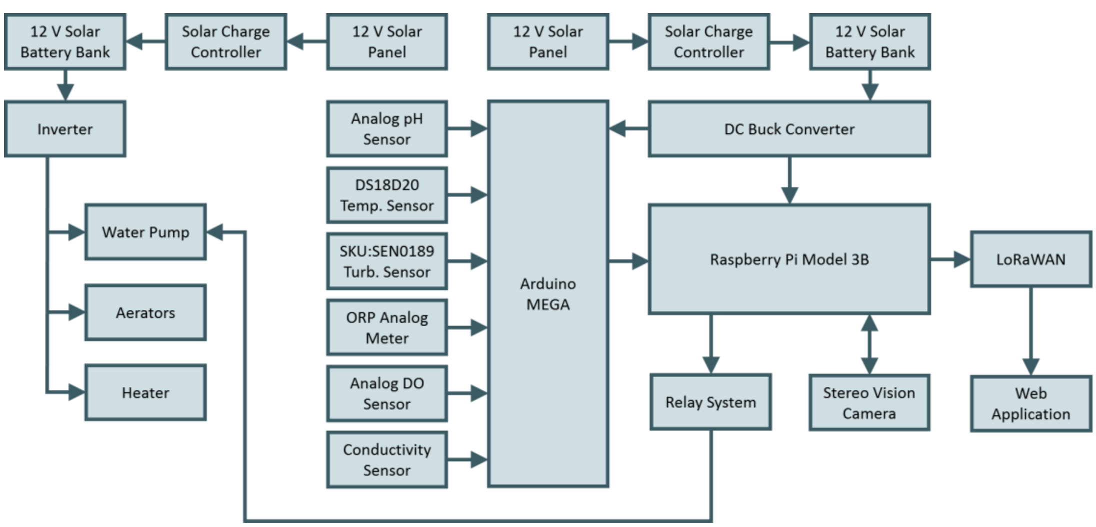


Figure 3.3 Block Diagram of the Aquaculture Setup

The aquaponics system was composed of three sections namely the water parameter detection and correction, fish growth monitoring system, and internet remote access.

3.1.3.1 Water Parameter Detection and Correction

Rearing of tilapia includes maintenance tasks in terms of water quality which is monitored by six different water parameter sensors. The water parameters of the aquaponics system such as turbidity level and water temperature were monitored and controlled to achieve optimum growth of the fishes in the tank. All sensors that is connected to Arduino Mega was set with threshold values to sustain the subsistence of Tilapia in fish tank while correcting devices were automatically turned on when the corresponding parameters were at critical levels. All of which can be discerned through the website.

3.1.3.2 Fish Growth Monitoring System

From the stereo-vision camera capturing the fishes in the aquarium, the captured image with the best side view of the fish is selected and then processed by the Raspberry Pi module for creating stereo, acquiring disparity values and return of disparity of pixel for the first camera. It is also used for object detection, image processing and eye tracking. The program acquires the projected area of fish using image processing and predicts the corresponding weight according to the test data that was fed to the system. The monitoring of fishes can be done through the website which will show data from the selected timeframe for which the capturing of image will take place.

3.1.3.3 Internet Remote Access

The Internet Remote Access includes the transmission and reception of data reports between the system and the web application through LoRaWAN. The output obtained from the parameter detection and its analysis on Raspberry Pi as well as the data acquired from the fish growth monitoring system were transmitted and were stored to the database. The values were visualized in a web application where we can also monitor the real-time state of the system.

3.1.4 Data Gathering of Related Facts or Information

Aquaculture

As the principal topic of this study, a foundation of knowledge concerning Aquaculture is necessary for the benefit of the Researchers. These include the correct manner of execution, the cultured species used and the optimal water quality parameters.

Nile Tilapia

Research concerning biological necessity and behavioral study are needed to be considered for the proper cultivation of the chosen species on an intensive aquaculture setup and how it responds to the provided environment.

Fish Growth Monitoring

Prediction-based fish growth monitoring required a deep understanding of stereo-vision and image processing as the instrument for acquiring fish weight. Studies of utilizing more effective solutions are needed to supersede the time-consuming and crude ways of measuring fish growth.

Water Parameter

Researching about the optimal water parameter for the development of the cultured fish is of equal importance. Dissolved oxygen, pH, temperature, turbidity, and salinity as well as the concept of correction was needed for the growth of the fishes.

Confinement

The advantages and disadvantages of using glass or acrylic as a means of confinement for the cultured species were considered. Research on how a material would best affect the growth the species is needed.

Searching for location

Location for the aquaculture set up was chosen deliberately. The proponents sought for an aquaculture site suitable for backyard fish culturing with factors to consider namely, conducting of testing, gathering and evaluation of data to complete the study.

3.2 Project Development

The development of this research project is divided into two parts namely the hardware development and the software development.

3.2.1 Hardware Development

Figure 3.4 Aquaculture Setup

The proponents used two aquariums for breeding Nile Tilapia. One aquarium was configured to emulate the environment normally used in traditional aquaculture; the other was equipped with stereo-vision camera and sensors. Certain modifications are created for the experimental setup to be fully functional according to the researcher's desire. Multiple sensors are attached to the system to identify the water quality parameters needed to be maintained such as water temperature sensor, dissolved oxygen sensor, oxidation-reduction potential, salinity or water conductivity meter, pH sensor and turbidity sensors. Hardware was composed of water quality parameter correctors such as aerators and water pump connected to a relay circuit system. Also included in the Hardware are Raspberry Pi as a microcontroller for the whole system and LoRaWAN as an IOT gateway. The solar power technology is also included on the hardware development which serves as the source of power for the whole system.

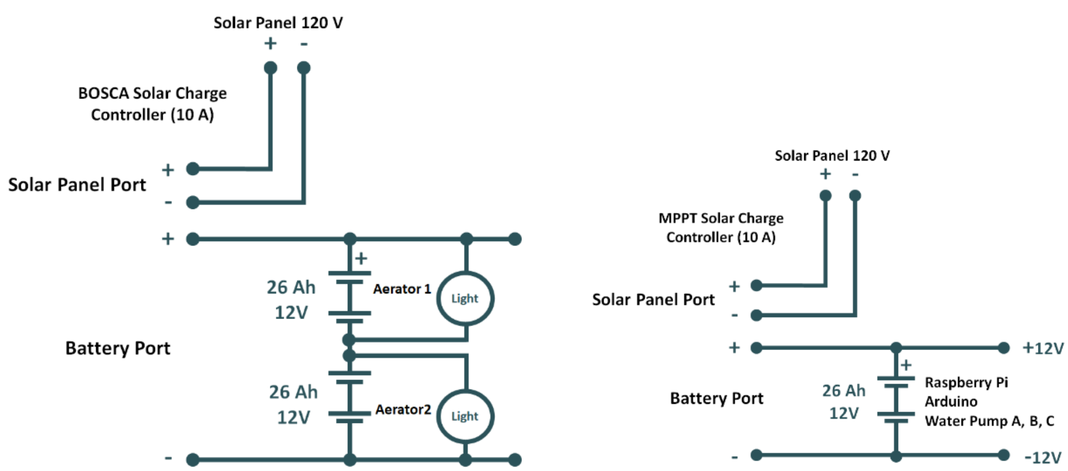


Figure 3.5 Solar Power Circuit Diagram

Figure 3.5 show the circuitry of the solar photovoltaic system. The entire system is powered by a couple of 265W solar panels together with three solar panels connected to the two solar charge controllers (Maximum Power Point Tracking) in parallel to the six 12V batteries. One of the 120W solar panel is dedicated to one of the solar charge controller (MPPT) that charges a 12V 26Ah battery that provides power for the Arduino, Raspberry Pi, water pumps, heater and automatic feeders. The 240W solar panel is a combination of two 120W solar panels in series configuration connected to the other solar charge controller (BOSCA) which supplies to the two 12V 26Ah batteries that were attached to the water pumps aerators and lights.

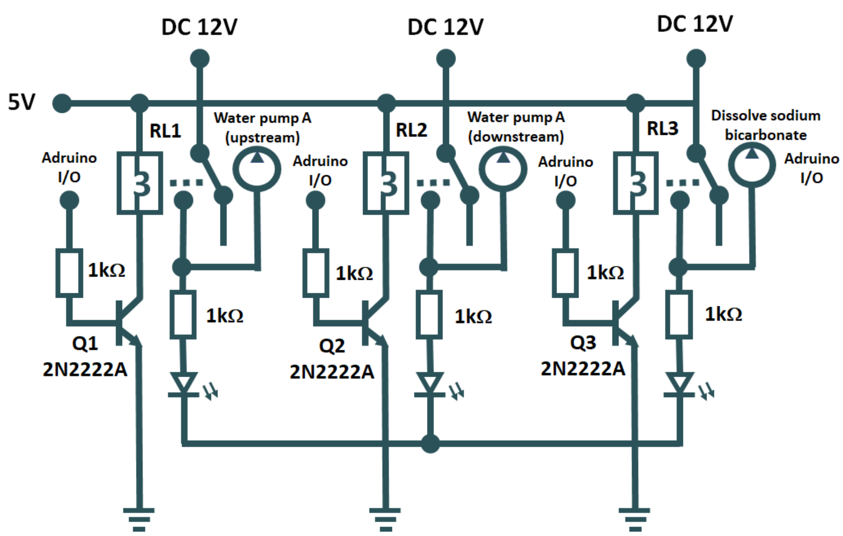


Figure 3.6 Relay System Circuit

As shown in Figure 3.6, the relay driver circuit is responsible for the water pump for the upstream and downstream of water for the correction of some water parameters. Two of the relay drivers are set to drain only half the amount of water in the tank every time they are activated to maintain the nutrients present in the water. The other relay driver is intended for pH correction.

3.2.2 Software Development

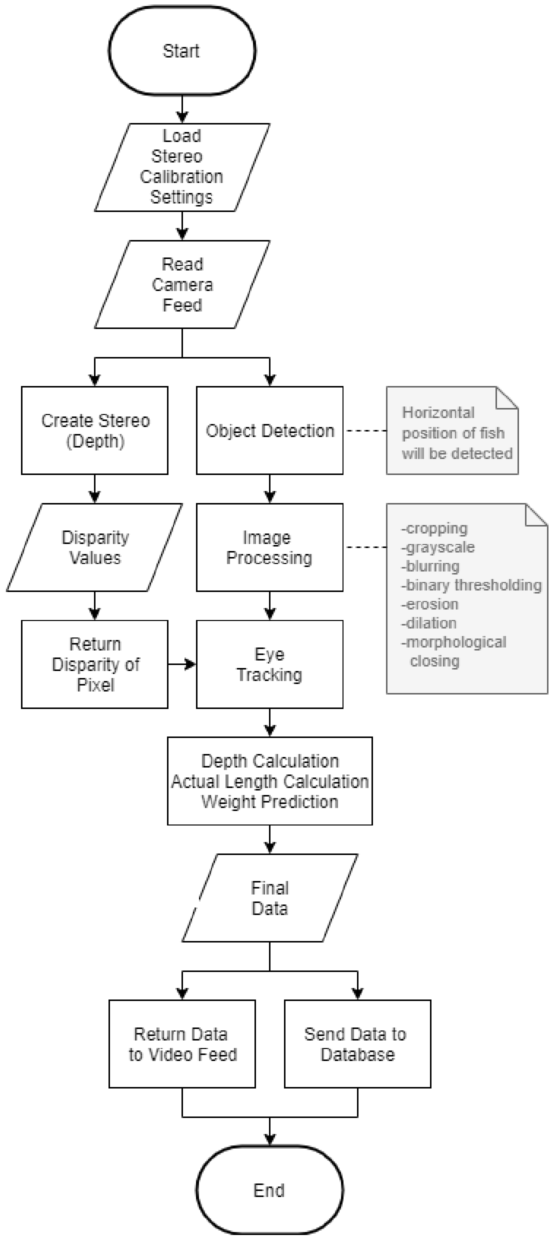


Figure 3.7 System Flowchart for Fish Weight Prediction

Figure 3.7 shows the Fish Weight Prediction process. The system makes use of two low-cost USB camera to implement stereo-vision technology. Distance measurements are then accurately determined by loading tested calibration settings. This enables simultaneous object detection and depth map creation after reading the camera feed. A series of image processing techniques were applied to the detected images of Nile Tilapia with the aim of removing noises and unwanted data, filtering only its contour. Eye-tracking technique was performed after obtaining the contour; this will pin-point consistent part of the fish as basis of its distance from the camera. From the contour of the fish, its pixel length was converted to an actual measurement by the use of pixel/metric conversion. Lastly, actual length was used for weight prediction through predictive analysis.

3.2.2.1 Distance Measurement

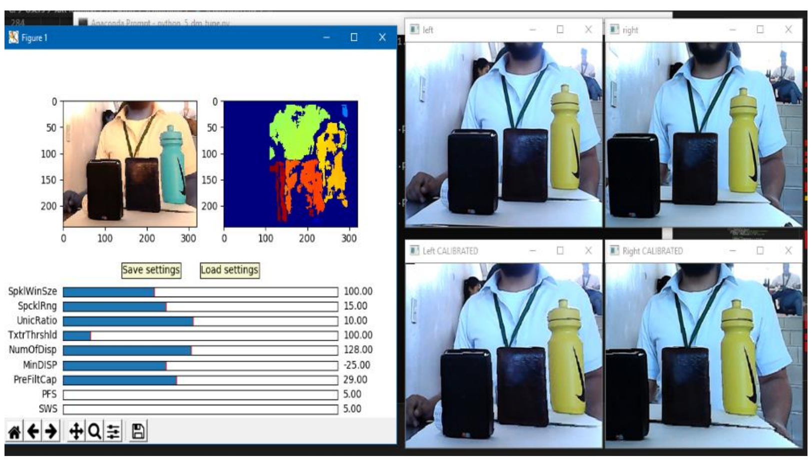


Figure 3.8 Stereo-Vision Camera Calibration Settings

A stereo-vision camera is capable of perceiving depth using two or more lenses with separate image sensor for each lens. The proponents were able to emulate these capabilities by calibrating two low-cost USB cameras. Figure 3.8 shows calculation of the disparity value between frames from the left and right camera.

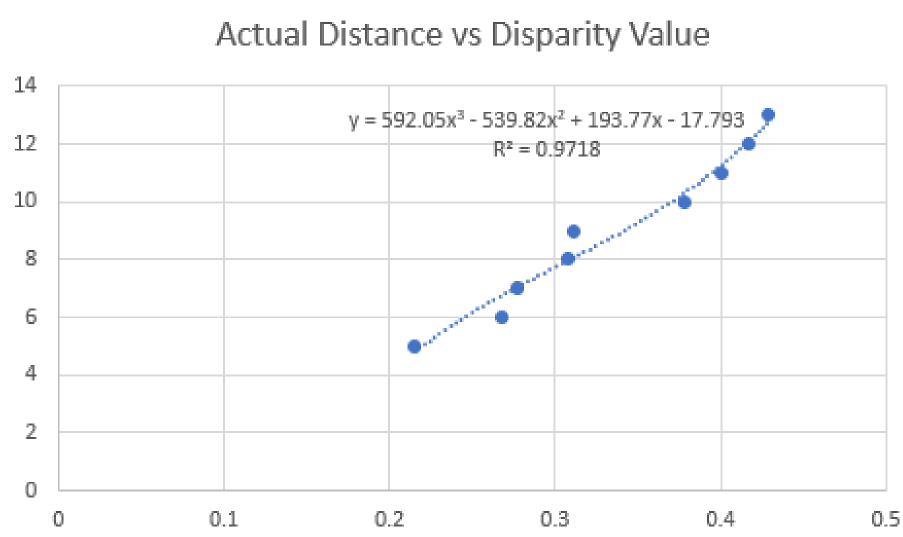


Figure 3.9 Actual Distance vs Disparity Value

Figure 3.9 shows the Regression analysis between the actual distance and disparity value. The acquired equation for the distance of the fish in terms of pixel will be calculated using the equation below:

$$\boldsymbol{D = 0.1017z^3 - 4.8944z^2 + 93.44z - 583.06}$$

Where: D = distance in terms of pixel

z = disparity value

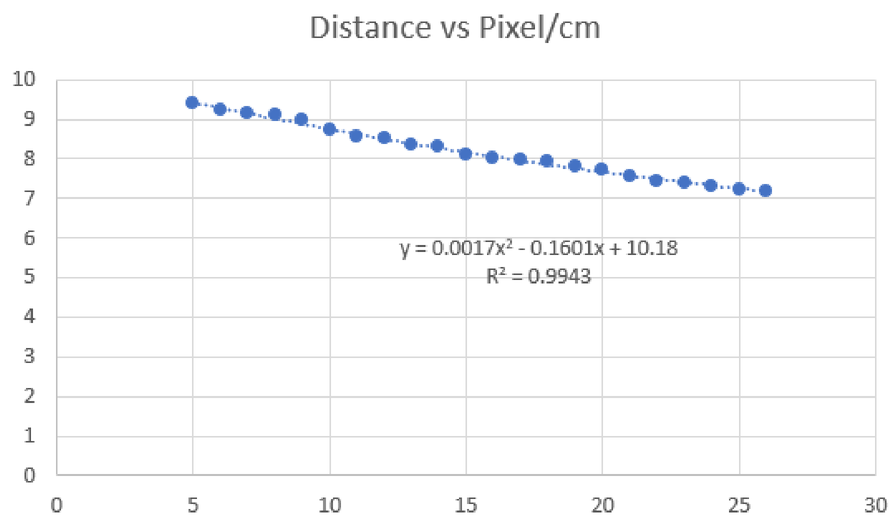


Figure 3.10 Actual Distance vs Disparity Value

Using regression analysis as shown in Figure 3.10, actual distance of the fish from the camera will be determined by calculating the pixel/cm ratio of the obtained virtual length. This is used to calculate the length of the fish that is needed for weight prediction.

The actual distance of the fish following pixel/cm conversion is represented as:

$$\boldsymbol{ppcm = 0.0017z^2 - 0.1601z - 10.18}$$

3.2.2.2 Image Processing

Image processing is composed of different sections: cropping, where an image is divided into two regions of interest;

grayscale, which is used to convert an image into shades of gray with no apparent color; blurring, a relative motion between an out-of-focus camera and a specific object; binary thresholding, which outputs a binary image; erosion, used to remove pixels on object boundaries; dilation, used for adding pixels on object boundaries; and morphological closing, used for filling small holes of an image from an image without affecting the size and shape in the process.

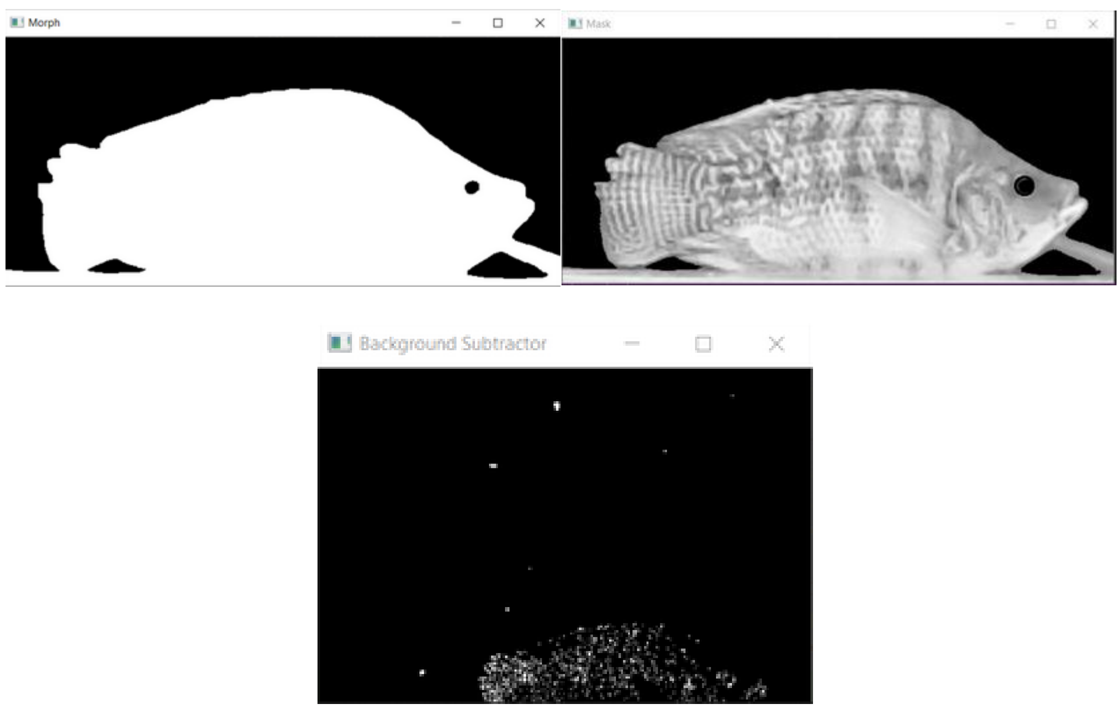


Figure 3.11 Image Processing

Figure 3.11 shows the image processing once the fish is detected. After the contour was drawn, circle detection takes place to detect the eye of the fish for basis of distance measurement. Hough Gradient Method is applied in order to detect circles. This method uses the gradient information of edges.

3.2.2.3 Predictive Modeling

One of the main objectives of this project is to formulate a predictive model for the fish dimensions in correlation with its corresponding weight. The measured length from image processing will then be used to determine the mass of Tilapia.

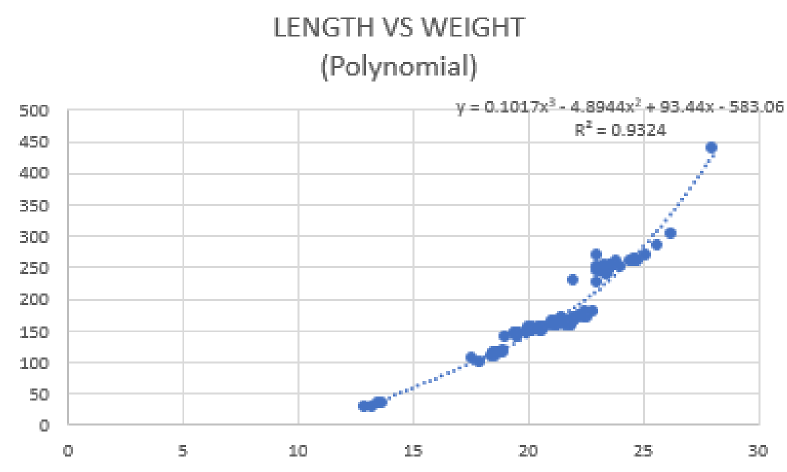


Figure 3.12 Length in cm vs Weight in grams (Polynomial)

Figure 3.12 shows the correlation of length and weight in a Polynomial trendline. The value of the R^2 acquired is 0.9324. The weight prediction derived using polynomial regression is represented as:

$$\boldsymbol{w = 0.1017x^3 - 4.8944x^2 + 93.44x - 583.06} \quad \textbf{(2)}$$

Where: w = weight of the fish predicted

x = length measured

Different trendlines such as Polynomial, Exponential, Linear and Logarithmic were examined to determine the most suitable for the data gathered. The value of the R^2 was used to determine the fitness of the line. The closer it is to the value of 1, the more appropriate it is for the dataset trendline.

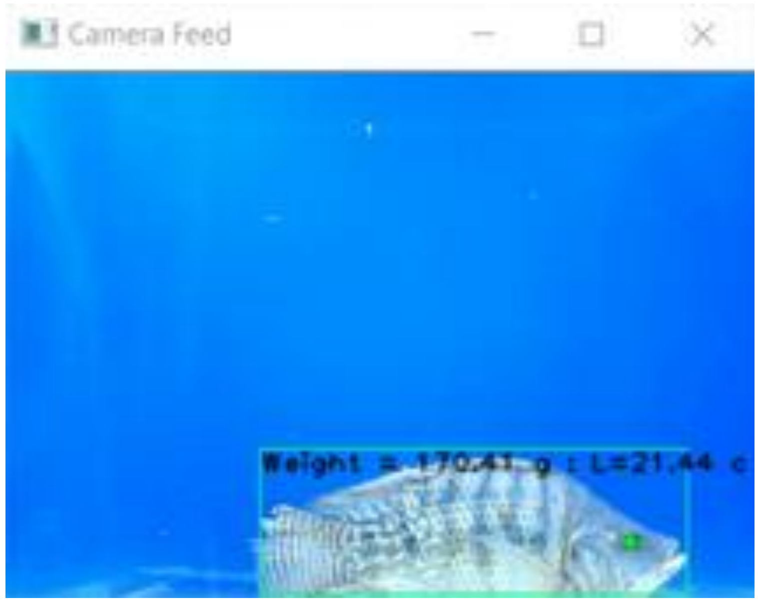


Figure 3.13 Weight Prediction

Figure 3.13 shows the actual fish detection length measurement and weight prediction. The contour of the fish was converted to its pixel length using pixel/metric conversion to find its actual length.

3.2.2.4 Sensor Calibration and Correction Process

3.2.2.4.1 pH Sensor

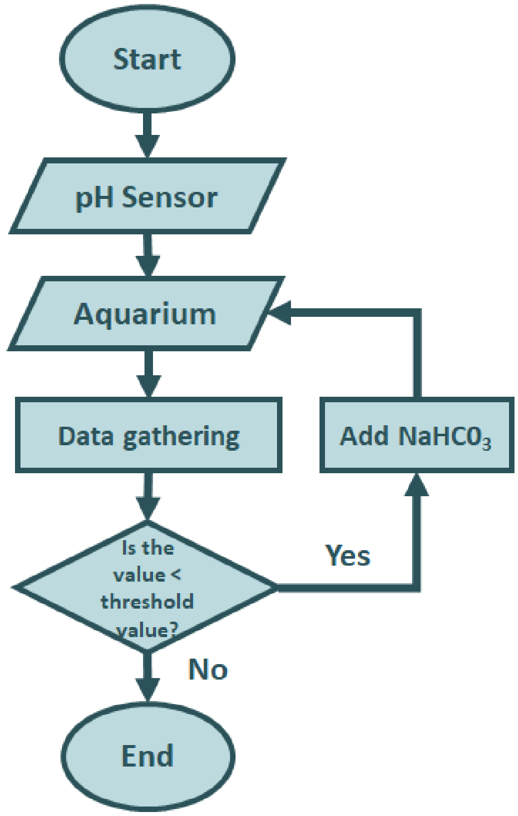


Figure 3.14 System Flowchart for pH Data Gathering and Correction

The study uses Industrial Analog pH sensor which measures the water's acidity and alkalinity. From the sensor node, the readings were first sent to the analog readout circuit (ARC) for voltage amplification, and then sent to the Raspberry Pi for data processing.

Figure 3.14 shows the flowchart following the data acquisition from the sensor node including the corresponding corrective method when the pH level goes beyond the threshold value. Tilapia can tolerate a wide pH range which is pH 5.0 - 10.0 (Thorarinsdottir, 2015) and based from it, the threshold range for the pH level for the system is set from 6 to 8 ppm. If the pH level goes beyond the limit, the corrective device, water pump, automatically releases Sodium Bicarbonate ($NaHCO_3$) to add base and increase the pH slowly until the readings return at the normal pH range.

3.2.2.4.2 Temperature Sensor

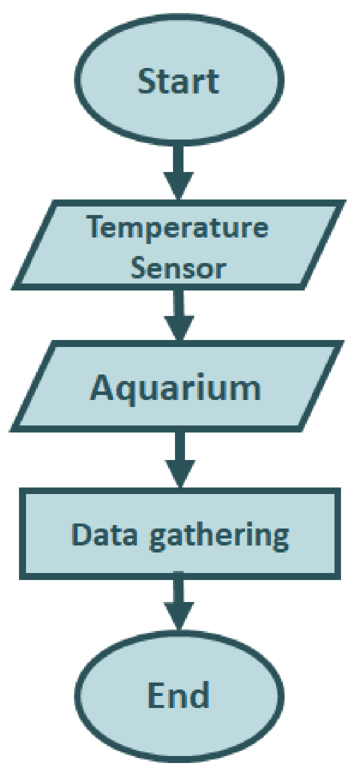


Figure 3.15 System Flowchart for Temperature Data Gathering and Correction

Water temperature affect the capacity of water to hold dissolved oxygen. Cold water at high atmospheric pressure holds more dissolved oxygen than warm water at low atmospheric pressure (University of Wisconsin, 2006). The DS18B20 water temperature sensor that was used provides Celsius temperature measurements.

3.2.2.4.3 Turbidity Sensor

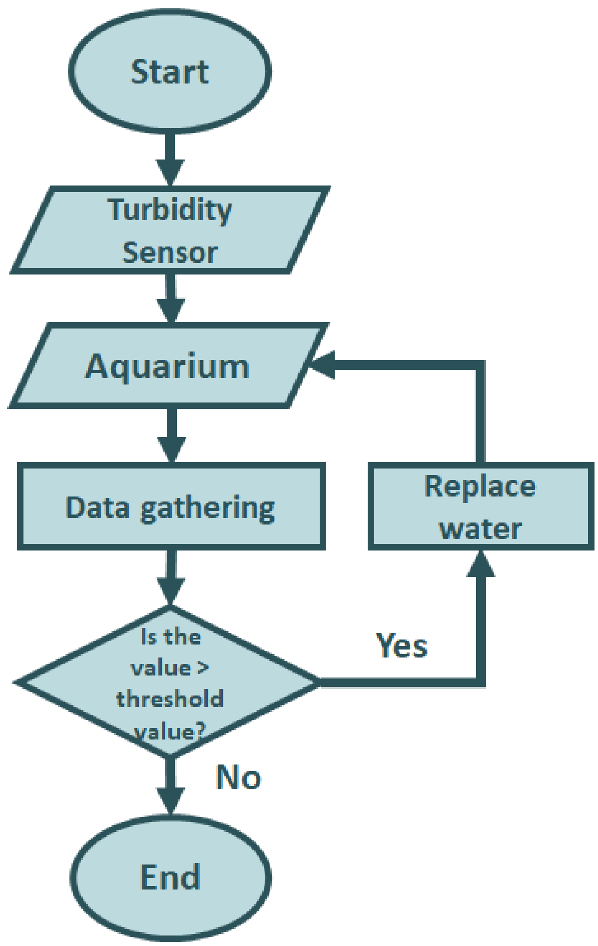


Figure 3.16 System Flowchart for Turbidity Data Gathering and Correction

Turbidity is described as the overall physical appearance of water and it is often obvious to know if the turbidity level is high enough to replace the water which is the only way to correct the said parameter. Though turbidity is easy to predict, it is important to still monitor it since it is considered as one of the most important parameters that is often disregarded. In this study, we use the SKU: SEN0189 turbidity sensor which measures turbidity level of water in terms of NTU or the Nephelometric Turbidity Unit. The values of the turbidity of water that is included in the range of 5-10 NTU characterizes a slightly turbid water and suitable to the best development of *O. niloticus.* (Allouko, et al., 2016).

3.2.2.4.4 Salinity Sensor

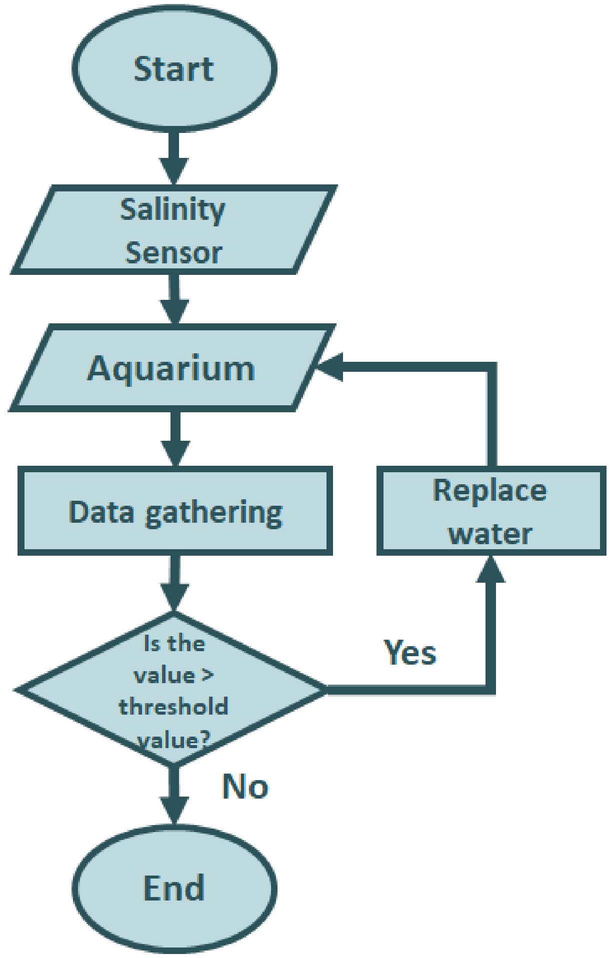


Figure 3.17 System Flowchart for Salinity Data Gathering and Correction

Salinity and Turbidity comes together as for water replacement is the only way to correct both, but we should also consider that Tilapia is one of the species that can tolerate high level of salinity even if they are considered freshwater fish. The Nile tilapias can reproduce in salinities up to 10 to 15 ppt but perform better at salinities below 5 ppt and when the water quality exceeds the required limit, considering other parameters, water replacement should take place.

3.2.2.4.5 Oxidation Reduction Potential (ORP) Sensor

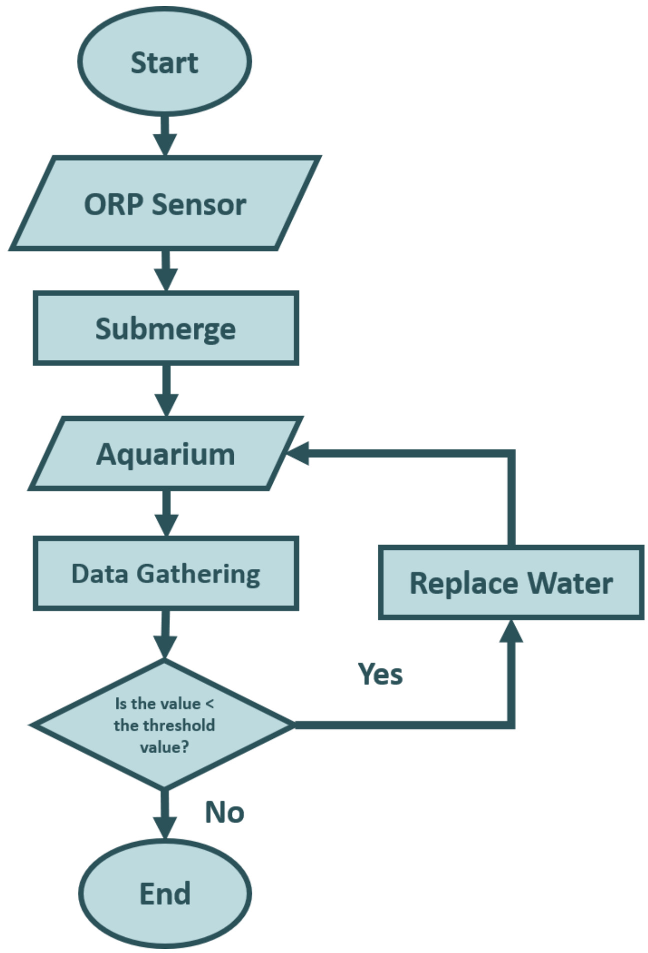


Figure 3.18 System Flowchart for ORP Data Gathering and Correction

Solutions that maintain high level of ORP are considered to have killing powers, which translates to a sterile environment. The killing power refers to a solution's ability to, in essence, kill the microorganisms inside that solution, causing a higher mV reading and an increased ORP. (Macri, 2017)

Figure 3.18 shows the flowchart for salinity data gathering that includes the correction process. Though there are solutions that can possibly increase the level of ORP in an intensive aquaponics set-up, replacing the water is the cheapest and easiest way to bring back the ideal value for aquaculture which ranges from 150 mV to 250 mV.

3.2.2.3.6 Dissolved Oxygen (DO) Sensor

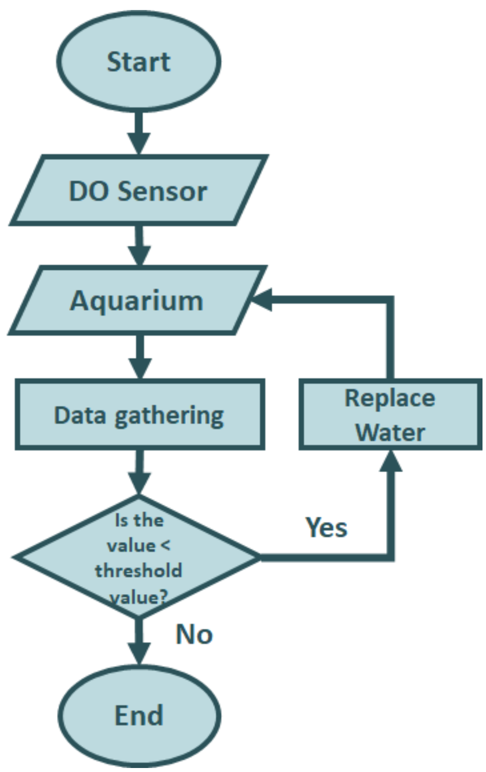


Figure 3.19 System Flowchart for DO Data Gathering and Correction

High turbidity and temperature level in an aquaculture set-up denote low level of Dissolved oxygen which means that correction of DO jointly corresponds to replacing the water. It is important to include DO level in a controlled environment since extremes level of it is fatal to fishes. Tilapia can tolerate up to less than 3mg/L that may be fatal for other farmed fish, but a level of 4mg/L of DO is the most preferable. (Popma & Masser, 1999)

3.2.2.5 Gateway Remote Management for LoRaWAN

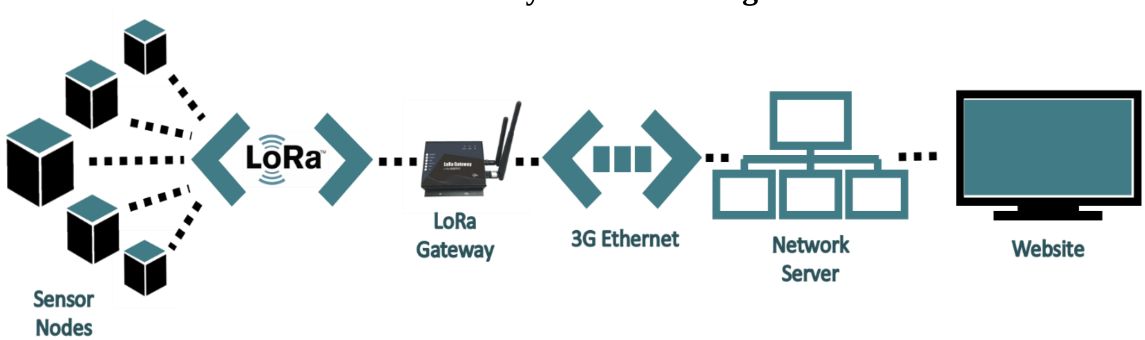


Figure 3.20 Gateway Remote Management for LoRaWAN

LoRaWAN (Long Range Wide Area Network) is designed to allow low-powered devices to communicate with internet connected applications over long-range wireless connections which makes it most suitable for a number of sensors that needs to be monitored using remote access. With that said, LoRaWAN plays a vital role in the water quality monitoring of this study. From the sensor nodes, the data gathered will first go through the Raspberry

Pi to be analyzed and from the Raspberry Pi, the data will pass through the LoRaWAN Gateway which will be the one to send the data to the internet then to the network server. From the internet, uploaded information will be available through the webpage that will be created. Figure 3.15 shows the diagram on how the data flows through LoRaWAN gateway.

3.3 Materials and Equipment List

Material	Quantity	Cost
Gravity: Analog Dissolved Oxygen Sensor	1	12,000.00
Gravity:Analog Turbidity Sensor	1	1,249.50
Gravity: Analog Electric Conductivity Sensor	1	4,999.75
Waterproof Temperature Sensor DS18B20	1	129.75
Industrial Analog pH Sensor	1	3,199.75
ORP Analog Meter	1	4,997.50
Raspberry Pi Model B	1	5,377.50
Arduino Mega 2560	1	699.75
LoRa IoT Development Kit	1	5,931.60 + SF: 1,774.41
Solar Photovoltaic Panel	1	25,000.00
Solar Charge Controller	1	600.00
DC-DC Buck Converter	1	5,972.00
A4 Tech PC Camera	2	495.00
4-Channel Relay Module	1	200.00
DC 12V Water Pump	3	165.00
Servomotor	1	210.00
12x12in-Acrylic Board	1	125.00
260-Liters-Aquarium	2	3,000.00
Aquarium Stand	1	1,500.00
200-Liters-Blue Plastic Drum	1	600.00
Aerator	1	300.00
LED Strip 5m	2	249.75
Adaptor 9V 1A	3	124.75
5m-Clear Hose	1	25.00
Water Heater	1	450.00
Water Pipes	3	50.00
	TOTAL	**83,520.26**

Table 3.1 Materials and Equipment List

3.4. Testing Procedure

Multiple sensors were used to measure the different water quality parameters by submerging each to the water. Each sensor has a threshold value programmed to maintain the preferred water quality parameter. Industrial Analog pH Sensor will be used to measure the pH level of the water with an ideal range of value from 6.0-8.0. The temperature of the water will be measured using DS18B20 sensor Nile Tilapia's ideal temperature for optimal growth lies between 27°C to 29°C. The amount of dissolved oxygen will also be measured using the Atlas Scientific Dissolved Oxygen Kit. An amount of .4mg/L is preferred to be maintained for the whole system. Atlas Scientific Electrical Conductivity sensor is used to measure the level of salinity of the water. A range of 15ppt to 20ppt is desired for most freshwater fish. Atlas Scientific ORP sensors must detect a range of value from 150mV to 250. The level of turbidity of the whole system is measured using a turbidity level sensor kit.

All sensors used to measure all the water quality parameters will be regularly tested and calibrated. The result of the testing procedure was read and processed by the Arduino then sent to the Raspberry Pi for transmission. At the same time, each corresponding sensor has a correction system triggered by the Arduino as it detected values exceeds the programmed range of threshold value.

3.5 Evaluation Procedure

The fish growth monitoring system was evaluated by assessing the accuracy of the Weight Prediction Algorithm that correlates the length of the fish to its weight. Under multiple data samples captured using the Stereo Vision Camera, the results from the algorithm will then be compared to the actual measurement results from manual measurement of length and weight of the fish.

The results of the evaluation procedure were assessed by the panel members and adviser of this study from Technological University of the Philippines.

3.6 Gantt Chart

Activities	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Research for Project Study Topic															
Preparation for Topic Defense															
Topic Defense															
Preparation for Title Defense															
Consultation with the Advisers															
Title Defense															
Planning and Designing															
Canvassing															
Testing of Sensors and Correction Devices															
Programming															
Gathering of Data															
Preparation for Progress Report															
Progress Defense															
Setting up and Operation of the Project															
Pre-Final Defense															
Project Monitoring and Deployment															
Final Defense															
Finalizing of Documents															

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Project Technical Description

The system is divided into three major parts: Monitoring and Correction, Machine Vision Technology, and Internet of Things Technology. The acidity, turbidity, conductivity, oxidation-reduction potential, temperature and dissolved oxygen levels of the aquaculture environment are monitored for the optimal development of Tilapia. The correction system maintains these parameters within the admissible range as often as possible. Furthermore, the machine vision technology ascertains the size of the cultured Tilapia and determines if it is ready for harvest. The measured readings from the sensors and the information about the size of the fish are sent to the web application using IoT gateway to notify the user about the system status.

The Raspberry Pi is responsible for handling the processes of the entire system. Arduino MEGA is used for reading the analog and digital data from the sensors. The Machine Vision Technology utilizes stereo-vision to capture images of Tilapia at a constant time interval and determine whether the Tilapia are ready for harvest.

4.1.1 Aquaculture System Setup

The setup consists a 260-Liters aquarium for the fish, a 200-Liters blue plastic drum for water replacement, six (6) water parameter sensors, and a stereo-vision camera for capturing the weight of the fish.

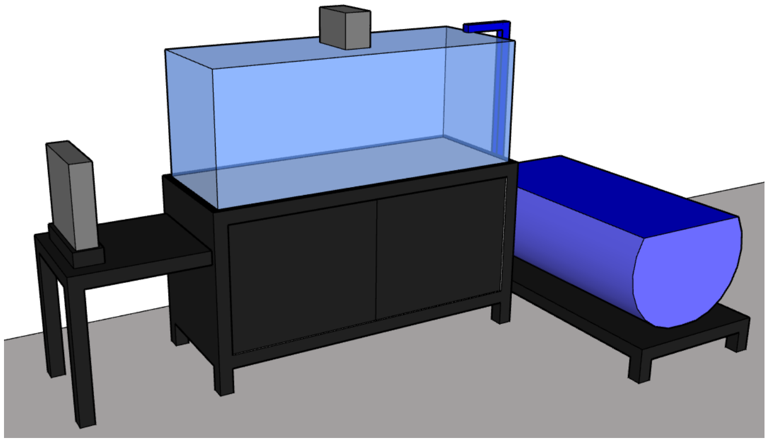
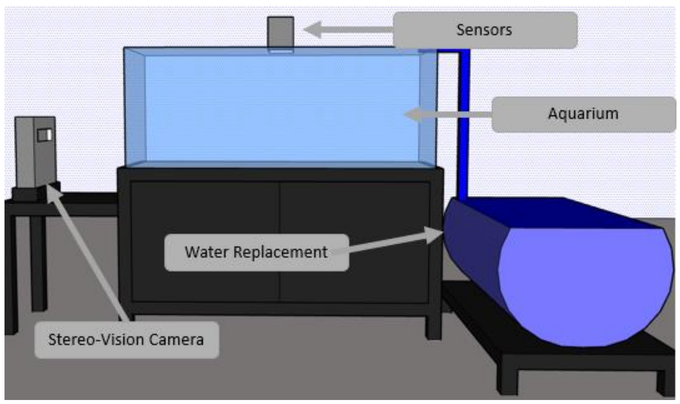
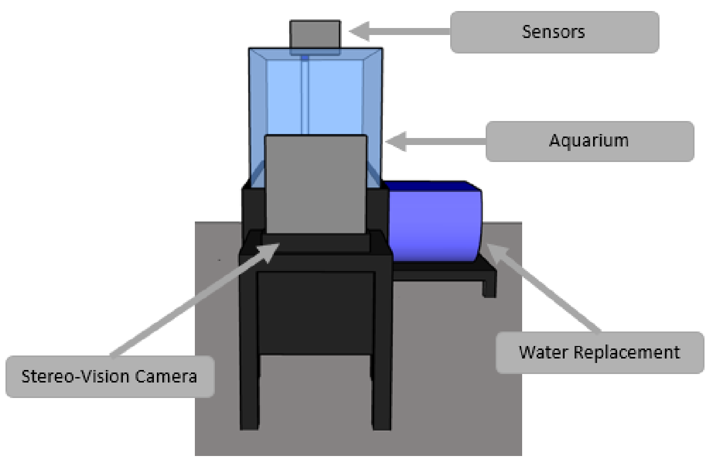


Figure 4.1 Isometric View of the System



(a)



(b)

Figure 4.2 System Setup (a) Front View (b) Back View

4.2 Project Structure

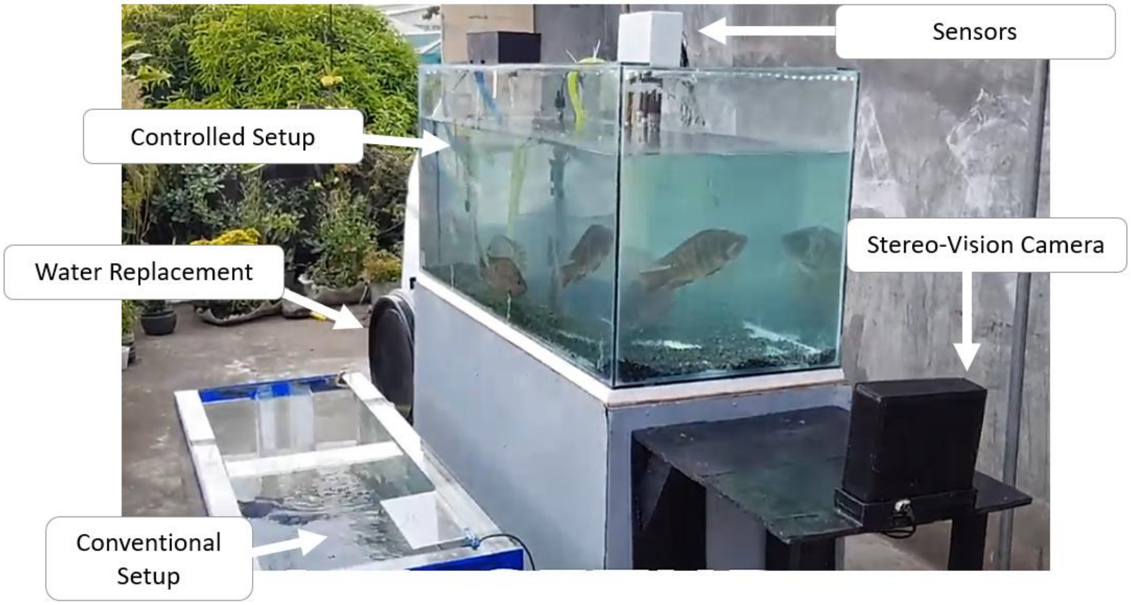


Figure 4.3 Aquaculture Setup

The Figure 4.3 shows the actual setup deployed. It is composed of two aquaculture system, conventional and controlled method of aquaculture. It has a stereo-vision camera installed at the right-most part of the system which captures the weight of the fish inside the aquarium. Above the aquarium are the sensors used for monitoring the water quality. Both aquariums have five (5) fishes but only the experimental setup has a water correction system.

4.2.1 Monitoring System Setup

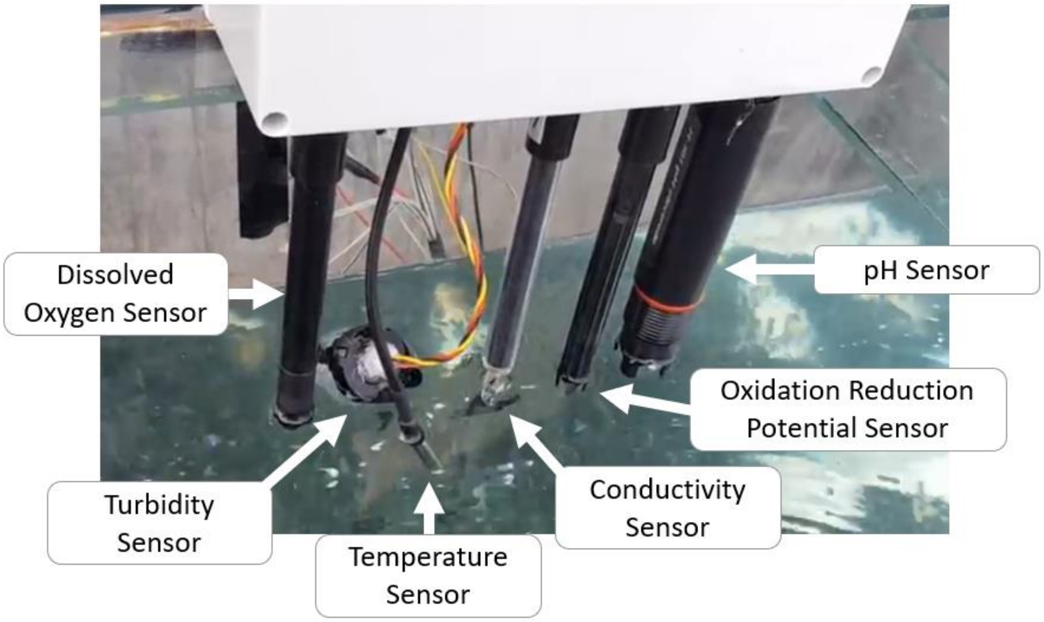


Figure 4.4 Water Parameter Sensors

The system has six (6) sensors wherein five are digital and one is analog. The temperature sensor, being an analog sensor, is connected to the analog input [2] of the Arduino; the pH, ORP, DP, turbidity and electric conductivity sensors are connected to the digital input [0], [1], [2], [3] and [4] of the Arduino respectively. It is important to note that the sensors are calibrated prior to being interfaced to the Arduino – which facilitates data collection for the Raspberry Pi. Once the Arduino is ON, it periodically prints the sensors readings into its serial monitor. The raspberry pi then converts the data into a file suitable for transmission using LoRaWAN to the LoRa Gateway that will upload the data to the TheThingsNetwork linked to the web server and can be viewed through the web app that will handle the presentation and visualization of data.

4.2.2 Machine Vision System

Figure 4.5 Stereo-Vision Camera Setup

Two low-cost USB camera were utilized to produce stereo-camera, implementing stereo-vision technology. They were enclosed in a black box to make them uniform and steady. After reading the camera feed, object detection and depth map creation were processed simultaneously. Once the contour was drawn, eye-tracking technique was performed to pin-point consistent part of fish as basis of its distance from the camera. Actual length was measured using the contour of fish, converting its pixel length to an actual measurement following pixel/metric conversion. Lastly, actual length was used for weight prediction through predictive analysis.

4.2.3 Web Application Interface

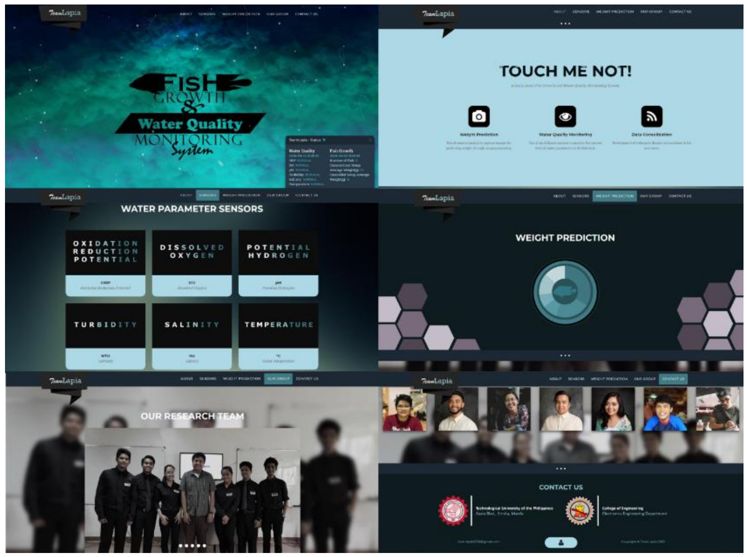


Figure 4.6 Preview of the Web Application Interface

The Web Application is divided into six different sections as shown in Figure 4.6. The first panel shows the header that says “Fish Growth and Water Quality Monitoring System” which is what our study is all about. As you scroll along the page, you’ll see the panel that describes the three main functions of the study each presented with the appropriate icons. The third part of the application is where the status of the sensors can be seen. There are six clickable rectangular panels that represents the six different water parameters of the system.

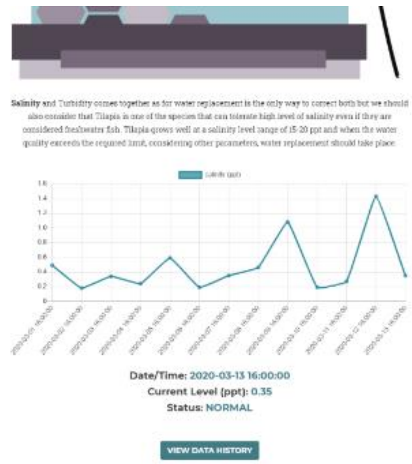


Figure 4.7 Preview of the Salinity Panel

Figure 4.7 shows what's inside the salinity panel located in the water parameter sensors section. A brief description of the parameter is included together with the graph of the past values gathered by the sensor. Also, a part where you can see the current level/status of the parameter is included in the pane. The fourth section of the application is where the growth of the fishes can be seen. As you click the pulsing circle in the middle of the page, you be directed inside the pane where you'll see the graph of all the past measurement of the fishes for both controlled and conventional setups including the current information of it just like the pane shown in figure 4.7. Right below the weight prediction panel is the part about the research team with their pictures and corresponding clickable social media handle. The footer of the application is allocated for the contact information in case of inquiries including the social media links associated to the study.

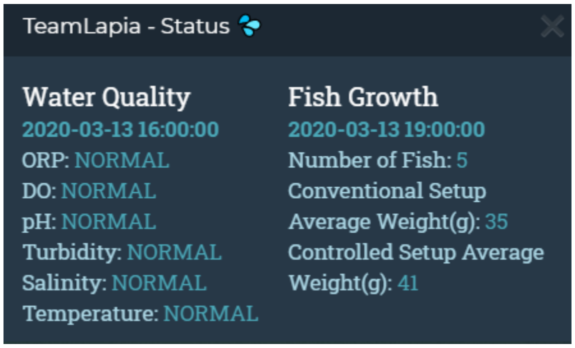


Figure 4.8 Preview for the Summary of the Current Status

The last part of the page shows a pop-up part of the web application that summarizes the current status for both water quality and fish growth system in one place as seen in Figure 4.8. It can be found on the lower right part of the interface which can be minimized or closed anytime.

4.3 Project Limitations and Capabilities

The website application displays the status of the system with the most recent data whenever possible. The preceding data together with the newer ones, are represented in a graph to show the trend of fish growth. This data includes the fish weight and sensor readings. To determine the weight of the fish, the system uses a low-cost camera that utilizes stereo-vision technology. This weight will provide information for the user if the fish are ready to harvest. The system is equipped with six sensors: ORP, DO, pH, EC, Temperature and Turbidity to monitor the water parameter. Once a sensor reading exceed normal values, the correction system initiates.

The image processing technique used in the study demands images where the fish being measured is properly contrasted to its surroundings. In which case, higher quantities of fish cultured may affect the outcome of weight prediction.

4.4 Project Evaluation

The system is deployed from 1 March 2020 to 13 March 2020 for a total of 13 days. The functionality of the system was evaluated through daily and weekly data gathering.

4.4.1 Weight Prediction Accuracy Results and Discussion

Fish Sample	Weight (g)		Difference	Error (%)
	Measured	Predicted		
1	105.5	110	-4.5	4.09 %
2	97.5	105	-7.5	7.14 %
3	148.4	150	-1.6	1.06 %
4	141.6	150	-8.4	5.6 %
5	152.4	150	2.4	1.6 %
6	104.7	105	-0.3	0.28 %
7	147.5	150	-2.5	1.67%
8	134.8	135	-0.2	0.15 %
9	108.4	110	-1.6	1.45 %
10	104.3	110	-5.7	5.18 %
		Mean Percentage Error		**2.82%**

Table 4.1 Experimental Data for Predicted and Measured Weight of Sample Fishes

To test the accuracy of the Weight Prediction Algorithm, a comparison was made between the actual measured weight of the fishes and the predicted weight. Table 4.4 shows the recorded weights of ten fish samples and their corresponding percentage error. With the data assimilated, it was inferred that the measured and predicted weight were almost the same as it only has a mean percentage error of 2.82%. With that in mind, to further test the accuracy of the device, a hypothesis test was also conducted.

4.4.1.1 Statistical Analysis of Experimental Data for Predicted and Measured Weight of the Fish Samples

With the two means being related to each other as it measures the weight of the same samples, a paired t-test was used to determine the accuracy of the weight prediction algorithm. The null hypothesis is set to prove that there is no significant difference between the two groups. The significance level was set to 0.01.

t-Test: Paired Two Sample for Means		
	Measured	*Predicted*
Mean	124.51	127.5
Variance	491.8232222	445.8333333
Observations	10	10
Pearson Correlation	0.988461845	
Hypothesized Mean Difference	0	
df	9	
t Stat	-2.736983041	
P(T<=t) one-tail	0.011480323	
t Critical one-tail	2.821437925	
P(T<=t) two-tail	0.022960645	
t Critical two-tail	3.249835542	

Table 4.2 Summary of Paired t-test for the Predicted and Measured Weight of the Sample Fishes

In the summary shown on Table 4.5, the p-value for two-tailed test is greater than the significance level of 0.01 which means that we have to accept the null hypothesis that there is no significant difference between the measured and predicted weight of the samples. This implies that the weight prediction algorithm used to measure the weight of the fishes is accurate and is acceptable to use.

4.4.2 Water Quality Monitoring System

	pH (6-9)		Turbidity (0.3-5 NTU)		ORP (150-250 mV)		Temp (25-27 °C)		Salinity (<15 ppt)		DO (1-2.5 mg/L)	
Day 1	7.23	--	2.16	--	165.52	--	25.86	--	0.49	--	1.83	--
Day 2	5.30	6.49	2.50	1.85	198.17	190.82	27.81	25.85	0.18	0.25	2.05	1.92
Day 3	7.67	--	4.32	--	152.19	--	26.50	--	0.34	--	1.76	--
Day 4	8.69	--	2.55	--	154.72	--	25.86	--	0.24	--	1.53	--
Day 5	13.66	8.42	3.41	2.81	190.01	167.40	27.67	25.32	0.59	0.43	1.80	1.72
Day 6	7.31	--	3.14	--	168.81	--	26.43	--	0.19	--	2.45	--
Day 7	8.23	--	2.88	--	154.01	--	28.22	--	0.35	--	2.49	--
Day 8	7.90	--	1.67	--	182.27	--	28.60	--	0.46	--	1.56	--
Day 9	13.33	8.24	1.73	2.09	161.23	173.27	26.61	25.74	1.08	0.72	2.28	1.41
Day 10	7.69	--	2.90	--	192.47	--	25.95	--	0.19	--	1.31	--
Day 11	8.60	--	2.60	--	175.56	--	25.85	--	0.27	--	1.86	--
Day 12	9.74	8.35	4.08	4.04	160.95	176.62	26.72	26.19	1.43	0.83	2.23	1.75
Day 13	7.82	--	2.24	--	191.16	--	26.15	--	0.35	--	1.89	--

	Tabs in orange indicates the readings that exceeded the threshold limit for the parameter
	Tabs in gray indicates the readings after performing water correction on the same day

Table 4.3 Data Gathered for Controlled Setup

After the start of the deployment, the system gathers data every day. Table 4.1 shows the data gathered for each water parameter sensor throughout the 13-day period which specifies what parameter exceeds the threshold limit and the readings after each water correction. pH, as one of the most important water parameter, is what often dictates the system to start a water correction. It triggers the water correction system to put Sodium Bicarbonate ($NaHCO_3$) into the water until the pH level goes up to the normal range. It also the one who triggers water replacement. During water replacement, only half the water is replaced to maintain the environment's water chemistry and nutrients.

For the Conventional Setup, the water is left unchanged and unchecked throughout the week as it turned almost dark-green in color because of dirt and fish manures. The water is only changed every Sunday where fish weight is also measured during the day.

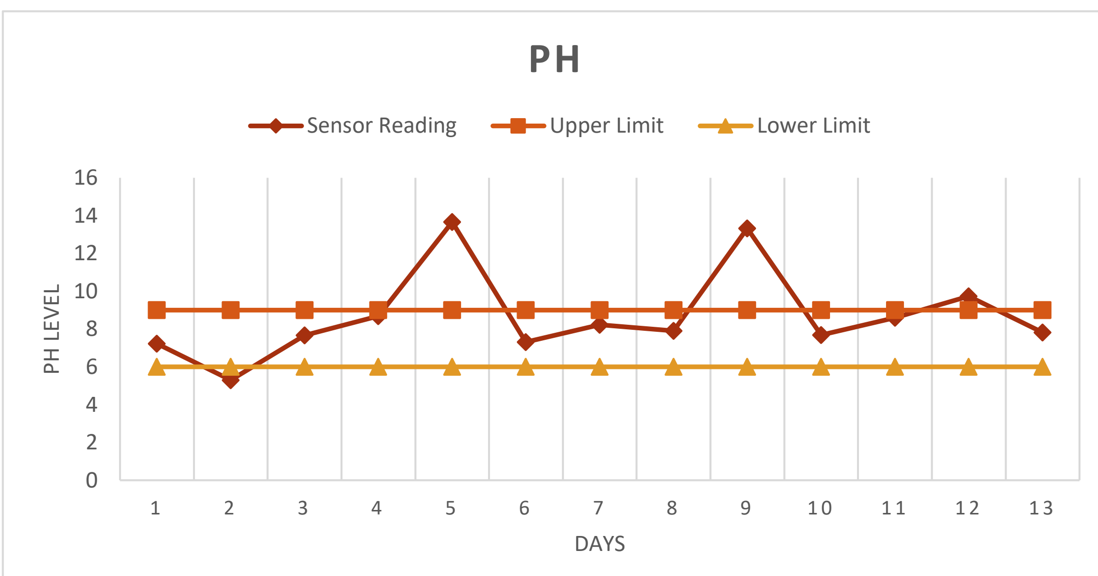


Figure 4.9 Potential Hydrogen Sensor Readings with Correction Response

Figure 4.6 shows the pH level for the aquaculture setup which corrects the water whenever the reading goes beyond the threshold level on the same day as it is measured. The ideal pH level for the *Nile Tilapia* is 6 to 9 (unitless).

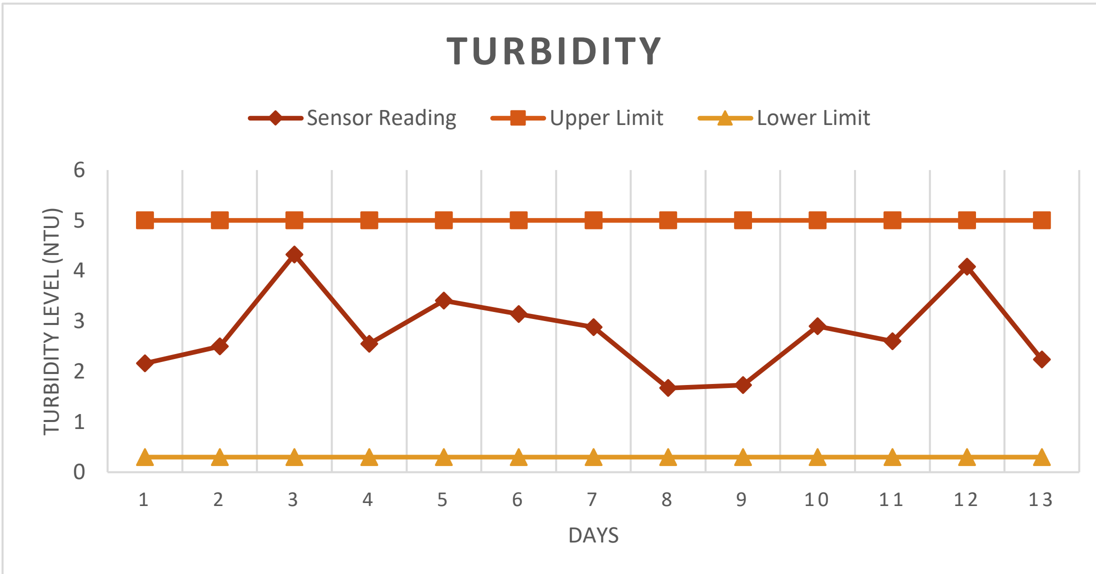


Figure 4.10 Turbidity Sensor Readings

Figure 4.7 shows the Turbidity sensor readings for the aquaculture setup. The threshold limit for the Turbidity which the fishes can tolerate ranges from 0.3-5 NTU.

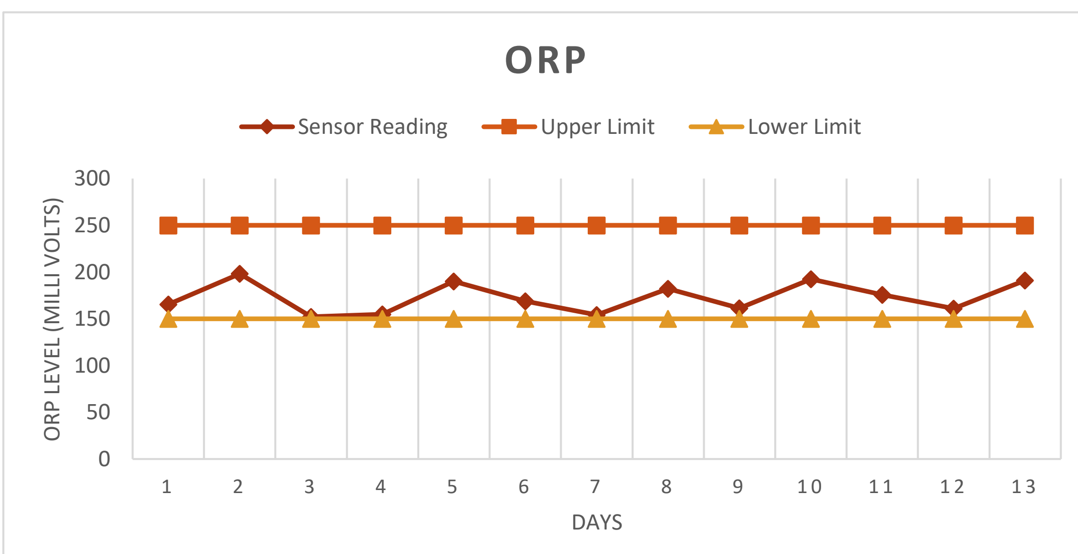


Figure 4.11 Oxidation Reduction Potential Sensor Readings

Figure 4.8 shows the graph of ORP sensor readings for the aquaculture setup throughout the deployment period. Oxidation-reduction potential (ORP) is almost peremptory to maintain as it measures the ability of a water to cleanse itself or break down waste products which is important for aquaculture setups. The ideal value of ORP for aquaculture ranges from 150mV to 250 mV.

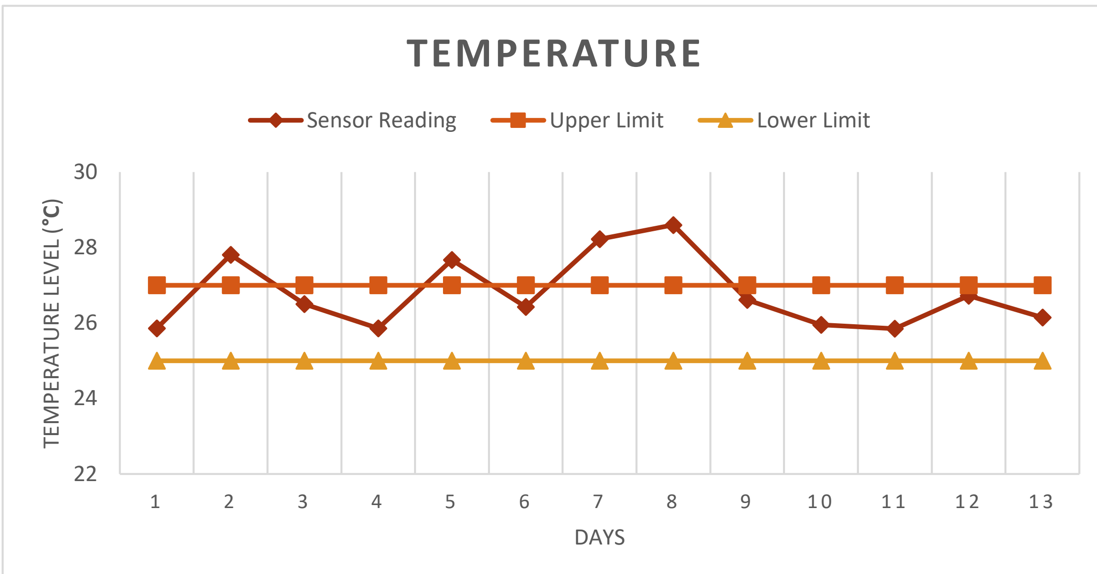


Figure 4.12 Temperature Sensor Readings

Figure 4.9 shows the Temperature sensor readings for the aquaculture setup. As tilapia is generally intolerant towards low temperatures, a water heater is then used to prevent the temperature from dropping. However, there's no correction needed whenever the temperature exceeds the upper limit. Optimal water temperature for tilapia growth is 25-27 °C.

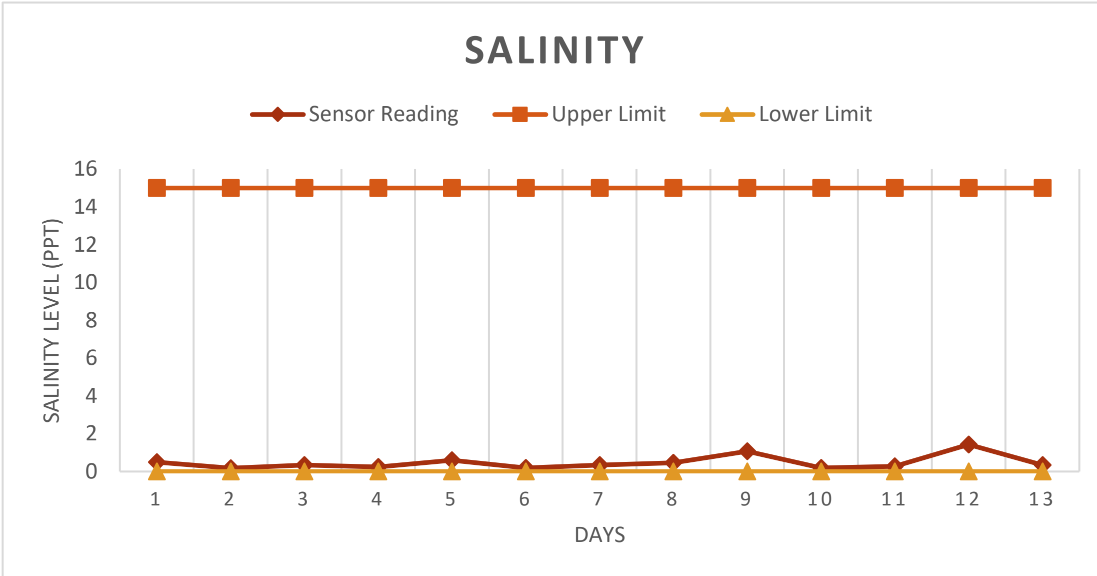


Figure 4.13 Salinity Sensor Readings

Figure 4.10 shows the Salinity sensor readings for the aquaculture setup. Salinity is heavily affected by the source of water and since Nile tilapias has a wide range of saltiness that they can tolerate and they perform better at salinities below 5 ppt, tap water is used in the setup. It is reflected by the low level of salinity shown in the graph.

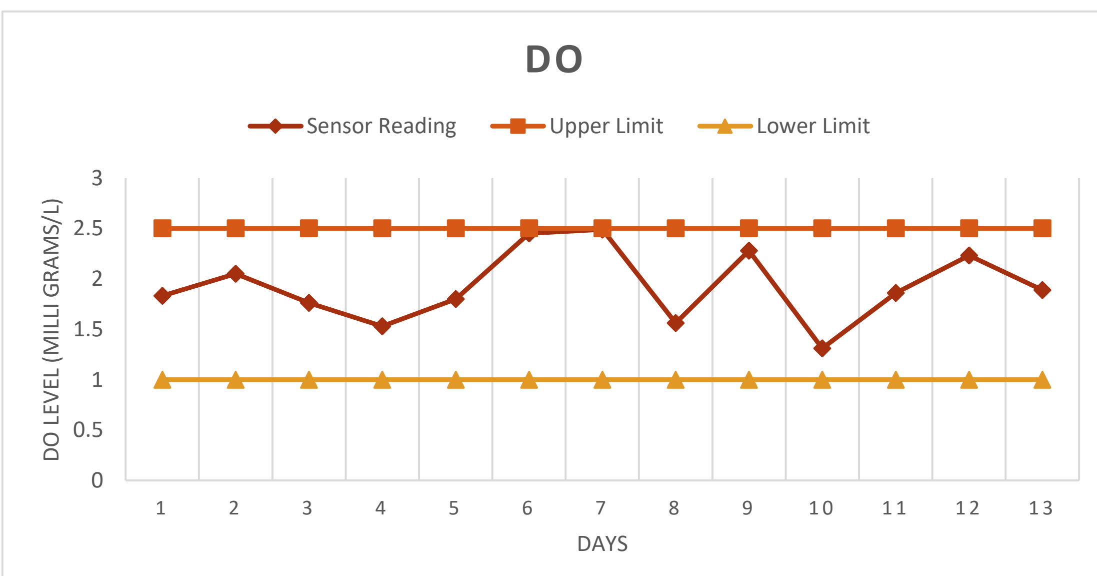


Figure 4.14 Dissolved Oxygen Sensor Readings

Figure 4.11 shows the DO sensor readings for the aquaculture setup. The desired range for Dissolved Oxygen level is from 1-2.5 mg/L. Since water is being replaced even before the DO concentration of the water goes beyond the threshold limit, the oxygen level is being maintained in the system with the help of an aerator.

4.4.3 Fish Growth

The weight of the fishes is measured weekly starting from the initial deployment date (Sunday) up until the second week. The data gathered is then averaged to measure the growth of the fishes as weeks go by.

Fish Weight (g) - Controlled Setup			
Fish No.	Initial	Week 1	Week2
1	20	25	35
2	20	30	35
3	25	35	40
4	25	35	45
5	30	40	50
Average	24	33	41

Table 4.4 Fish Weight in the Controlled Setup measured using Camera

Table 4.4 shows the weekly weight of the fishes in the controlled aquaculture setup taken using the weight-prediction system. The Average weight is computed every week which shows the steady growth of the fishes in the tank.

Fish Weight (g) - Conventional Setup			
Fish No.	Initial	Week 1	Week2
1	20	25	30
2	20	25	30
3	25	30	35
4	25	30	40
5	30	35	40
Average	24	28	35

Table 4.5 Fish Weight in the Conventional Setup measured manually

Table 4.5 shows the weekly weight of the fishes in the conventional aquaculture setup measured manually using a digital weighing scale. Similarly, the average weight is also computed every week to show the overall growth of the fishes.

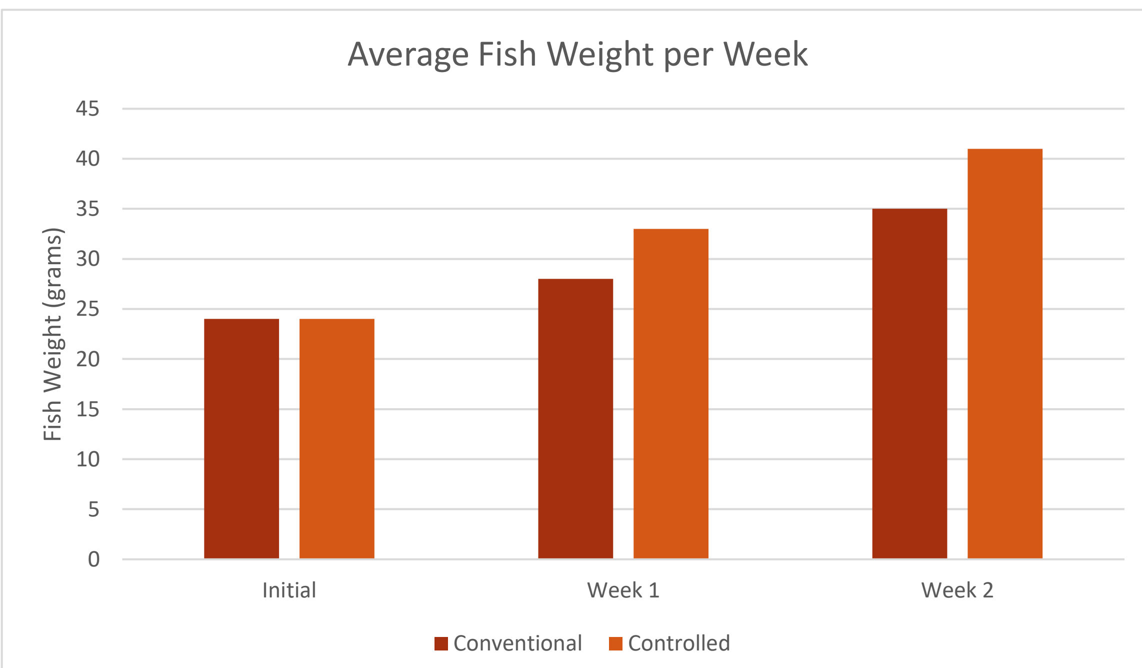


Figure 4.15 Controlled Vs Conventional Fish Weight per Week

Figure 4.12 shows the bar graph of the average weight of fish in both conventional and controlled setups. Initially, the weight of the fishes on both tanks are equal and the growth of the fishes is monitored in two weeks.

Days Elapsed	Average Fish Weight (g)	
	Proposed Aquaculture Setup	Conventional Setup
0 (initial)	24	24
7(week 1)	33	28
12 (week 2)	41	35

Table 4.6 Summary of Fish Growth in each setup

Based on the data shown on Table 4.6, the results show that the growth rate in the controlled aquaculture setup is 30.70% each week and is greater compared to the conventional setup which has 20.76% growth rate per week. The proposed aquaculture setup improves the growth of the fishes in terms of weight by 47.88%.

Both the aquaculture setup has five fishes each but only the controlled setup is being closely monitored starting from its vital water parameters up to the handling of fishes where weight prediction matters. The conventional setup, on the other hand, is being unmonitored. Aeration is given to both the setups as well as the daily feeding of fishes.

CHAPTER 5

SUMMARY OF FINDINGS, CONCLUSIONS AND RECOMMENDATIONS

This chapter presents the summary of findings, conclusions, and recommendation relative to the results of the study conducted.

5.1 Summary of Findings

The whole study is composed of two main functions: the weight prediction system and the water quality monitoring system, both of which monitors the growth of the fishes in the aquaculture setup. The following information summarizes the overall performance of the system based on the data gathered presented on the previous chapter:

For the water quality monitoring system:

The realized result of the study suggests that to achieve the optimal growth of the fishes particularly for Nile Tilapia (*Oreochromis Niloticus*) in an aquaculture setup, it is of utmost importance to closely monitor all of the critical water parameters there is. The stable condition of each parameter present signifies the good quality of the water and therefore improves the overall functionality of the system.

For the weight prediction system:

The development of weight prediction algorithm greatly affects the growth of the fishes as it lessens the stress-level received by them. Aside from the fact that it is more convenient to use instead of manual weighing, one can easily supervise the growth of the fishes anytime according to the optimal weight allowed for commercial consumption. It is reflected in the data gathered that using stereo-vision camera to predict the weight of the fishes is just as good as manually weighing them using a digital scale.

Overall, the performance of the system can be considered congenial for future use because it improves the growth rate of the fishes as it offers ease of use for the end users. Everything that is recorded by the system can be viewed and reviewed through the web application developed to compile and display the status of the system in the easiest way.

5.2 Conclusion

Based on the data gathered and test conducted, the proponents have come up with several conclusions:

1. The proponents have been able to design and develop a stereo-vision camera that predicts the weight of the fishes through image processing and predictive analysis.
2. The system effectively monitors parameters of the water needed for optimal environment to ensure a higher growth rate of the fishes. It also automatically corrects and replace the water when it reaches its threshold limit.
3. The proponents have been able to develop a website that consolidates and display information for the end-user.
4. The proponents were able to provide seamless connectivity of the system to the database using LoRaWAN.
5. The growth rate of the fishes from the controlled setup is eminent compared to the conventional setup which results to higher quality of the fish.

5.3 Recommendation

After thorough analysis of different factors and problems encountered in this study, the following recommendations are stated for further improvement:

1. To provide a newer version of Raspberry Pi for faster processing of data and a ventilation system for it to prevent the overheating of the device.
2. To use a Depth camera instead for more accurate distance measurement.
3. To use different fish species not just tilapia.
4. To automate ng fish feeding process to prevent spoiling of water because of left overs since it releases small amount of feeds at regular intervals instead.

References:

Agarwal, T. (2015). *Ion-Sensitive Field Effect Transistor - ISFET Working Principle.*

Agarwal, T. (2015). *Know About Different Types of Sensors with their Applications.*

Amora, S., Bartolata, D., Sarucam, J., & Kristine, S. (2017). *SMART AQUAPONICS IN A TEMPERATURE-CONTROLLED GREENHOUSE WITH PLANT GROWTH MONITORING SYSTEM USING RASPBERRY PI VIA ANDROID IOT APPLICATION.*

Boyd, C. E., & Tucker, C. S. (1998). *Turbidity and Appearance of Water*. *Pond Aquaculture Water Quality Management*, 374-393.

Bundu Power Co. (2018). *Solar Panels* .

Circuit Rocks. (2015). *Lora RFM95 Shield 915MHz.*

de Castro-Silva, T., dos Santos, L. D., da Silva, L. C., Michelato, M., Furuya, V., & Furuya, W. (2015). *Length-weight Relationship and Prediction Equations of body composition for Growing-Finishing Cage-farmed Nile Tilapia.*

DFRobot. (2015).

DFRobot. (2018). *Raspberry Pi 3 Model B.*

Elec Design Works. (2015). *Atlas Scientific Dissolved Oxygen Test Kit.*

Environment Health. (2016). *Oxidation-Reduction Potential.*

Fisheries Policy and Economics Division. (2008).

Food and Agriculture Organization. (2018). *Aquaculture* .

Food and Nutrition Research Institute. (2008). *Philippine Nutrition and Facts 2008.*

Grant, W. (2016). *Learning Python for Raspberry Pi.*

Griffith, E. (2015). *What is Cloud Computing?* PC Mag.

Guerrero. (2013).

Harper, C., & Wolf, J. C. (n.d.). *Morphologic Effects of the Stress Response in Fish.*

Ibrahim. (2018). *Fish Image Segmentation Using Salp Swarm Algorithm.*

(2016). *Is tilapia Brazil's secret weapon in aquaculture growth strategy?* www.feednavigator.com.

Junk Mail Classifieds . (2017). *Action Camera 4K Ultra HD* .

Keith, J. (2015). *Electro Schematics. Relay Driver.*

LoRa-Aliiance.org. (2018). *LoRaWAN*\.

MakerLab Electronics. (2015). *DC-DC Buck Converter LM2596S.*

MakerLab Electronics. (2016).

MakerLab Electronics. (2016). *Analog Turbidity Sensor Module Gravity Series* .

MakerLab Electronics. (2016). *DS18B20 Temperature Sensor.*

MakerLab Electronics. (2016). *Raspberry Pi 3 Model B.*

Mammano, R. (2001). *Switching Power Supply Topology Voltage Mode vs Current Mode. Elektron Journal-South African Institute of Electrical Engineers* , 25-27.

Maxim Integrated . (2015). *DS18B20.*

Mifflin, H. (2016). *The Free Dictionary.*

Mordvinstev, A., & Revision, A. (2013). *Image Segmentation with Watershed Algorithm* . *OpenCV-Python Tutorials.*

New Energy Co. (2015). *Green Goods 4U.*

PC Mag. (2016). *Cloud.*

PNG Icons. (2017).

Popma, T., & Masser, M. (1999). *Tilapia: Life History and Biology.*

Ranger, S. (2018). *The Internet of Things Explained* .

Ricker, W. (1975). *Computation and Interpretation of Biological Statistics of Fish Populations. 185-198.*

Rosebrock, A. (2016). *Measuring Size of Objects in an Image with OpenCV.*

Rossana. (2016).

Seitz, P. (2015). *Action Camera Market to See Growth Through 2019.* Investors Business Daily.

Shevechenko. (2017). Fish Detection for Species Recognition.

Solar Direct . (2016). *Solar Electric Photovoltaic Modules* .

Sunpower UK. (2013). What is an Inverter?

Techopedia. (2018). *Web Developmet.*

The Pi Hut. (2018). *Gravity: Analog ORP Sensor Meter For Arduino.*

Towers, L. (2015, JUNE 15). *How to Maintain Good Water Quality on Your Tilapia.*

Wade. (2003).

Winsense Co. (2013).

Yegulalp, s. (2017). *What is Python? Everything You Need to Know.* InfoWorld.

Zach. (2015). *SJCAM SJ4000 Wifi Action Camera Review* . pevly.com.

APPENDIX A

Source Codes

(Source code for Weight Prediction)

```
import time
import cv2
import numpy as np
import pandas as pd
import json
import argparse
from stereovision.calibration import StereoCalibrator
from stereovision.calibration import StereoCalibration
from datetime import datetime
import os
import tensorflow as tf
import sys
from stereovision.calibration import StereoCalibrator
from stereovision.calibration import StereoCalibration
from sklearn.preprocessing import normalize
from datetime import datetime
from matplotlib import pyplot as plt
import imutils
import MySQLdb
from collections import namedtuple
from array import *
import statistics
from statistics import mode
from collections import Counter
import csv

# This is needed since the notebook is stored in the object_detection folder.
sys.path.append("..")

# Import utilites
from utils import label_map_util
from utils import visualization_utils as vis_util
import configurations as config

# Depth map default preset
SWS = 5
PFS = 5
PFC = 29
MDS = -30
NOD = 160
TTH = 100
UR = 20
SR = 14
SPWS = 100
kernel= np.ones((3,3),np.uint8)
```

```
# Data model for easy data inserts
Data = namedtuple("Data", "numfish length weight")
t_sub = 0
t_samp = 1000*60*60*3 # three hours in milliseconds, time.time() returns milliseconds.

def db_insert(data):
    """
    Insert data to remote database via tunnel

    :param data:
    :type data: Data
    :return:
    """
    assert type(data) == Data
    db = MySQLdb.connect(
        host='127.0.0.1',
        port=9999,
        user='root',
        passwd='DmcwesNf7MSW1c',
        db='teamlapia_database'
    )
    curs = db.cursor()
    # Single Insert
    curs.execute("""INSERT INTO `weight_prediction` (`no_of_fishes`, `average_length`,
`average_weight`)
                VALUES (%s, %s, %s);""", data)
    db.commit()
    db.close()

def mode_function2(arr):
    counter = Counter(arr)
    _,val = counter.most_common(1)[0]
    return [x for x,y in counter.items() if y == val]

def coords_mouse_disp(event,x,y,flags,param):
    # print("test")
    # print("target: %s\tevent: %s" % (cv2.EVENT_MBUTTONDBLCLK, event))
    # print("dims: %s %s %s" % np.shape(disp))
    if event == 7:
        print (x,y,disp[y,x],filt_color[y,x])
        average=0

        for u in range (-1,2):
            for v in range (-1,2):
                average += disp[y+u,x+v]
```

```
        #print(average)
        Distance= -526.08*average**(3) + 671.21*average**(2) -341.66*average + 74.907
        Distance= np.around(Distance,decimals=2)
        print('Distance: '+ str(Distance)+' cm')

# Mouseclick callback
# wb=Workbook()
# ws=wb.active

def remove_garbage(im_in):
    im_out = im_in.copy()
    nb_components, output, stats, centroids = cv2.connectedComponentsWithStats(
        im_out, connectivity=8)
    sizes = stats[1:, -1]
    nb_components = nb_components - 1

    min_size = max(sizes) - 1
    for i in range(0, nb_components):
        if sizes[i] < min_size:
            im_out[output == i + 1] = 0

    return im_out

# Camera settimgs
cam_width = 640
cam_height = 480

# Final image capture settings
scale_ratio = 0.5

# Camera resolution height must be dividable by 16, and width by 32
cam_width = int((cam_width+31)/32)*32
cam_height = int((cam_height+15)/16)*16
print ("Used camera resolution: "+str(cam_width)+" x "+str(cam_height))

# Buffer for captured image settings
img_width = int (cam_width * scale_ratio)
img_height = int (cam_height * scale_ratio)
capture = np.zeros((img_height, img_width, 4), dtype=np.uint8)
print ("Scaled image resolution: "+str(img_width)+" x "+str(img_height))

# Initialize the camera
#camera = PiCamera(stereo_mode='side-by-side',stereo_decimate=False)
cam1 = cv2.VideoCapture(0)
cam1.set(cv2.CAP_PROP_FRAME_HEIGHT, 240)
cam1.set(cv2.CAP_PROP_FRAME_WIDTH, 320)
cam2 = cv2.VideoCapture(1)
```

```
cam2.set(cv2.CAP_PROP_FRAME_HEIGHT, 240)
cam2.set(cv2.CAP_PROP_FRAME_WIDTH, 320)

fgbg = cv2.createBackgroundSubtractorMOG2()

# Implementing calibration data
print('Read calibration data and rectifying stereo pair...')
calibration = StereoCalibration(input_folder='calib_result')

disparity = np.zeros((img_width, img_height), np.uint8)
sbm = cv2.StereoBM_create(numDisparities=0, blockSize=21)
min_disp = 2
num_disp = 130-min_disp

# WLS FILTER Parameters
lmbda = 80000
sigma = 1.8
visual_multiplier = 1.0

wls_filter = cv2.ximgproc.createDisparityWLSFilter(matcher_left=sbm)
wls_filter.setLambda(lmbda)
wls_filter.setSigmaColor(sigma)

def stereo_depth_map(rectified_pair):
    dmLeft = rectified_pair[0]
    dmRight = rectified_pair[1]
    #grayL= cv2.cvtColor(dmLeft, cv2.COLOR_BGR2GRAY)
    disparity = sbm.compute(dmLeft, dmRight)
    dispR= np.int16(disparity)
    local_max = disparity.max()
    local_min = disparity.min()
    disparity_grayscale = (disparity-local_min)*(65535.0/(local_max-local_min))
    # Using the WLS filter
    filteredImg= wls_filter.filter(disparity,dmLeft,None,dispR)
    filteredImg = cv2.normalize(src=filteredImg, dst=filteredImg, beta=0, alpha=255,
norm_type=cv2.NORM_MINMAX)
    filteredImg = np.uint8(filteredImg)
    disparity_fixtype = cv2.convertScaleAbs(disparity_grayscale, alpha=(255.0/65535.0))
    # time.sleep(2)
    closing= cv2.morphologyEx(disparity_fixtype ,cv2.MORPH_CLOSE, kernel)
    disparity_color = cv2.applyColorMap(closing, cv2.COLORMAP_JET)
    filt_Color= cv2.applyColorMap(filteredImg,cv2.COLORMAP_JET)
    # cv2.imshow("Image", disparity_color)
    cv2.setMouseCallback("Filtered Color Depth", coords_mouse_disp, disparity_color)
    # print()
    # cv2.setMouseCallback("Test", test_cb, disparity_color)
```

```
    return (disparity_color, filt_Color)

def load_map_settings( fName ):
    global SWS, PFS, PFC, MDS, NOD, TTH, UR, SR, SPWS, loading_settings
    print('Loading parameters from file...')
    f=open(fName, 'r')
    data = json.load(f)
    SWS=data['SADWindowSize']
    PFS=data['preFilterSize']
    PFC=data['preFilterCap']
    MDS=data['minDisparity']
    NOD=data['numberOfDisparities']
    TTH=data['textureThreshold']
    UR=data['uniquenessRatio']
    SR=data['speckleRange']
    SPWS=data['speckleWindowSize']
    #sbm.setSADWindowSize(SWS)
    sbm.setPreFilterType(1)
    sbm.setPreFilterSize(PFS)
    sbm.setPreFilterCap(PFC)
    sbm.setMinDisparity(MDS)
    sbm.setNumDisparities(NOD)
    sbm.setTextureThreshold(TTH)
    sbm.setUniquenessRatio(UR)
    sbm.setSpeckleRange(SR)
    sbm.setSpeckleWindowSize(SPWS)
    f.close()
    print ('Parameters loaded from file '+fName)

(sess, image_tensor, detection_boxes, detection_scores, detection_classes,
    num_detections, label_map, categories, category_index) = config.detect_object()
load_map_settings ("3dmap_set.txt")

#Object detection
arr = []
# capture frames from the camera
while (True):
    ret1, frame1 = cam1.read()
    ret2, frame2 = cam2.read()

    #resize frame dimensions
    frame1 = cv2.resize(frame1, (320, 240))
    frame2 = cv2.resize(frame2, (320, 240))
    frame3 = frame1.copy()
    frame4 = frame2.copy()
    hh, ww = frame1.shape[:2]
    frame1 = frame1[0:hh-15, 0:ww]
```

```
    frame2 = frame2[0:hh-15, 0:ww]
    frame_expanded = np.expand_dims(frame1, axis=0)
    t1 = datetime.now()
    ave=0
    #apply background subtractor

    fgmask = fgbg.apply(frame1)

    #pair_img = cv2.cvtColor (frame, cv2.COLOR_BGR2GRAY)
    imgLeft = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
    imgRight = cv2.cvtColor(frame2, cv2.COLOR_BGR2GRAY)
    rectified_pair = calibration.rectify((imgLeft, imgRight))
    disparity_color, filt_color = stereo_depth_map(rectified_pair)
    # print(np.shape(filt_color))
    disp = cv2.cvtColor(disparity_color, cv2.COLOR_BGR2GRAY)
    # print(np.shape(disp))
    disp= ((disp.astype(np.float32)/ 16)-min_disp)/num_disp # Calculation allowing us to
have 0 for the most distant object able to detect

    # Perform the actual detection by running the model with the image as input
    (boxes, scores, classes, num) = sess.run(
        [detection_boxes, detection_scores, detection_classes, num_detections],
        feed_dict={image_tensor: frame_expanded})

    # print(boxes)

    # Draw the results of the detection (aka 'visulaize the results')
    vis_util.visualize_boxes_and_labels_on_image_array(
        frame1,
        np.squeeze(boxes),
        np.squeeze(classes).astype(np.int32),
        np.squeeze(scores),
        category_index,
        use_normalized_coordinates=True,
        line_thickness=1,
        min_score_thresh=0.99,
        skip_scores=True,
        skip_labels=True,)

# print(frame.shape)
    height, width = frame1.shape[:2]
    for i, box in enumerate(np.squeeze(boxes)):
        if(np.squeeze(scores)[i] > 0.99):
            (ymain, xmain, ymaix, xmaix) = (int(box[0]*height), int(box[1]*width),
int(box[2]*height), int(box[3]*width))
            if ymain-15 < 0:
                ymin = 0
```

```
            dif_ymin = ymain
        else:
            ymin = ymain - 15
            dif_ymin = 15
            if xmain-15 < 0:
                xmin = 0
                dif_xmin = xmain
            else:
                xmin = xmain - 15
                dif_xmin = 15
                if ymaix+15 > hh:
                    ymax = hh
                    dif_ymax = hh - ymaix
                else:
                    ymax = ymaix + 15
                    dif_ymax = 15
                    if xmaix+15 > ww:
                        xmax = ww
                        dif_xmax = ww - xmaix
                    else:
                        xmax = xmaix + 15
                        dif_xmax = 15

                        #crop and resize
                        # print(xmin, xmax, ymin, ymax)
                        color_crop = frame3[ymin:ymax, xmin:xmax].copy()
                        fgbg_crop = fgmask[ymin:ymax, xmin:xmax].copy()
                        h, w = color_crop.shape[:2]
                        color_crop = cv2.resize(color_crop, (w*3, h*3))
                        color = color_crop.copy()
                        fgbg_crop = cv2.resize(fgbg_crop, (w*3, h*3))

                        dst = cv2.fastNlMeansDenoisingColored(color_crop.copy(), None, 10,
10, 7, 15)

                        hsv = cv2.cvtColor(dst, cv2.COLOR_BGR2HSV)
                        lower_blue = np.array([100,198,0])
                        upper_blue = np.array([140,255,255])
                        mask = cv2.inRange(hsv, lower_blue, upper_blue)
                        _, mask = cv2.threshold(mask,0,255,cv2.THRESH_BINARY_INV |
cv2.THRESH_OTSU)
                        # wisebit = cv2.bitwise_and(color_crop, color_crop, mask= mask)
                        # wow = cv2.bitwise_and(color_crop, color_crop, mask= fgbg_crop)
                        # wisebit = cv2.cvtColor(wisebit, cv2.COLOR_BGR2GRAY)
                        # wow = cv2.cvtColor(wow, cv2.COLOR_BGR2GRAY)
                        # magic = cv2.bitwise_or(wisebit, wow)
```

```
            # (thresh, magic) = cv2.threshold(wisebit, 0, 255,
            #                   cv2.THRESH_BINARY | cv2.THRESH_OTSU)

            magic = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel,
iterations=3)
            wisebit = cv2.bitwise_and(color_crop, color_crop, mask= mask)
            wisebit = cv2.cvtColor(wisebit, cv2.COLOR_BGR2GRAY)
            wisebit_x2 = cv2.GaussianBlur(wisebit, (5, 5), 0)

            contours,                    hierarchy                    =
cv2.findContours(magic,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
            cnt = contours[0]

            if len(contours) != 0:
              for i in range(len(contours)):
                if len(contours[i]) >= 5:
                  # cv2.drawContours(magic ,contours,-1,(150,10,255),3)
                  (x,y),(ma,MA),angle = cv2.fitEllipse(contours[0])
                  x_frame = int(x/3)-dif_xmin
                  y_frame = int(y/3)-dif_ymin
                  MA_frame = int(MA/6)
                  ma_frame = int(ma/6)
                  major_axis = MA/3
                  # #feed the parsed parameters into cv2.ellipse
                  if MA_frame > 30:
                    #                    cv2.ellipse(frame1,(x_frame+xmain
,y_frame+ymain),(ma_frame, MA_frame),angle,0,360,(0,255,0))
                    #       cv2.line(frame1,       (x_frame-MA_frame+xmain
,y_frame+ymain), (x_frame+MA_frame+xmain ,y_frame+ymain) ,(0,255,0),1)

                    # Apply Hough transform on the blurred image.
                    detected_circles = cv2.HoughCircles(wisebit,
                            cv2.HOUGH_GRADIENT, 1, 75, param1 = 50,
                          param2 = 30, minRadius = 5, maxRadius = 20)

                    # Draw circles that are detected.
                    if detected_circles is not None:

                      # Convert the circle parameters a, b and r to integers.
                      detected_circles = np.uint16(np.around(detected_circles))

                      for pt in detected_circles[0, :]:
                        aa, bb, r = pt[0], pt[1], pt[2]

                        cv2.circle(wisebit, (aa, bb), r, (0, 255, 0), 2)
                        # Draw a small circle (of radius 1) to show the center.
                        cv2.circle(wisebit, (aa, bb), 1, (0, 0, 255), 3)
```

```
aa = int((aa/3)-dif_ymin)
bb = int((bb/3)-dif_xmin)
cv2.circle(disparity_color, (xmain+aa, ymain+bb), 1, (255,
255, 255), 3)

cv2.circle(frame1, (xmain+aa, ymain+bb), 1, (0, 255, 0), 3)

average = 0

for u in range (-1,2):
    for v in range (-1,2):
        average += (disp[bb+ymain+u,aa+xmain+v])
        z = average
        D = int(592.05*z**(3) - 539.82*z**(2) + 193.77*z -
17.793)

        ppcm = (0.0017*z**(2) - 0.1601*z + 10.18)
        actual = (major_axis/ppcm)
        weight = -0.0019*actual**(3) + 2.1305*actual**(2) -
63.591*actual + 573.19

        label = 'Weight = ' + str('%.2f' % weight) + ' g : L=' +
str('%.2f' % actual) + ' cm'

        label2 = 'Length=' + str('%.2f' % actual) + ' cm'

if D > 0:
        # if angle > 86 and angle < 100:
    print(label)
    weight_r = int(round(weight/1.0)*1.0)
    arr.append(weight_r)
    cv2.putText(frame1,
                label,
                (int(xmain), int(ymain+10)),
                cv2.FONT_HERSHEY_PLAIN,
                .8,
                (0, 0, 0),
                1,
                cv2.LINE_AA)

    DATA INSERT
    numfish = 5

    if time.time() - t_sub >= t_samp:
         try:
             valid_data = mode_
             # send valid data
             db_insert(valid_data)
             # record time of successful data transfer
             t_sub = time.time()
```

```
                        except Exception as e:
                            print(e)

                    myFile = open('weight.csv', 'w')
                    with myFile:
                        writer = csv.writer(myFile)
                        writer.writerow(['Weight'])
                        writer.writerow(arr)

                    if len(arr) == 87:
                        print(mode_function2(arr))
                        arr.clear()

                cv2.imshow('hsv', mask)
                cv2.imshow('Morph', magic)
                cv2.imshow('Mask', wisebit)


    cv2.imshow('Camera Feed',frame1)
    cv2.imshow('Background Subtractor', fgmask)
    cv2.imshow('StereoVision', disparity_color)


    # Press 'q' to quit
    if cv2.waitKey(1) == ord('q'):
        quit()

# Clean up
cv2.destroyAllWindows()
```

(Source code for Weigh Prediction)

```
//////////////////////////// T E M P E R A T U R E ////////////////////////////////
#include <OneWire.h>
#include <DallasTemperature.h>

#define ONE_WIRE_BUS 2           // Data wire is plugged into pin 2 on the Arduino
OneWire oneWire(ONE_WIRE_BUS);   // Setup a oneWire instance to communicate with
any OneWire devices (not just Maxim/Dallas temperature ICs
DallasTemperature sensors(&oneWire);
```

```
/////////////////////////// O R P /////////////////////////////////

#define VOLTAGE 5.00    //system voltage
#define OFFSET 0        //zero drift voltage
#define LED 13         //operating instructions
#define ArrayLenth  40    //times of collection
#define orpPin 2          //orp meter output,connect to Arduino controller ADC pin

double orpValue;
int orpArray[ArrayLenth];
int orpArrayIndex=0;
double avergearray(int* arr, int number){
   int i;
  int max,min;
  double avg;
  long amount=0;
  if(number<=0){
    printf("Error number for the array to avraging!/n");
    return 0;
  }
  if(number<5){   //less than 5, calculated directly statistics
    for(i=0;i<number;i++){
      amount+=arr[i];
    }
    avg = amount/number;
    return avg;
  }else{
    if(arr[0]<arr[1]){
      min = arr[0];max=arr[1];
    }
    else{
      min=arr[1];max=arr[0];
    }
    for(i=2;i<number;i++){
      if(arr[i]<min){
        amount+=min;        //arr<min
        min=arr[i];
      }else {
        if(arr[i]>max){
          amount+=max;    //arr>max
          max=arr[i];
        }else{
          amount+=arr[i]; //min<=arr<=max
        }
      }//if
    }//for
    avg = (double)amount/(number-2);
```

```
  }//if
  return avg;
}
////////////////////////////////////////E C///////////////////////////////////////////////////////////
#include "DFRobot_EC.h"
#include <EEPROM.h>
#define EC_PIN A4
float voltage,ecValue, temperature;
DFRobot_EC ec;

/////////////////////////////////////////D O//////////////////////////////////////////////////////////
#include <avr/pgmspace.h>
#include <EEPROM.h>

#define DoSensorPin  A4    //dissolved oxygen sensor analog output pin to arduino mainboard
#define VREF 5000    //for arduino uno, the ADC reference is the AVCC, that is 5000mV(TYP)
float doValue;    //current dissolved oxygen value, unit; mg/L

#define EEPROM_write(address, p) {int i = 0; byte *pp = (byte*)&(p);for(; i < sizeof(p); i++) EEPROM.write(address+i, pp[i]);}
#define EEPROM_read(address, p)  {int i = 0; byte *pp = (byte*)&(p);for(; i < sizeof(p); i++) pp[i]=EEPROM.read(address+i);}

#define ReceivedBufferLength 20
char receivedBuffer[ReceivedBufferLength+1];    // store the serial command
byte receivedBufferIndex = 0;

#define SCOUNT  30           // sum of sample point
int analogBuffer[SCOUNT];    //store the analog value in the array, readed from ADC
int analogBufferTemp[SCOUNT];
int analogBufferIndex = 0,copyIndex = 0;

#define SaturationDoVoltageAddress 12          //the address of the Saturation Oxygen voltage stored in the EEPROM
#define SaturationDoTemperatureAddress 16      //the address of the Saturation Oxygen temperature stored in the EEPROM
float SaturationDoVoltage,SaturationDoTemperature;
float averageVoltage;

const float SaturationValueTab[41] PROGMEM = {      //saturation dissolved oxygen concentrations at various temperatures
14.46, 14.22, 13.82, 13.44, 13.09,
12.74, 12.42, 12.11, 11.81, 11.53,
11.26, 11.01, 10.77, 10.53, 10.30,
10.08, 9.86,  9.66,  9.46,  9.27,
```

```
9.08,  8.90,  8.73,  8.57,  8.41,
8.25,  8.11,  7.96,  7.82,  7.69,
7.56,  7.43,  7.30,  7.18,  7.07,
6.95,  6.84,  6.73,  6.63,  6.53,
6.41,
};

///////////////////////////////////////P H/////////////////////////////////////////////////////////
#define SensorPin 0          //pH meter Analog output to Arduino Analog Input 0
#define turbidity 1          //turbidity Analog output to Arduino Analog Input 1
unsigned long int avgValue;  //Store the average value of the sensor feedback
float b;
int buf[10],temp;
int phLOW = 10;
int phHIGH = 31;
int upStream = 8;
int downStream = 9;
int tempLOW = 34;
int tempHIGH = 35;
bool turb_Reset;
bool orp_Reset;
bool ec_Reset;
bool do_Reset;
bool ph_High;
bool ph_Low;
bool temp_High;
bool temp_Low;
bool waterFill;


void setup(){
  Serial.begin(9600);     // start serial port
  sensors.begin();
  Serial.println("Ready");    //Test the serial monitor
  pinMode(phLOW,OUTPUT);
  pinMode(phHIGH,OUTPUT);
  pinMode(upStream,OUTPUT);
  pinMode(downStream,OUTPUT);
  pinMode(tempLOW,OUTPUT);
  pinMode(tempHIGH,OUTPUT);
  pinMode(DoSensorPin,INPUT);
  readDoCharacteristicValues();
}
void loop(){
   void(* resetFunc) (void) = 5; //declare reset function @ address 0
  /////////////////////////////////////D O///////////////////////////////////////////
static unsigned long analogSampleTimepoint = millis();
```

```
  if(millis()-analogSampleTimepoint > 30U)      //every 30 milliseconds,read the analog
value from the ADC
  {
    analogSampleTimepoint = millis();
    analogBuffer[analogBufferIndex] = analogRead(DoSensorPin);   //read the analog value
and store into the buffer
    analogBufferIndex++;
    if(analogBufferIndex == SCOUNT)
        analogBufferIndex = 0;
  }

  static unsigned long tempSampleTimepoint = millis();
  if(millis()-tempSampleTimepoint > 500U)  // every 500 milliseconds, read the temperature
  {
     tempSampleTimepoint = millis();
     temperature = sensors.getTempCByIndex(0);  // add your temperature codes here to read
the temperature, unit:^C
  }

  static unsigned long printTimepoint = millis();
  if(millis()-printTimepoint > 1000U)
  {
     printTimepoint = millis();
     for(copyIndex=0;copyIndex<SCOUNT;copyIndex++)
     {
       analogBufferTemp[copyIndex]= analogBuffer[copyIndex];
       Serial.print(analogBuffer[copyIndex]);
  }
     averageVoltage = getMedianNum(analogBufferTemp,SCOUNT) * (float)VREF /
1024.0; // read the value more stable by the median filtering algorithm
     doValue = pgm_read_float_near( &SaturationValueTab[0] +
(int)(SaturationDoTemperature+0.5) ) * averageVoltage / SaturationDoVoltage;  //calculate
the do value, doValue = Voltage / SaturationDoVoltage * SaturationDoValue(with
temperature compensation)
     doValue = (voltage * (SaturationDoTemperature+0.5)) / SaturationDoVoltage;
  }
  if(serialDataAvailable() > 0)
  {
     byte modeIndex = uartParse();  //parse the uart command received
     doCalibration(modeIndex);     // If the correct calibration command is received, the
calibration function should be called.
   }
 /////////////////////////////////////PH SENSOR/////////////////////////////////////////////////////////
 delay(400);
 for(int i=0;i<10;i++)      //Get 10 sample value from the sensor for smooth the value
 {
   buf[i]=analogRead(SensorPin);
```

```
    delay(10);
  }
  for(int i=0;i<9;i++)        //sort the analog from small to large
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
        temp=buf[i];
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }
  avgValue=0;
  for(int i=2;i<8;i++)                      //take the average value of 6 center sample
    avgValue+=buf[i];
  float phValue=(float)avgValue*5.0/1024/6; //convert the analog into millivolt
  phValue=3.5*phValue;                      //convert the millivolt into pH value
  ////////////////////////////////////TURBIDITY//////////////////////////////////////////////////////////
  delay(200);
  int sensorValue = analogRead(A1); // read the input on analog pin 2:
  float voltage1 = sensorValue * (5.0 / 1024.0); // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):
  ///////////////////////////////////////ORP/////////////////////////////////////////////////////////////
  delay(600);{
  static unsigned long orpTimer=millis();   //analog sampling interval
  static unsigned long printTime=millis();
  if(millis() >= orpTimer)
    orpTimer=millis()+20;
    orpArray[orpArrayIndex++]=analogRead(orpPin);    //read an analog value every 20ms
    if (orpArrayIndex==ArrayLenth) {
      orpArrayIndex=0;
    }
    orpValue=(((30*(double)VOLTAGE*1000)-(75*avergearray(orpArray, ArrayLenth)*VOLTAGE*1000/1024))/75-OFFSET)/10;   //convert the analog value to orp according the circuit
  }
  ////////////////////////////////////TEMPERATURE////////////////////////////////////////////////
  delay(300);
  sensors.requestTemperatures();

  ////////////////////////////////////E C//////////////////////////////////////////
  delay(500);
  {
  voltage = analogRead(EC_PIN)/1024.0*5000;  // read the voltage
```

```
  temperature = sensors.getTempCByIndex(0);  // read your temperature sensor to execute
temperature compensation
  ecValue =   (ec.readEC(voltage,temperature)*0.64);   // convert voltage to EC with
temperature compensation
  }

  //PRINT
  delay(1000);
  Serial.print("pH: ");
  Serial.print(phValue,2);
  if(phValue > 9){
    ph_High = true;
    Serial.println(" | HIGH");
  }
    if(phValue < 6){
      ph_Low = true;
      Serial.println(" | LOW");
    }
      else if((phValue >6) && (phValue < 9)){
        Serial.println(" | NORMAL");
  }
  Serial.print("Turbidity: ");
  Serial.print(voltage1);
  if(voltage1 >5){
    turb_Reset = true;
    Serial.println(" | HIGH");
  }
    if(voltage1 < 0.3){
      turb_Reset = true;
      Serial.println(" | LOW");
    }
      else if((voltage1 >0.3) && (voltage < 1)){
        turb_Reset = false;
        Serial.println(" | NORMAL");
      }
  Serial.print("ORP: ");
  Serial.print((int)orpValue);
  Serial.print("mV");
   if(orpValue > 250){
    orp_Reset = true;
    Serial.println(" | HIGH");
  }
    if(orpValue < 150){
      orp_Reset = true;
      Serial.println(" | LOW");
    }
      else if((orpValue >150) && (orpValue < 250)){
```

```
      orp_Reset = false;
      Serial.println(" | NORMAL");
     }
  Serial.print("Temperature: ");
  Serial.print(sensors.getTempCByIndex(0));
  if(sensors.getTempCByIndex(0) > 29){
    temp_High = true;
    Serial.println(" | HIGH");
  }
  if(sensors.getTempCByIndex(0) < 27){
    temp_Low = true;
    Serial.println(" | LOW");
  }
  else if((sensors.getTempCByIndex(0) > 27) && (sensors.getTempCByIndex(0) < 29)){
    Serial.println(" | NORMAL");
  }
  Serial.print("Electric Conductivity: ");
  Serial.print(ecValue,2);
  Serial.print("ppt");
  if(ecValue > 20){
    ec_Reset = true;
    Serial.println(" | HIGH");
  }
  else if(ecValue <= 20){
    ec_Reset = false;
    Serial.println(" | NORMAL");
  }
  Serial.print("Dissolved Oxygen: ");
  Serial.print(doValue,2);
  Serial.print("mg/L");
  if(doValue < 4){
    do_Reset = true;
    Serial.println(" | LOW");
  }
  else if(doValue >= 4){
    do_Reset = false;
    Serial.println(" | NORMAL");
  }

  Serial.println("-----------------------------------------------------------------------------------------
--------------");
  Serial.println("Correction: Water Replacement");
  Serial.println("-----------------------------------------------------------------------------------------
--------------");
  delay(100);
```

```
  if((turb_Reset = true) || (orp_Reset = true) || (ec_Reset = true) || (do_Reset = true)){
//draining
    waterFill = true;
    digitalWrite(downStream, LOW);
    digitalWrite(upStream, HIGH);
    delay(60000);
  }

  //fill
  digitalWrite(downStream, HIGH);
  if(ph_High = true){
    digitalWrite(phHIGH, LOW);
  }
  if(ph_Low = true){
    digitalWrite(phLOW, LOW);
  }
  if(temp_High = true){
    digitalWrite(tempHIGH, LOW);
  }
  if(temp_Low = true){
    digitalWrite(tempLOW, LOW);
  }
  if(waterFill = true){
    digitalWrite(upStream, LOW);
}
  delay(60000);
  //reset and extended draining time
  digitalWrite(phLOW, HIGH);
  digitalWrite(phHIGH, HIGH);
  digitalWrite(tempLOW, HIGH);
  digitalWrite(tempHIGH, HIGH);
  digitalWrite(upStream, HIGH);
  waterFill = false;
  resetFunc();  //call reset
}

boolean serialDataAvailable(void)
{
  char receivedChar;
  static unsigned long receivedTimeOut = millis();
  while ( Serial.available() > 0 )
  {
    if (millis() - receivedTimeOut > 500U)
    {
      receivedBufferIndex = 0;
      memset(receivedBuffer,0,(ReceivedBufferLength+1));
    }
```

```
    receivedTimeOut = millis();
    receivedChar = Serial.read();
    if (receivedChar == '\n' || receivedBufferIndex == ReceivedBufferLength)
    {
    receivedBufferIndex = 0;
    strupr(receivedBuffer);
    return true;
    }else{
      receivedBuffer[receivedBufferIndex] = receivedChar;
      receivedBufferIndex++;
    }
  }
  return false;
}

byte uartParse()
{
    byte modeIndex = 0;
    if(strstr(receivedBuffer, "CALIBRATION") != NULL)
      modeIndex = 1;
    else if(strstr(receivedBuffer, "EXIT") != NULL)
      modeIndex = 3;
    else if(strstr(receivedBuffer, "SATCAL") != NULL)
      modeIndex = 2;
    return modeIndex;
}

void doCalibration(byte mode)
{
    char *receivedBufferPtr;
    static boolean doCalibrationFinishFlag = 0,enterCalibrationFlag = 0;
    float voltageValueStore;
    switch(mode)
    {
     case 0:
     if(enterCalibrationFlag)
       Serial.println(F("Command Error"));
     break;

     case 1:
     enterCalibrationFlag = 1;
     doCalibrationFinishFlag = 0;
     Serial.println();
     Serial.println(F(">>>Enter Calibration Mode<<<"));
     Serial.println(F(">>>Please put the probe into the saturation oxygen water! <<<"));
     Serial.println();
     break;
```

```
    case 2:
     if(enterCalibrationFlag)
     {
        Serial.println();
        Serial.println(F(">>>Saturation Calibration Finish!<<<"));
        Serial.println();
        EEPROM_write(SaturationDoVoltageAddress, averageVoltage);
        EEPROM_write(SaturationDoTemperatureAddress, temperature);
        SaturationDoVoltage = averageVoltage;
        SaturationDoTemperature = temperature;
        doCalibrationFinishFlag = 1;
     }
     break;

       case 3:
       if(enterCalibrationFlag)
       {
          Serial.println();
          if(doCalibrationFinishFlag)
             Serial.print(F(">>>Calibration Successful"));
          else
            Serial.print(F(">>>Calibration Failed"));
          Serial.println(F(",Exit Calibration Mode<<<"));
          Serial.println();
          doCalibrationFinishFlag = 0;
          enterCalibrationFlag = 0;
       }
       break;
    }
}

int getMedianNum(int bArray[], int iFilterLen)
{
     int bTab[iFilterLen];
     for (byte i = 0; i<iFilterLen; i++)
     {
     bTab[i] = bArray[i];
     }
     int i, j, bTemp;
     for (j = 0; j < iFilterLen - 1; j++)
     {
     for (i = 0; i < iFilterLen - j - 1; i++)
         {
       if (bTab[i] > bTab[i + 1])
           {
       bTemp = bTab[i];
```

```
        bTab[i] = bTab[i + 1];
      bTab[i + 1] = bTemp;
       }
    }
    }
    if ((iFilterLen & 1) > 0)
  bTemp = bTab[(iFilterLen - 1) / 2];
    else
  bTemp = (bTab[iFilterLen / 2] + bTab[iFilterLen / 2 - 1]) / 2;
    return bTemp;
}

void readDoCharacteristicValues(void)
{
  EEPROM_read(SaturationDoVoltageAddress, SaturationDoVoltage);
  EEPROM_read(SaturationDoTemperatureAddress, SaturationDoTemperature);
  if(EEPROM.read(SaturationDoVoltageAddress)==0xFF                    &&
EEPROM.read(SaturationDoVoltageAddress+1)==0xFF                       &&
EEPROM.read(SaturationDoVoltageAddress+2)==0xFF                       &&
EEPROM.read(SaturationDoVoltageAddress+3)==0xFF)
  {
    SaturationDoVoltage = 1127.6;   //default voltage:1127.6mv
    EEPROM_write(SaturationDoVoltageAddress, SaturationDoVoltage);
  }
  if(EEPROM.read(SaturationDoTemperatureAddress)==0xFF                &&
EEPROM.read(SaturationDoTemperatureAddress+1)==0xFF                   &&
EEPROM.read(SaturationDoTemperatureAddress+2)==0xFF                   &&
EEPROM.read(SaturationDoTemperatureAddress+3)==0xFF)
  {
    SaturationDoTemperature = 25.0;   //default temperature is 25^C
    EEPROM_write(SaturationDoTemperatureAddress, SaturationDoTemperature);
  }
}
```

(Web Application Source Codes)

HTML/PHP index code

```
<!DOCTYPE html>
<html lang="en">

<head>
   <meta charset="utf-8">
   <meta http-equiv="X-UA-Compatible" content="IE=edge">
   <meta name="viewport" content="width=device-width, initial-scale=1">
   <meta name="description" content="">
   <meta name="author" content="">

   <title>Team|Lapia</title>
   <link rel="icon" type="image/ico" href="img/headress.png" />

   <!-- Bootstrap Core CSS -->
   <link href="vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">

   <!-- Fonts -->
   <link href="vendor/font-awesome/css/font-awesome.min.css" rel="stylesheet"
type="text/css">
   <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700"
rel="stylesheet" type="text/css">
   <link href='https://fonts.googleapis.com/css?family=Kaushan+Script' rel='stylesheet'
type='text/css'>
   <link
href='https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic'
rel='stylesheet' type='text/css'>
   <link href='https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700'
rel='stylesheet' type='text/css'>

   <!-- CSS -->
   <link href="css/agency.css" rel="stylesheet">

</head>

<body id="page-top" class="index">

   <!-- Navigation -->
   <nav id="mainNav" class="navbar navbar-default navbar-custom navbar-fixed-top">
      <div>
            <a class="navbar-brand page-scroll" href="#page-top" style=" top:0px;
padding-left:150px; padding-top:0px; position:fixed">
                  <img src="img/Headeerr.png" alt="Header"
style="width:220px; height:170px; position:relative; padding-bottom:15px; margin:0px;">
```

```
                        </a>
            </div>

        <div class="container">
            <div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
                <ul class="nav navbar-nav navbar-right">
                    <li class="hidden">
                        <a href="#page-top"></a>
                    </li>
                    <li>
                        <a class="page-scroll" href="#about">About</a>
                    </li>
                    <li>
                        <a class="page-scroll" href="#portfolio">Sensors</a>
                    </li>
                    <li>
                        <a class="page-scroll" href="#prediction">Weight Prediction</a>
                    </li>
                    <li>
                        <a class="page-scroll" href="#team">Our Group</a>
                    </li>
                    <li>
                        <a class="page-scroll" href="#contact">Contact Us</a>
                    </li>
                </ul>
            </div>
        </div>
    </nav>

    <!-- Header -->
    <header>
        <div id="particles-js"></div>
    </header>
    <section id="divider">• • •</section>

    <!-- Information Section -->
    <section id="about" style="background-color: lightblue;">
        <div class="container">
            <div class="row">
                <div class="col-lg-12 text-center">
                    <h1 class="loader">TOUCH ME NOT!</h1>
                    <h3 class="section-subheading text-muted">A Study about Fish Growth and
Water Quality Monitoring System.

                    </h3>
                </div>
            </div>
```

```
          <div class="row text-center">
            <div class="col-md-4">
              <span class="fa-stack fa-4x">
                <i class="fa fa-square fa-stack-2x text-primary" style="color: black"></i>
                <i class="fa fa-camera fa-stack-1x fa-inverse"></i>
              </span>
              <h4 class="service-heading">Weight Prediction</h4>
              <p class="text-muted">Use of camera module to capture images for predicting
weight through image processing.</p>
            </div>
            <div class="col-md-4">
              <span class="fa-stack fa-4x">
                <i class="fa fa-square fa-stack-2x text-primary" style="color: black"></i>
                <i class="fa fa-eye fa-stack-1x fa-inverse"></i>
              </span>
              <h4 class="service-heading">Water Quality Monitoring</h4>
              <p class="text-muted">Use of six different sensors to monitor the current level
of water parameters in the fish tank. </p>
            </div>
            <div class="col-md-4">
              <span class="fa-stack fa-4x">
                <i class="fa fa-square fa-stack-2x text-primary" style="color: black"></i>
                <i class="fa fa-rss fa-stack-1x fa-inverse"></i>
              </span>
              <h4 class="service-heading">Data Consolidation</h4>
              <p class="text-muted">Development of webapp to display informations to the
end-users.</p>
            </div>
          </div>
        </div>
      </section>

      <!-- Sensors Section -->
      <section id="portfolio" class="bg-light-gray" >
        <div class="container">
          <div class="row">
            <div class="col-lg-12 text-center">
              <h2 class="section-heading">WATER PARAMETER SENSORS</h2>
              <h3 class="section-subheading text-muted"></h3>
            </div>
          </div>
          <div class="row">
            <div class="col-md-4 col-sm-6 portfolio-item">
              <a href="#ORP" class="portfolio-link" data-toggle="modal">
                <div class="portfolio-hover">
                  <div class="portfolio-hover-content">
                    <img src="img/portfolio/sensor.png" class="img-responsive" alt="">
```

```
            </div>
        </div>
        <img src="img/portfolio/ORP.png" class="img-responsive" alt="">
    </a>
    <div class="portfolio-caption">
        <h4>ORP</h4>
        <p class="text-muted">Oxidation Reduction Potential</p>
    </div>
</div>
<div class="col-md-4 col-sm-6 portfolio-item">
    <a href="#DO" class="portfolio-link" data-toggle="modal">
        <div class="portfolio-hover">
            <div class="portfolio-hover-content">
                <img src="img/portfolio/sensor.png" class="img-responsive" alt="">
            </div>
        </div>
        <img src="img/portfolio/DO.png" class="img-responsive" alt="">
    </a>
    <div class="portfolio-caption">
        <h4>DO</h4>
        <p class="text-muted">Dissolved Oxygen</p>
    </div>
</div>
<div class="col-md-4 col-sm-6 portfolio-item">
    <a href="#pH" class="portfolio-link" data-toggle="modal">
        <div class="portfolio-hover">
            <div class="portfolio-hover-content">
                <img src="img/portfolio/sensor.png" class="img-responsive" alt="">
            </div>
        </div>
        <img src="img/portfolio/pH.png" class="img-responsive" alt="">
    </a>
    <div class="portfolio-caption">
        <h4>pH</h4>
        <p class="text-muted">Potential Hydrogen</p>
    </div>
</div>
<div class="col-md-4 col-sm-6 portfolio-item">
    <a href="#Turb" class="portfolio-link" data-toggle="modal">
        <div class="portfolio-hover">
            <div class="portfolio-hover-content">
                <img src="img/portfolio/sensor.png" class="img-responsive" alt="">
            </div>
        </div>
        <img src="img/portfolio/Turb.png" class="img-responsive" alt="">
    </a>
    <div class="portfolio-caption">
```

```html
                <h4>NTU</h4>
                <p class="text-muted">Turbidity</p>
            </div>
        </div>
        <div class="col-md-4 col-sm-6 portfolio-item">
            <a href="#Na" class="portfolio-link" data-toggle="modal">
                <div class="portfolio-hover">
                    <div class="portfolio-hover-content">
                        <img src="img/portfolio/sensor.png" class="img-responsive" alt="">
                    </div>
                </div>
                <img src="img/portfolio/Na.png" class="img-responsive" alt="">
            </a>
            <div class="portfolio-caption">
                <h4>Na</h4>
                <p class="text-muted">Salinity</p>
            </div>
        </div>
        <div class="col-md-4 col-sm-6 portfolio-item">
            <a href="#Temp" class="portfolio-link" data-toggle="modal">
                <div class="portfolio-hover">
                    <div class="portfolio-hover-content">
                        <img src="img/portfolio/sensor.png" class="img-responsive" alt="">
                    </div>
                </div>
                <img src="img/portfolio/Temp.png" class="img-responsive" alt="">
            </a>
            <div class="portfolio-caption">
                <h4>°C</h4>
                <p class="text-muted">Water Temperature</p>
            </div>
        </div>
    </div>
</div>
</section>

<!-- Weight Prediction Section -->
<section id="prediction" style="background-color: #0f1c1f">
    <div class="column3">
        <div class="row">
            <div >
                <div >
                    <img class="timeline-image2" src="img/about/Picture1.png" alt="">
                </div>
            </div>
        </div>
        <div class="row">
```

```
          <div class="col-lg-12 text-center">
            <h2 class="section-heading2">WEIGHT PREDICTION</h2>
            <h3 class="section-subheading text-muted"></h3>
          </div>
          <div class="col-lg-12">
            <ul class="timeline">
              <li>
                <div class="timeline-image">
                  <img class="img-circle img-responsive" src="img/about/1.png" alt=""
                    href="#weightpredict" data-toggle="modal" style="">
                </div>
              </li>
            </ul>
          </div>
        </div>
        <div class="row">
          <div >
            <div >
              <img class="timeline-image3" src="img/about/Picture2.png" alt="">
            </div>
          </div>
        </div>
      </div>
    </section>

    <section id="divider">• • •</section>

    <!-- Team Section -->
    <section id="team" class="bg-light-grayish">
      <div class="container">
        <div class="row">
          <div class="col-lg-12 text-center">
            <h2 class="section-heading2">OUR RESEARCH TEAM</h2>
          </div>
        </div>
      </div>
      <div class="slidershow middle">
        <div class="slides">
          <input type="radio" name="r" id="r1" checked>
          <input type="radio" name="r" id="r2">
          <input type="radio" name="r" id="r3">
          <input type="radio" name="r" id="r4">
          <input type="radio" name="r" id="r5">

          <div class="slide s1">
           <img src="img/head1.jpg" alt="">
          </div>
```

```
        <div class="slide">
          <img src="img/head2.jpg" alt="">
        </div>
        <div class="slide">
          <img src="img/head3.jpg" alt="">
        </div>
        <div class="slide">
          <img src="img/head4.jpg" alt="">
        </div>
        <div class="slide">
          <img src="img/head5.jpg" alt="">
        </div>
      </div>
      <div class="navigation">
        <label for="r1" class="bar"></label>
        <label for="r2" class="bar"></label>
        <label for="r3" class="bar"></label>
        <label for="r4" class="bar"></label>
        <label for="r5" class="bar"></label>
      </div>
    </div>
    </div>
      <div class="column">
        <div class="flip-card">
        <div class="flip-card-inner">
          <div class="flip-card-front">
            <img src="img/team/5.jpg" alt="Avatar"
style="width:100%;height:100%;">
          </div>
        <div class="flip-card-back">
         <h4>Gian Sobrevilla</h4>
        <ul class="list-inline social-buttons">
          <li><a href="https://www.facebook.com/giandearroz"><i class="fa fa-
facebook"></i></a></li>
        </ul>
        </div>
        </div>
      </div>
      <div class="flip-card">
        <div class="flip-card-inner">
          <div class="flip-card-front">
            <img src="img/team/4.jpg" alt="Avatar"
style="width:100%;height:100%;">
          </div>
        <div class="flip-card-back">
         <h4>Luigi Salvacion</h4>
        <ul class="list-inline social-buttons">
```

```
                <li><a href="https://www.facebook.com/luigidosado"><i class="fa fa-
facebook"></i></a></li>
            </ul>
            </div>
            </div>
        </div>
        <div class="flip-card">
            <div class="flip-card-inner">
                <div class="flip-card-front">
                    <img src="img/team/2.jpg" alt="Avatar"
style="width:100%;height:100%;">
                </div>
            <div class="flip-card-back">
             <h4>Jatt Icamina</h4>
            <ul class="list-inline social-buttons">
                <li><a href="https://www.facebook.com/icaminajatt"><i class="fa fa-
facebook"></i></a></li>
            </ul>
            </div>
            </div>
        </div>
         <div class="flip-card">
            <div class="flip-card-inner">
                <div class="flip-card-front">
                    <img src="img/team/7.jpg" alt="Avatar"
style="width:100%;height:100%;">
                </div>
            <div class="flip-card-back">
             <h4>Lean Tolentino</h4>
             <h4>-Adviser-</h4>
            <ul class="list-inline social-buttons">
                <li><a
href="https://scholar.google.com/citations?user=p0i2zeMAAAAJ&hl=en"><i class="fa fa-
google"></i></a></li>
            </ul>
            </div>
            </div>
        </div>
        <div class="flip-card">
            <div class="flip-card-inner">
                <div class="flip-card-front">
                    <img src="img/team/1.jpg" alt="Avatar"
style="width:100%;height:100%;">
                </div>
            <div class="flip-card-back">
             <h4>Celline De Pedro</h4>
            <ul class="list-inline social-buttons">
```

```
                <li><a    href="https://www.facebook.com/cellinederella"><i    class="fa    fa-
facebook"></i></a></li>
            </ul>
            </div>
            </div>
        </div>
        <div class="flip-card">
            <div class="flip-card-inner">
                <div class="flip-card-front">
                    <img                         src="img/team/3.jpg"                         alt="Avatar"
style="width:100%;height:100%;">
                </div>
            <div class="flip-card-back">
             <h4>JB Navarro</h4>
            <ul class="list-inline social-buttons">
                <li><a     href="https://www.facebook.com/jebs.orravan"><i     class="fa     fa-
facebook"></i></a></li>
            </ul>
            </div>
            </div>
        </div>
        <div class="flip-card">
            <div class="flip-card-inner">
                <div class="flip-card-front">
                    <img                         src="img/team/6.jpg"                         alt="Avatar"
style="width:100%;height:100%;">
                </div>
            <div class="flip-card-back">
             <h4>Apolo Villanueva</h4>
            <ul class="list-inline social-buttons">
                <li><a     href="https://www.facebook.com/OlopapolO"><i     class="fa     fa-
facebook"></i></a></li>
            </ul>
            </div>
            </div>
        </div>
        </div>
    </section>

    <section id="divider">• • •</section>

    <!-- Contact Section -->
    <footer id="contact">
        <div class="container">
            <div class="row">
                <div class="col-lg-12 text-center">
                    <h2 class="section-heading0">CONTACT US</h2>
```

```
                <h3 class="section-subheading text-muted"></h3>
            </div>
        </div>
         <div class="row">
            <div class="col-md-6 col-sm-6">
                <a href="https://www.tup.edu.ph">
                    <img src="img/logos/TUP.png" class="img-responsive img-centered" alt=""
style="float: left">
                </a>
            </div>
            <div class="col-md-6 col-sm-6">
                <a href="https://www.facebook.com/TUPCOE/">
                    <img src="img/logos/CUE.png" class="img-responsive img-centered"
alt="" style="float: right">
                </a>
            </div>
        </div>
    </div>
</footer>

<!-- Footer Section -->
<footer>
    <div class="container">
        <div class="column3">
            <div >
                <span class="copyright">teamlapia2020@gmail.com</span>
            </div>
            <div class="share-button">
                <span class="fa fa-user"></span>
                <a href="https://www.facebook.com/TeamLapia-234676423890272/"><i
class="fa fa-facebook-f"></i></a>
                <a href="https://www.thethingsnetwork.org/u/TeamLapia"><i class="fa fa-
cloud"></i></a>
            </div>
            <div >
                <span class="copyright">Copyright &copy; TeamLapia 2020</span>
            </div>
        </div>
    </div>
</footer>

<!-- Portfolio Modal 1 -->
<div class="portfolio-modal modal fade" id="ORP" tabindex="-1" role="dialog" aria-
hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="close-modal" data-dismiss="modal">
```

```
                <div class="lr">
                    <div class="rl">
                    </div>
                </div>
            </div>
            <div class="container">
                <div class="row">
                    <div class="col-lg-8 col-lg-offset-2">
                        <div class="modal-body">
                            <h2>ORP</h2>
                            <p class="item-intro text-muted">Oxidation Reduction Potential</p>
                            <img class="img-responsive img-centered" src="img/portfolio/aquarium.png" alt="">
                            <p><STRONG>ORP or Oxidation-Reduction Potential</STRONG> sensor works by measuring the amount of dissolve oxygen present in the water. With higher ORP level, water has greater ability to destroy foreign contaminants including microbes and other carbon-based contaminants.  Various levels of ORP corresponds to different applications. The ideal value for the level of ORP for aquaculture ranges from 150mV to 250 mV.</p>
                            <p>ORP Analog Meter V1.0 measures the ability of the ability of oxidation and reduction of aqueous solutions that follows a logarithmic curve and requires adjustments in terms of calibration. ORP detection is a reliable method of measuring water quality for water quality testing, marine science, hydroponic gardening and aquaculture. It can measure from -2000mV to 2000mV with a response time less than 20secs and has an accuracy of ±10mV (25°C) at a suitable temperature range of 5°C to 70°C.  </p>

                            <div>
                            <canvas id="Apols"></canvas>
                            </div>
                            <div class="timeline-heading">
                                <h4 class="subheading">Date/Time: <span id="human" style="color: #417b85"></span></h4>
                                <h4 class="subheading">Current Level (mV): <span id="dog" style="color: #417b85"></span></h4>
                                <h4 class="subheading">Status: <span id="dogstat" style="color: #417b85"></span></h4>
                            </div>
                            <br><br>

                            <button type="button" class="btn btn-primary" data-dismiss="modal">View Data History</button>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```
    </div>

    <!-- Portfolio Modal 2-->
    <div class="portfolio-modal modal fade" id="DO" tabindex="-1" role="dialog" aria-hidden="true">
        <div class="modal-dialog">
            <div class="modal-content">
                <div class="close-modal" data-dismiss="modal">
                    <div class="lr">
                        <div class="rl">
                        </div>
                    </div>
                </div>
                <div class="container">
                    <div class="row">
                        <div class="col-lg-8 col-lg-offset-2">
                            <div class="modal-body">
                                <h2>DO</h2>
                                <p class="item-intro text-muted">Dissolved Oxygen</p>
                                <img class="img-responsive img-centered" src="img/portfolio/aquarium.png" alt="">
                                <p><strong>DO or dissolved oxygen</strong> is essential for respiration and decomposition. Dissolved Oxygen comes from atmospheric oxygen and photosynthesis from aquatic plants. Reduced DO can be detected if the fish don’t feed well or come to the surface to breathe. Tilapia can tolerate up to less than .3mg/L that may be fatal for other farmed fish, but a level of 4mg/L of DO is the most preferable. </p>

                                <p>Atlas Scientific Dissolved Oxygen Test Kit has everything you need in terms of accurate full-range DO readings in an environmental monitoring like hydroponics and fish keeping. It has an operating voltage of 3.3V to 5V with full-range DO readings from 0.01 to +35.99 mg/L with an accuracy reading of ±0.2. It is compatible to any microprocessors that supports UART or I2C protocol.</p>

                                <div>
                                <canvas id="Carlo"></canvas>
                                </div>
                                <div class="timeline-heading">
                                        <h4 class="subheading">Date/Time: <span id="human2" style="color: #417b85"></span></h4>
                                        <h4 class="subheading">Current Level (mg/L): <span id="cat" style="color: #417b85"></span></h4>
                                        <h4 class="subheading">Status: <span id="catstat" style="color: #417b85"></span></h4>
                                </div>
                                <br><br>
```

```
                                         <button type="button" class="btn btn-
primary" data-dismiss="modal">View Data History</button>
                       </div>
                    </div>
                 </div>
              </div>
           </div>
        </div>
    </div>

    <!-- Portfolio Modal 3 -->
    <div class="portfolio-modal modal fade" id="pH" tabindex="-1" role="dialog" aria-
hidden="true">
        <div class="modal-dialog">
           <div class="modal-content">
              <div class="close-modal" data-dismiss="modal">
                 <div class="lr">
                    <div class="rl">
                    </div>
                 </div>
              </div>
              <div class="container">
                 <div class="row">
                    <div class="col-lg-8 col-lg-offset-2">
                       <div class="modal-body">
                          <h2>pH</h2>
                          <p class="item-intro text-muted">Potential hydrogen</p>
                          <img class="img-responsive img-centered"
src="img/portfolio/aquarium.png" alt="">
                          <p><strong>The pH range or acidity level</strong> of water should be
maintained between a pH level of 5 to 10, but a range of 6 to 9 can give Tilapia an optimal
growth. Soluble carbonate or any bicarbonate source can be used to balance the level of
acidity of the water as these are easy to obtain, extremely soluble and safe to handle.</p>

                          <p>Ion-Sensitive Field Effect Transistor or ISFET is a silicon-based
potentiometric field effect transistor sensor used to measure ion concentrations (H+ and
OH+ ) in a solution. Ion-sensitive membrane replaces metal gates as input.  The
measurement principle using ISFET is based on the flow of current between two
semiconductors namely the drain and the source. The sensitivity of ISFET is completely
controlled by the properties of electrolyte. ISFET can be integrated with MOSFET and other
standard transistors.</p>

                          <div>
                          <canvas id="Benjamin"></canvas>
                          </div>
                          <div class="timeline-heading">
```

```
                    <h4 class="subheading">Date/Time: <span id="human3"
style="color: #417b85"></span></h4>
                    <h4 class="subheading">Current Level: <span id="rat"
style="color: #417b85"></span></h4>
                    <h4 class="subheading">Status: <span id="ratstat" style="color:
#417b85"></span></h4>
                  </div>
                  <br><br>

                  <button type="button" class="btn btn-primary" data-
dismiss="modal">View Data History</button>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>

    <!-- Portfolio Modal 4 -->
    <div class="portfolio-modal modal fade" id="Turb" tabindex="-1" role="dialog" aria-
hidden="true">
      <div class="modal-dialog">
        <div class="modal-content">
          <div class="close-modal" data-dismiss="modal">
            <div class="lr">
              <div class="rl">
              </div>
            </div>
          </div>
          <div class="container">
            <div class="row">
              <div class="col-lg-8 col-lg-offset-2">
                <div class="modal-body">
                  <h2>NTU</h2>
                  <p class="item-intro text-muted">Turbidity</p>
                  <img class="img-responsive img-centered"
src="img/portfolio/aquarium.png" alt="">
                  <p><strong>Turbidity</strong> is described as the overall physical
appearance of water and it is often obvious to know if the turbidity level is high enough to
replace the water which is the only way to correct the said parameter. Though turbidity is
easy to predict, it is important to still monitor it since it is considered as one of the most
important parameter that is often disregarded. Presence of foreign substances such as foams
and scums provide clues to the characteristics of water quality. Clear water promotes light
penetration for underwater organisms.</p>
```

```
<p>Analog Turbidity Sensor Module detects particles present in the water by measuring the amount of light that pass through the water which is directly proportional to the amount of total suspended soils (TTS) in the water to identify the quality of the water. It has a response time less than 500ms with an insulation resistance of 100M (minutes). It provides a digital output of High and Low level signal where the threshold value can be adjusted using the potentiometer. </p>

                <div>
                <canvas id="James"></canvas>
                </div>
                <div class="timeline-heading">
                    <h4 class="subheading">Date/Time: <span id="human4" style="color: #417b85"></span></h4>
                    <h4 class="subheading">Current Level (NTU): <span id="pig" style="color: #417b85"></span></h4>
                    <h4 class="subheading">Status: <span id="pigstat" style="color: #417b85"></span></h4>
                </div>
                <br><br>

                <button type="button" class="btn btn-primary" data-dismiss="modal">View Data History</button>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

  <!-- Portfolio Modal 5 -->
  <div class="portfolio-modal modal fade" id="Na" tabindex="-1" role="dialog" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="close-modal" data-dismiss="modal">
          <div class="lr">
            <div class="rl">
            </div>
          </div>
        </div>
        <div class="container">
          <div class="row">
            <div class="col-lg-8 col-lg-offset-2">
              <div class="modal-body">
                <h2>Na</h2>
                <p class="item-intro text-muted">Salinity</p>
```

```
                        <img class="img-responsive img-centered" src="img/portfolio/aquarium.png" alt="">
                        <p><strong> Salinity</strong> and Turbidity comes together as for water replacement is the only way to correct both but we should also consider that Tilapia is one of the species that can tolerate high level of salinity even if they are considered freshwater fish. Tilapia grows well at a salinity level range of 15-20 ppt and when the water quality exceeds the required limit, considering other parameters, water replacement should take place.</p>

                        <div>
                        <canvas id="Jatt"></canvas>
                        </div>
                        <div class="timeline-heading">
                            <h4 class="subheading">Date/Time: <span id="human5" style="color: #417b85"></span></h4>
                            <h4 class="subheading">Current Level (ppt): <span id="goat" style="color: #417b85"></span></h4>
                            <h4 class="subheading">Status: <span id="goatstat" style="color: #417b85"></span></h4>
                        </div>
                        <br><br>

                        <button type="button" class="btn btn-primary" data-dismiss="modal">View Data History</button>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
</div>

<!-- Portfolio Modal 6 -->
<div class="portfolio-modal modal fade" id="Temp" tabindex="-1" role="dialog" aria-hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="close-modal" data-dismiss="modal">
                <div class="lr">
                    <div class="rl">
                    </div>
                </div>
            </div>
            <div class="container">
                <div class="row">
                    <div class="col-lg-8 col-lg-offset-2">
                        <div class="modal-body">
```

```
                <h2>°C</h2>
                <p class="item-intro text-muted">Temperature</p>
                <img class="img-responsive img-centered"
src="img/portfolio/aquarium.png" alt="">
                <p>Fish species are temperature-dependent. Fishes like tilapia, catfish
and bass are warm water species with a preferred temperature that ranges from 65 to 85°F
(18-29°C). The desired temperature for tilapia ranges from 81-85°F (27-29°C) for an
optimum growth. When the temperature ranges below 70°F (21°C), Tilapia growth
decreases and can stop reproduction. Temperatures below 50°F (10°C) is fatal to tilapia.
Stress and mortality from handling increases below 65°F during the harvesting of
Tilapia.</p>

                <p>DS18B20 is a digital thermometer ideal for underwater
applications.  It's pre-wired waterproof temperature sensor provides 9-bit to 12-bit Celsius
temperature measurements that communicates through a digital wire bus. It is used for
thermostatic controls and thermally sensitive systems that can withstand from -55°C to
125°C (-67°F to +257°F).</p>

                <div>
                <canvas id="Celline"></canvas>
                </div>
                <div class="timeline-heading">
                    <h4 class="subheading">Date/Time: <span id="human6"
style="color: #417b85"></span></h4>
                    <h4 class="subheading">Current Level (°C): <span id="bird"
style="color: #417b85"></span></h4>
                    <h4 class="subheading">Status: <span id="birdstat" style="color:
#417b85"></span></h4>
                </div>
                <br><br>
                <button type="button" class="btn btn-primary" data-
dismiss="modal">View Data History</button>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

  <!-- Portfolio Modal Weight Prediction -->
  <div class="portfolio-modal modal fade" id="weightpredict" tabindex="-1"
role="dialog" aria-hidden="true">
    <div class="modal-dialog">
      <div class="modal-content">
        <div class="close-modal" data-dismiss="modal">
          <div class="lr">
```

```
                    <div class="rl">
                    </div>
                </div>
            </div>
            <div class="container">
                <div class="row">
                    <div class="col-lg-8 col-lg-offset-2">
                        <div class="modal-body">
                            <h2>Weight Prediction</h2>
                            <img class="img-responsive img-centered" src="img/portfolio/aquariumwtcam.png" alt="">
                            <p><STRONG>Weight Prediction</STRONG> is an automatic predictive model that predicts the weight of a fish through the use of various image processing algorithms. In here, we get the estimated weight of all the fishes in the tank and then displays it in relation with the number of fishes present.</p>
                            <div class="timeline-panel">
                                    <div>
                                        <canvas id="meep"></canvas>
                                    </div>
                                <div class="timeline-heading">
                                    <h4 class="subheading">Date/Time: <span id="ab" style="color: #417b85"></span></h4>
                                    <h4 class="subheading">Current Fish Count: <span id="cd" style="color: #417b85"></span></h4>
                                    <h4 class="subheading">Conventional Setup Average Weight (g): <span id="ef" style="color: #417b85"></span></h4>
                                    <h4 class="subheading">Controlled Setup Average Weight (g): <span id="gh" style="color: #417b85"></span></h4>
                                </div>
                            </div>
                            <br><br>
                            <button type="button" class="btn btn-primary" data-dismiss="modal">View Data History</button>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>

<!-- jQuery -->
<script src="vendor/jquery/jquery.min.js"></script>

<!-- Bootstrap Core JavaScript -->
<script src="vendor/bootstrap/js/bootstrap.min.js"></script>
```

```
   <!-- Plugin JavaScript -->
   <script                                src="https://cdnjs.cloudflare.com/ajax/libs/jquery-easing/1.3/jquery.easing.min.js"                                integrity="sha384-mE6eXfrb8jxl0rzJDBRanYqgBxtJ6Unn4/1F7q4xRRyIw7Vdg9jP4ycT7x1iVsgb"
crossorigin="anonymous"></script>

   <!-- Theme JavaScript -->
   <script src="js/agency.min.js"></script>

   <!-- Particles JS -->
   <script src="js/particles.js"></script>
   <script src="js/app.js"></script>

   <!-- Charts Theme JS -->
   <script type ="text/javascript" src="chart_prediction.js"></script>
   <script type ="text/javascript" src="chart_sensor.js"></script>
   <script src="https://cdnjs.cloudflare.com/ajax/libs/Chart.js/2.8.0/Chart.min.js"></script>
   <script type ="text/javascript" src="js/jquery.min.js"></script>
   <script type = "text/javascript">
     $(document).ready(function(){
       setInterval(function(){
         var datetime = [];
         var averagelength = [];
         var averageweight = [];
         var nooffish = [];

         $.ajax({
          url : "latest_prediction.php",
          type : "GET",
          success : function(data){

           $.each(data, function(i, value){
            datetime.push(value.date);
            averagelength.push(value.average_length);
            averageweight.push(value.average_weight);
            nooffish.push(value.no_of_fishes);
           });
           document.getElementById("ab").innerHTML = datetime;
           document.getElementById("cd").innerHTML = nooffish;
           document.getElementById("ef").innerHTML = averagelength;
           document.getElementById("gh").innerHTML = averageweight;
           document.getElementById(".ab").innerHTML = datetime;
           document.getElementById(".cd").innerHTML = nooffish;
           document.getElementById(".ef").innerHTML = averagelength;
           document.getElementById(".gh").innerHTML = averageweight;

          },
```

```
          error : function(xhr, status, error){
            console.log(status + " in pushJsonData: " + error + " " + xhr)
          },
        dataType: 'json'
        });
      }, 5000);
    });
</script>
<script type = "text/javascript">
    $(document).ready(function(){
      setInterval(function(){
        var datetime2 = [];
        var d_o = [];
        var d_o_status = [];
        var te_mp = [];
        var te_mp_status = [];
        var p_h = [];
        var p_h_status = [];
        var or_p = [];
        var or_p_status = [];
        var n_a = [];
        var n_a_status = [];
        var tu_rb = [];
        var tu_rb_status = [];

        $.ajax({
          url : "latest_sensor.php",
          type : "GET",
          success : function(data){

            $.each(data, function(i, value){
              datetime2.push(value.date);
              d_o.push(value.do);
              d_o_status.push(value.do_status);
              te_mp.push(value.temp);
              te_mp_status.push(value.temp_status);
              p_h.push(value.ph);
              p_h_status.push(value.ph_status);
              or_p.push(value.orp);
              or_p_status.push(value.orp_status);
              n_a.push(value.na);
              n_a_status.push(value.na_status);
              tu_rb.push(value.turb);
              tu_rb_status.push(value.turb_status);
            });
            console.log(or_p);
            document.getElementById("human").innerHTML = datetime2;
```

```
            document.getElementById("human0").innerHTML = datetime2;
            document.getElementById("dog").innerHTML = or_p;
            document.getElementById("dogstat").innerHTML = or_p_status;
            document.getElementById("dogstat2").innerHTML = or_p_status;
            document.getElementById("human2").innerHTML = datetime2;
            document.getElementById("cat").innerHTML = d_o;
            document.getElementById("catstat").innerHTML = d_o_status;
            document.getElementById("catstat2").innerHTML = d_o_status;
            document.getElementById("human3").innerHTML = datetime2;
            document.getElementById("rat").innerHTML = p_h;
            document.getElementById("ratstat").innerHTML = p_h_status;
            document.getElementById("ratstat2").innerHTML = p_h_status;
            document.getElementById("human4").innerHTML = datetime2;
            document.getElementById("pig").innerHTML = tu_rb;
            document.getElementById("pigstat").innerHTML = tu_rb_status;
            document.getElementById("pigstat2").innerHTML = tu_rb_status;
            document.getElementById("human5").innerHTML = datetime2;
            document.getElementById("goat").innerHTML = n_a;
            document.getElementById("goatstat").innerHTML = n_a_status;
            document.getElementById("goatstat2").innerHTML = n_a_status;
            document.getElementById("human6").innerHTML = datetime2;
            document.getElementById("bird").innerHTML = te_mp;
            document.getElementById("birdstat").innerHTML = te_mp_status;
            document.getElementById("birdstat2").innerHTML = te_mp_status;

          },
          error : function(xhr, status, error){
            console.log(status + " in pushJsonData: " + error + " " + xhr)
          },
        dataType: 'json'
        });
      }, 5000);
    });
</script>
<script type="text/javascript">
    $(document).ready(function(){
      var arr = []; // List of users
     $(document).on('click', '.msg_head', function() {
      var chatbox = $(this).parents().attr("rel") ;
      $('[rel="'+chatbox+'"] .msg_wrap').slideToggle('fast');
      return false;
     });


     $(document).on('click', '.close', function() {
      var chatbox = $(this).parents().parents().attr("rel") ;
      $('[rel="'+chatbox+'"]').hide();
```

```
         arr.splice($.inArray(chatbox, arr), 1);
         displayChatBox();
         return false;
        });

        $(document).ready(function() {
         var userID = $(this).attr("class");
         arr.unshift(userID);
         chatPopup =  '<div class="msg_box" style="right:20px" rel="'+""+'">'+
           '<div class="msg_head">'+" &nbspTeamLapia - Status " +
           '<div class="close">'+"&#10006;"+'</div> </div>'+
           '<div class="msg_wrap"><div class="msg_body">'+
           '<div><h8>Water Quality</h8><br>'+
             '<h7><strong><span id="human0" style="color:
#4ba8b8"></span></strong></h7><br>'+
             '<h7>ORP: <span id="dogstat2" style="color: #4ba8b8"></span></h7><br>'+
             '<h7>DO: <span id="catstat2" style="color: #4ba8b8"></span></h7><br>'+
             '<h7>pH: <span id="ratstat2" style="color: #4ba8b8"></span></h7><br>'+
             '<h7>Turbidity: <span id="pigstat2" style="color:
#4ba8b8"></span></h7><br>'+
             '<h7>Salinity: <span id="goatstat2" style="color:
#4ba8b8"></span></h7><br>'+
             '<h7>Temperature: <span id="birdstat2" style="color:
#4ba8b8"></span></h7><br></div>'+
          '<div><h8>Fish Growth</h8><br>'+
             '<h7><strong><span id=".ab" style="color:
#4ba8b8"></span></strong></h7><br>'+
             '<h7>Number of Fish: <span id=".cd" style="color:
#4ba8b8"></span></h7><br>'+
             '<h7>Conventional Setup Average Weight(g): <span id=".ef" style="color:
#4ba8b8"></span></h7><br>'+
             '<h7>Controlled Setup Average Weight(g): <span id=".gh" style="color:
#4ba8b8"></span></h7><br>'+
           '</div></div></div></div>';
           $("body").append(  chatPopup  );
         displayChatBox();
        });
       });
    </script>
</body>
</html>
```

Core CSS codes

```
body, html {
   height: 100%;
}
body {
  overflow-x: hidden;
  font-family: "Roboto Slab", "Helvetica Neue", Helvetica, Arial, sans-serif;
}
* {
  cursor: url(http://ani.cursors-4u.net/cursors/cur-13/cur1162.ani),
  url(http://ani.cursors-4u.net/cursors/cur-13/cur1162.png),
  auto !important;
  margin: 0;
  padding: 0;
}
.text-muted {
  color: #777777;
}
.text-primary {
  color: #417b85;
}
p {
  font-size: 14px;
  line-height: 1.75;
}
p.large {
  font-size: 16px;
}
a,
a:hover,
a:focus,
a:active,
a.active {
  outline: none;
}
a {
  color: #417b85;
}
a:hover,
a:focus,
a:active,
a.active {
  color: #417b85;
}
h1,
h2,
h3,
h4,
h5,
h6 {
  font-family: "Montserrat", "Helvetica Neue", Helvetica, Arial, sans-serif;
  text-transform: none;
  font-weight: 700;
}
h7 {
  font-size: 15px;
}
h8 {
  font-size: 18px;
  color: white;
}
.img-centered {
  padding: 2px;

}
.bg-light-gray {
  background-image: url("headeer.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;

}
.bg-light-grayish {
  background-image: url("headeerrr.jpg");
  background-repeat: no-repeat;
  background-attachment: fixed;
  background-size: cover;

}
.bg-darkest-gray {
  background-color: #222222;
}
.btn-primary { color: white;
  background-color: #417B85;
  border-color: #417B85;
  font-family: "Montserrat", "Helvetica Neue", Helvetica, Arial, sans-serif;
  text-transform: uppercase;
  font-weight: 700;
```

```
}
.btn-primary:hover,
.btn-primary:focus,
.btn-primary:active,
.btn-primary.active,
.open .dropdown-toggle.btn-primary {
  color: white;
  background-color: #03233F;
  border-color: #03233F;
}
.btn-primary:active,
.btn-primary.active,
.open .dropdown-toggle.btn-primary {
  background-image: none;
}
.btn-primary.disabled,
.btn-primary[disabled],
fieldset[disabled] .btn-primary,
.btn-primary.disabled:hover,
.btn-primary[disabled]:hover,
fieldset[disabled] .btn-primary:hover,
.btn-primary.disabled:focus,
.btn-primary[disabled]:focus,
fieldset[disabled] .btn-primary:focus,
.btn-primary.disabled:active,
.btn-primary[disabled]:active,
fieldset[disabled] .btn-primary:active,
.btn-primary.disabled.active,
.btn-primary[disabled].active,
fieldset[disabled] .btn-primary.active {
  background-color: #417b85;
  border-color: #417b85;
}
.btn-primary .badge {
  color: #417b85;
  background-color: white;
}
.btn-xl {
  color: white;
  background-color: #417b85;
  border-color: #417b85;
  font-family: "Montserrat", "Helvetica Neue", Helvetica, Arial, sans-serif;
  text-transform: uppercase;
  font-weight: 700;
  border-radius: 3px;
  font-size: 18px;
  padding: 20px 40px;
}
.btn-xl:hover,
.btn-xl:focus,
.btn-xl:active,
.btn-xl.active,
.open .dropdown-toggle.btn-xl {
  color: white;
  background-color: #417b85;
  border-color: #417b85;
}
.btn-xl:active,
.btn-xl.active,
.open .dropdown-toggle.btn-xl {
  background-image: none;
}
.btn-xl.disabled,
.btn-xl[disabled],
fieldset[disabled] .btn-xl,
.btn-xl.disabled:hover,
.btn-xl[disabled]:hover,
fieldset[disabled] .btn-xl:hover,
.btn-xl.disabled:focus,
.btn-xl[disabled]:focus,
fieldset[disabled] .btn-xl:focus,
.btn-xl.disabled:active,
.btn-xl[disabled]:active,
fieldset[disabled] .btn-xl:active,
.btn-xl.disabled.active,
.btn-xl[disabled].active,
fieldset[disabled] .btn-xl.active {
  background-color: #417b85;
  border-color: #417b85;
}
.btn-xl .badge {
  color: #417b85;
  background-color: white;
}
.navbar-custom {
  background-color: #222222;
  border-color: transparent;
}
.navbar-custom .navbar-brand {
  color: lightblue;
  font-family: "Kaushan Script", "Helvetica Neue", Helvetica, Arial, cursive;
```

```
}
.navbar-custom .navbar-brand:hover,
.navbar-custom .navbar-brand:focus,
.navbar-custom .navbar-brand:active,
.navbar-custom .navbar-brand.active {
 color: lightblue;
}
.navbar-custom .navbar-collapse {
 border-color: rgba(255, 255, 255, 0.02);
}
.navbar-custom .navbar-toggle {
 background-color: lightblue;
 border-color: lightblue;
 font-family: "Montserrat", "Helvetica
Neue", Helvetica, Arial, sans-serif;
 text-transform: uppercase;
 color: white;
 font-size: 12px;
}
.navbar-custom .navbar-toggle:hover,
.navbar-custom .navbar-toggle:focus {
 background-color: lightblue;
}
.navbar-custom .nav li a {
 font-family: "Montserrat", "Helvetica
Neue", Helvetica, Arial, sans-serif;
 text-transform: uppercase;
 font-weight: 400;
 letter-spacing: 1px;
 color: white;
}
.navbar-custom .nav li a:hover,
.navbar-custom .nav li a:focus {
 color: lightblue;
 outline: none;
}
.navbar-custom .navbar-nav > .active > a
{
 border-radius: 0;
 color: white;
 background-color: #417B85;
}
.navbar-custom .navbar-nav > .active >
a:hover,
.navbar-custom .navbar-nav > .active >
a:focus {
 color: white;
 background-color: #417B85;
}
@media (min-width: 768px) {
 .navbar-custom {
  background-color: transparent;
  padding: 25px 0;
  -webkit-transition: padding 0.3s;
  -moz-transition: padding 0.3s;
  transition: padding 0.3s;
  border: none;
 }
 .navbar-custom .navbar-brand {
  font-size: 2em;
  -webkit-transition: all 0.3s;
  -moz-transition: all 0.3s;
  transition: all 0.3s;
 }
 .navbar-custom .navbar-nav > .active > a
{
  border-radius: 3px;
 }
}
@media (min-width: 768px) {
 .navbar-custom.affix {
  background-color: #1C2833;
  padding: 10px 0;
 }
 .navbar-custom.affix .navbar-brand {
  font-size: 1.5em;
 }
}
header {
 background-image: url('../img/header-
bg.jpg');
 background-repeat: no-repeat;
 background-attachment: fixed;
 background-position: center center;
 -webkit-background-size: cover;
 -moz-background-size: cover;
 background-size: cover;
 -o-background-size: cover;
 text-align: center;
 color: white;
 height:100%;
}
section {
 padding: 100px 0;
```

```
}
section h2.section-heading0 {
  font-size: 40px;
  color: white;
  margin-top: 2px;
  margin-bottom: 15px;
}
section h2.section-heading {
  font-size: 40px;
  color: white;
  margin-top: 0;
  margin-bottom: 15px;
}
section h2.section-heading2 {
  font-size: 40px;
  color: white;
  background-color: none;
  margin-top: 0;
  margin-bottom: 15px;
}
section h2.section-heading3 {
  font-size: 40px;
  color: #17202A;
  margin-top: 0;
  margin-bottom: 15px;
}
section h3.section-subheading {
  font-size: 16px;
  font-family: "Droid Serif", "Helvetica
Neue", Helvetica, Arial, sans-serif;
  text-transform: none;
  font-style: italic;
  font-weight: 400;
  margin-bottom: 75px;
}
@media (min-width: 768px) {
  section {
    padding: 150px 0;
  }
}
.service-heading {
  margin: 15px 0;
  text-transform: none;
}
#portfolio .portfolio-item {
  margin: 0 0 15px;
  right: 0;
}
#portfolio .portfolio-item .portfolio-link {
  display: block;
  position: relative;
  max-width: 400px;
  margin: 0 auto;
}
#portfolio .portfolio-item .portfolio-link
.portfolio-hover {
  background: #417B85 ;
  position: absolute;
  width: 100%;
  height: 100%;
  opacity: 0;
  transition: all ease 0.5s;
  -webkit-transition: all ease 0.5s;
  -moz-transition: all ease 0.5s;
}
#portfolio .portfolio-item .portfolio-link
.portfolio-hover:hover {
  opacity: 1;
}
#portfolio .portfolio-item .portfolio-link
.portfolio-hover .portfolio-hover-content
{
  position: absolute;
  width: 90px;
  height: 90px;
  font-size: 20px;
  text-align: center;
  top: 33%;
  left: 38%;
  margin-top: -12px;
  color: white;
}
#portfolio .portfolio-item .portfolio-link
.portfolio-hover .portfolio-hover-content i
{
  margin-top: -12px;
}
#portfolio .portfolio-item .portfolio-link
.portfolio-hover .portfolio-hover-content
h3,
#portfolio .portfolio-item .portfolio-link
.portfolio-hover .portfolio-hover-content
h4 {
  margin: 0;
```

```css
}
#portfolio .portfolio-item .portfolio-caption {
  max-width: 400px;
  margin: 0 auto;
  background-color: lightblue;
  text-align: center;
  padding: 25px;
  border-bottom-left-radius: 20px;
  border-bottom-right-radius: 20px;

}
#portfolio .portfolio-item .portfolio-caption h4 {
  text-transform: none;
  margin: 0;
}
#portfolio .portfolio-item .portfolio-caption p {
  font-family: "Droid Serif", "Helvetica Neue", Helvetica, Arial, sans-serif;
  font-style: italic;
  font-size: 16px;
  margin: 0;
}
#portfolio * {
  z-index: 2;
}
@media (min-width: 767px) {
  #portfolio .portfolio-item {
    margin: 0 0 30px;
  }
}
.timeline {
  list-style: none;
  padding: 0;
  position: relative;
}
.timeline > li {
  margin-bottom: 50px;
  position: relative;
  min-height: 50px;
}
.timeline > li .timeline-panel {
  width: 100%;
  float: right;
  padding: 0 20px 0 100px;
  position: relative;
  text-align: center;
}
.timeline > li .timeline-image {
  display: block;
  margin: 0 auto;
  margin-bottom: -90px;
  width: 300px;
  height: 300px;
  position: relative;
  background-color: #417b85;
  color: white;
  border-radius: 50%;
  border: 7px solid black;
  text-align: center;
  animation: animate 3s linear infinite;
}
.timeline-image2 {
  justify-self: left;
  display: block;
  width: 500px;
  height: 300px;
  position: relative;
  color: white;
  text-align: center;
  left: -20px;
  bottom: -228px;
  float: left;
}
.timeline-image3 {
  justify-self: right;
  display: block;
  width: 375px;
  height: 360px;
  position: relative;
  color: white;
  text-align: center;
  right: -20px;
  bottom: -168px;
  float: right;
}
@keyframes animate {
  0%  {
    box-shadow: 0 0 0 0 rgba(26,51,58,1), 0 0 0 0 rgba(26,51,58,1), 0 0 0 0 rgba(26,51,58,1);
  }
```

```
  40%  {
   box-shadow: 0 0 0 50px rgba(26,51,58,0), 0 0 0 0 rgba(26,51,58,1), 0 0 0 0 rgba(26,51,58,1);
  }
  80%  {
   box-shadow: 0 0 0 50px rgba(26,51,58,0), 0 0 0 30px rgba(26,51,58,0), 0 0 0 0 rgba(26,51,58,1);
  }
  100% {
   box-shadow: 0 0 0 0 rgba(26,51,58,0), 0 0 0 30px rgba(26,51,58,0), 0 0 0 20px rgba(26,51,58,0);
  }
}
.timeline > li .timeline-image h4 {
 font-size: 10px;
 margin-top: 12px;
 line-height: 14px;
}
.timeline .timeline-heading h4.subheading {
 text-transform: none;
 text-align: center;
}
.timeline .timeline-body > p,
.timeline .timeline-body > ul {
 margin-bottom: 0;
}
.team-member {
 text-align: center;
 background-color:#1C2833;
 margin-bottom: 50px;
 max-height: 800px;
 padding-top: 20px;
 padding-bottom: 20px;
}
.team-member img {
 margin: 0 auto;
 border: 7px solid white;
 padding-top: 5px;
 padding-bottom: 5px;

}
.team-member h4 {
 margin-top: 25px;
 margin-bottom: 0;
 text-transform: none;
 background-color: lightblue;
 padding-top: 5px;
 padding-bottom: 5px;

}
.team-member p {
 margin-top: 0;
 background-color: white;

}
aside.clients img {
 margin: 5px auto;
}
section#contact {
 background-color: #222222;
 background-position: center;
 background-repeat: no-repeat;
}
section#contact .section-heading {
 color: white;
}
section#contact .form-group {
 margin-bottom: 25px;
}
section#contact .form-group input,
section#contact .form-group textarea {
 padding: 20px;
}
section#contact .form-group input.form-control {
 height: auto;
}
section#contact .form-group textarea.form-control {
 height: 236px;
}
section#contact .form-control:focus {
 border-color: #222222;
 box-shadow: none;
}
section#contact ::-webkit-input-placeholder {
```

```
  font-family: "Montserrat", "Helvetica
Neue", Helvetica, Arial, sans-serif;
  text-transform: uppercase;
  font-weight: 700;
  color: #eeeeee;
}
section#contact :-moz-placeholder {
  /* Firefox 18- */
  font-family: "Montserrat", "Helvetica
Neue", Helvetica, Arial, sans-serif;
  text-transform: uppercase;
  font-weight: 700;
  color: #eeeeee;
}
section#contact ::-moz-placeholder {
  /* Firefox 19+ */
  font-family: "Montserrat", "Helvetica
Neue", Helvetica, Arial, sans-serif;
  text-transform: uppercase;
  font-weight: 700;
  color: #eeeeee;
}
section#contact :-ms-input-placeholder {
  font-family: "Montserrat", "Helvetica
Neue", Helvetica, Arial, sans-serif;
  text-transform: uppercase;
  font-weight: 700;
  color: #eeeeee;
}
section#contact .text-danger {
  color: #e74c3c;
}
footer {
  padding: 25px 0;
  text-align: center;
  background-color:#0f1c1f;
  color: #9ac4ce;
}
footer span.copyright {
  line-height: 40px;
  font-family: "Montserrat", "Helvetica
Neue", Helvetica, Arial, sans-serif;
  text-transform: uppercase;
  text-transform: none;
  justify-self: center;
  text-align: center;
}
footer ul.quicklinks {
  margin-bottom: 0;
  line-height: 40px;
  font-family: "Montserrat", "Helvetica
Neue", Helvetica, Arial, sans-serif;
  text-transform: uppercase;
  text-transform: none;
}
ul.social-buttons {
  margin-bottom: 0;
}
ul.social-buttons li a {
  display: block;
  background-color: #222222;
  height: 40px;
  width: 40px;
  border-radius: 100%;
  font-size: 20px;
  line-height: 40px;
  color: white;
  outline: none;
  -webkit-transition: all 0.3s;
  -moz-transition: all 0.3s;
  transition: all 0.3s;
}
ul.social-buttons li a:hover,
ul.social-buttons li a:focus,
ul.social-buttons li a:active {
  background-color: #417b85;
}
.btn:focus,
.btn:active,
.btn.active,
.btn:active:focus {
  outline: none;
}
.portfolio-modal .modal-dialog {
  margin: 0;
  height: 100%;
  width: auto;
}
.portfolio-modal .modal-content {
  border-radius: 0;
  background-clip: border-box;
  -webkit-box-shadow: none;
  box-shadow: none;
  border: none;
```

```css
  min-height: 100%;
  padding: 100px 0;
  text-align: center;
}
.portfolio-modal .modal-content h2 {
  margin-bottom: 15px;
  font-size: 3em;
}
.portfolio-modal .modal-content p {
  margin-bottom: 30px;
}
.portfolio-modal .modal-content p.item-intro {
  margin: 20px 0 30px;
  font-family: "Droid Serif", "Helvetica Neue", Helvetica, Arial, sans-serif;
  font-style: italic;
  font-size: 16px;
}
.portfolio-modal .modal-content ul.list-inline {
  margin-bottom: 30px;
  margin-top: 0;
}
.portfolio-modal .modal-content img {
  margin-bottom: 30px;
}
.portfolio-modal .close-modal {
  position: absolute;
  width: 75px;
  height: 75px;
  background-color: transparent;
  top: 25px;
  right: 25px;
  cursor: pointer;
}
.portfolio-modal .close-modal:hover {
  opacity: 0.3;
}
.portfolio-modal .close-modal .lr {
  height: 75px;
  width: 1px;
  margin-left: 35px;
  background-color: #222222;
  transform: rotate(45deg);
  -ms-transform: rotate(45deg);
  /* IE 9 */
  -webkit-transform: rotate(45deg);
  /* Safari and Chrome */
  z-index: 1051;
}
.portfolio-modal .close-modal .lr .rl {
  height: 75px;
  width: 1px;
  background-color: #222222;
  transform: rotate(90deg);
  -ms-transform: rotate(90deg);
  /* IE 9 */
  -webkit-transform: rotate(90deg);
  /* Safari and Chrome */
  z-index: 1052;
}
.portfolio-modal .modal-backdrop {
  opacity: 0;
  display: none;
}
::-moz-selection {
  text-shadow: none;
  background: #417b85;
}
::selection {
  text-shadow: none;
  background: #417b85;
}
img::selection {
  background: transparent;
}
img::-moz-selection {
  background: transparent;
}
body {
  webkit-tap-highlight-color: #417b85;
}
#divider {
  background-color:#1C2833;
  margin: 0px;
  background-attachment: relative;
  padding: 3px;
  text-align: center;
  color: white;
  font-size: 30px;
}
.flip-card {
  border-radius: 20%;
```

```
  background-color: transparent;
  width: 200px;
  height: 200px;
  perspective: 1000px;
  justify-self: center;
}
.flip-card-inner {
  position: relative;
  width: 100%;
  height: 100%;
  text-align: center;
  transition: transform 0.6s;
  transform-style: preserve-3d;
  box-shadow: 0 6px 10px 0
rgba(0,0,0,0.5);
}
.flip-card:hover .flip-card-inner {
  transform: rotateY(180deg);
}
.flip-card-front, .flip-card-back {
  position: absolute;
  width: 100%;
  height: 100%;
  backface-visibility: hidden;
}
.flip-card-front {
  background-color: #bbb;
  border-color: white;
  color: black;
}
.flip-card-back {
  padding-top: 50px;
  background-color: #00464b;
  color: white;
  transform: rotateY(180deg);
}
* {
  box-sizing: border-box;
}
.column{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr 1fr 1fr
1fr 1fr;
}
.column2{
  display: grid;
  grid-template-columns: 1fr;
}
.column3{
  display: grid;
  grid-template-columns: 1fr 1fr 1fr;
}
.row::after {
  content: "";
  clear: both;
  display: table;
}
.loader {
  margin-top: 0;
  margin-bottom: 15px;
}
section h1.loader {
  font-size: 5em;
  margin-top: 0;
  margin-bottom: 15px;
  color: rgba(26,51,58,.3);
  background-image: url(wave.png);
  background-repeat: repeat-x;
  -webkit-background-clip: text;
  animation: anime 10s linear infinite;
}
@keyframes anime {
  0%  {
    background-position: left 0px top 70px;
  }
  40% {
    background-position: left 400px top
5px;
  }
  80% {
    background-position: left 800px top
5px;
  }
  100%  {
    background-position: left 1200px top
70px;
  }
}
.chart-container {
      height: auto;
}

.slidershow{
  width: 975px;
```

```
  height: 565px;
  overflow: hidden;
  margin-top: 300px;
  margin-bottom: -250px;
}
.middle{
  position: relative;

  left: 50%;
  transform: translate(-50%,-50%);
}
.navigation{
  position: absolute;
  bottom: 20px;
  left: 50%;
  transform: translateX(-50%);
  display: flex;
}
.bar{
  width: 15px;
  height: 15px;
  margin: 6px;
  cursor: pointer;
  border: 2px solid white;
  transition: 0.4s;
  background-color: white;
  border-radius: 50%;
}
.bar:hover{
  background: none;
}
input[name="r"]{
    position: absolute;
    visibility: hidden;
}
.slides{
  width: 500%;
  height: 100%;
  display: flex;
}
.slide{
  width: 20%;
  transition: 0.6s;
}
.slide img{
  width: 100%;
  height: 100%;
}
#r1:checked ~ .s1{
  margin-left: 0;
}
#r2:checked ~ .s1{
  margin-left: -20%;
}
#r3:checked ~ .s1{
  margin-left: -40%;
}
#r4:checked ~ .s1{
  margin-left: -60%;
}
#r5:checked ~ .s1{
  margin-left: -80%;
}


.share-button{
  width: 150px;
  height: 50px;
  background: #1C2833;
  border-radius: 50px;
  display: flex;
  align-items: center;
  justify-content: center;
  justify-self: center;
  overflow: hidden;
  position: relative;
  cursor: pointer;
  transition: .3s linear;
}
.share-button:hover{
  transform: scale(1.1);
}
.share-button span{
  position: absolute;
  width: 100%;
  height: 100%;
  background: lightblue;
  color: #1C2833;
  text-align: center;
  align-items: center;
  line-height: 50px;
  z-index: 999;
  transition: .6s linear;
```

```
  border-radius: 50px;
  font-size: 30px;
}
.share-button:hover span{
  transform: translateX(-101%);
  transition-delay: .3s;
}
.share-button a{
  flex: 1;
  font-size: 26px;
  color: lightblue;
  text-align: center;
  transform: translateX(-100%);
  opacity: 0;
  transition: 0.3s linear;
}
.share-button:hover a{
  opacity: 1;
  transform: translateX(0);
}
.share-button a:nth-of-type(1){
  transition-delay: 1s;
}
.share-button a:nth-of-type(2){
  transition-delay: 0.8s;
}
.share-button a:nth-of-type(3){
  transition-delay: 0.6s;
}
.share-button a:nth-of-type(4){
  transition-delay: 0.4s;
}

.msg_box{
 position:fixed;
 bottom:0px;
}
.minimize{
 float:right;
 cursor:pointer;
 padding-right:5px;
}
 width:400px;
 background:white;
 border-radius:15px 15px 0px 0px;
 z-index: 1055;
 box-shadow: 0 6px 10px 0 rgba(0,0,0,0.8);
}
.msg_head{
 background:#1C2833;
 color:white;
 padding:8px;
 font-weight:normal;
 font-family:"Montserrat", "Helvetica Neue", Helvetica, Arial, sans-serif;
 font-size: 15px;
 cursor:pointer;
 border-radius:9px 9px 0px 0px;
 height: 40px;
}
.msg_body{
 background:#273847;
 height:200px;
 font-size:12px;
 padding:15px;
 overflow:auto;
 overflow-x: hidden;
 color: lightblue;
 display: grid;
 grid-template-columns: 1fr 1fr;
}
.close{
 float:right;
 cursor:auto;
 font-weight:normal;
 color:gray;
```

Graphs/Charts JavaScript Codes

Water monitoring:

```
$(document).ready(function(){
        setInterval(function(){
          var datetime = [];
          var d_o = [];
          var te_mp = [];
          var p_h = [];
          var or_p = [];
          var n_a = [];
          var tu_rb = [];

          var LineGraph1 = new Chart($('#Apols'), {
           type: 'line'
          });
          var LineGraph2 = new Chart($('#Carlo'), {
           type: 'line'
          });
          var LineGraph3 = new Chart($('#Benjamin'), {
           type: 'line'
          });
          var LineGraph4 = new Chart($('#James'), {
           type: 'line'
          });
          var LineGraph5 = new Chart($('#Jatt'), {
           type: 'line'
          });
          var LineGraph6 = new Chart($('#Celline'), {
           type: 'line'
          });
            $.ajax({
             url : "chart_sensor.php",
             type : "GET",
             success : function(data){

              $.each(data, function(i, value){
               datetime.push(value.date);
               d_o.push(value.do);
               te_mp.push(value.temp);
               p_h.push(value.ph);
               or_p.push(value.orp);
               n_a.push(value.na);
               tu_rb.push(value.turb);
              });
```

```
var bulbasaur = {
  labels: datetime,
  datasets: [
    {
      label: "dissolved oxygen (mg/L)",
      fill: false,
      lineTension: 0.1,
      backgroundColor: "rgba(47, 141, 161, 0.75)",
      borderColor: "rgba(47, 141, 161, 1)",
      pointHoverBackgroundColor: "rgba(47, 141, 161, 1)",
      pointHoverBorderColor: "rgba(47, 141, 161, 1)",
      data: d_o
    },
  ]
};
var charmander = {
  labels: datetime,
  datasets: [
    {
      label: "temperature (°C)",
      fill: false,
      lineTension: 0.1,
      backgroundColor: "rgba(47, 141, 161, 0.75)",
      borderColor: "rgba(47, 141, 161, 1)",
      pointHoverBackgroundColor: "rgba(47, 141, 161, 1)",
      pointHoverBorderColor: "rgba(47, 141, 161, 1)",
      data: te_mp
    },
  ]
};
var squirtle = {
  labels: datetime,
  datasets: [
    {
      label: "potential hydrogen",
      fill: false,
      lineTension: 0.1,
      backgroundColor: "rgba(47, 141, 161, 0.75)",
      borderColor: "rgba(47, 141, 161, 1)",
      pointHoverBackgroundColor: "rgba(47, 141, 161, 1)",
      pointHoverBorderColor: "rgba(47, 141, 161, 1)",
      data: p_h
    },
  ]
};
var venusaur = {
  labels: datetime,
```

```
  datasets: [
    {
      label: "oxidation reduction potential (mV)",
      fill: false,
      lineTension: 0.1,
      backgroundColor: "rgba(47, 141, 161, 0.75)",
      borderColor: "rgba(47, 141, 161, 1)",
      pointHoverBackgroundColor: "rgba(47, 141, 161, 1)",
      pointHoverBorderColor: "rgba(47, 141, 161, 1)",
      data: or_p
    },
  ]
};
var charizard = {
  labels: datetime,
  datasets: [
    {
      label: "salinity (ppt)",
      fill: false,
      lineTension: 0.1,
      backgroundColor: "rgba(47, 141, 161, 0.75)",
      borderColor: "rgba(47, 141, 161, 1)",
      pointHoverBackgroundColor: "rgba(47, 141, 161, 1)",
      pointHoverBorderColor: "rgba(47, 141, 161, 1)",
      data: n_a
    },
  ]
};
var blastoise = {
  labels: datetime,
  datasets: [
    {
      label: "turbidity (NTU)",
      fill: false,
      lineTension: 0.1,
      backgroundColor: "rgba(47, 141, 161, 0.75)",
      borderColor: "rgba(47, 141, 161, 1)",
      pointHoverBackgroundColor: "rgba(47, 141, 161, 1)",
      pointHoverBorderColor: "rgba(47, 141, 161, 1)",
      data: tu_rb
    },
  ]
};

var blurb1 = $("#Apols");
var blurb2 = $("#Carlo");
var blurb3 = $("#Benjamin");
```

```
            var blurb4 = $("#James");
            var blurb5 = $("#Jatt");
            var blurb6 = $("#Celline");

            var LineGraph1 = new Chart(blurb1, {
              type: 'line',
              data: venusaur
            });
            var LineGraph2 = new Chart(blurb2, {
              type: 'line',
              data: bulbasaur
            });
            var LineGraph3 = new Chart(blurb3, {
              type: 'line',
              data: squirtle
            });
            var LineGraph4 = new Chart(blurb4, {
              type: 'line',
              data: blastoise
            });
            var LineGraph5 = new Chart(blurb5, {
              type: 'line',
              data: charizard
            });
            var LineGraph6 = new Chart(blurb6, {
              type: 'line',
              data: charmander
            });
          },
          error : function(xhr, status, error) {
            console.log(status + " in pushJsonData: " + error + " " + xhr)
          },
        dataType: 'json'
        });
    }, 5000);
});
```

Weight Prediction:

```
$(document).ready(function(){
    setInterval(function(){
        var datetime = [];
        var averagelength = [];
        var averageweight = [];
        var LineGraph = new Chart($('#meep'), {
          type: 'line'
        });
            $.ajax({
              url : "chart_prediction.php",
              type : "GET",
              success : function(data){

                $.each(data, function(i, value){
                  datetime.push(value.date);
                  averagelength.push(value.average_length);
                  averageweight.push(value.average_weight);
                });

                var charizard = {
                  labels: datetime,
                  datasets: [
                    {
                      label: "Conventional Setup Average Weight (g)",
                      fill: false,
                      lineTension: 0.1,
                      backgroundColor: "rgba(47, 141, 161, 0.75)",
                      borderColor: "rgba(47, 141, 161, 1)",
                      pointHoverBackgroundColor: "rgba(47, 141, 161, 1)",
                      pointHoverBorderColor: "rgba(47, 141, 161, 1)",
                      data: averagelength
                    },
                    {
                      label: "Controlled Setup Average Weight (g)",
                      fill: false,
                      lineTension: 0.1,
                      backgroundColor: "rgba(61, 192, 220, 0.75)",
                      borderColor: "rgba(61, 192, 220, 1)",
                      pointHoverBackgroundColor: "rgba(61, 192, 220, 1)",
                      pointHoverBorderColor: "rgba(61, 192, 220, 1)",
                      data: averageweight
                    }
                  ]
                };
```

```
          var blurb = $("#meep");

          var LineGraph = new Chart(blurb, {
            type: 'line',
            data: charizard
          });
        },
        error : function(xhr, status, error) {
          console.log(status + " in pushJsonData: " + error + " " + xhr)
        },
      dataType: 'json'
      });
   }, 5000);
});
```

PHP Codes (Data acquisition from database)

Water Monitoring: Whole Set

```
<?php

header('Content-Type: application/json');

$db = mysqli_connect("localhost", "root", "", "teamlapia_database");

if (!$db) {
   echo "Error: Unable to connect to MySQL." . PHP_EOL;
   echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;
   echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;
   exit;
}

$query = "SELECT * FROM sensor_data";
$result = $db->query("$query");
$data = [];
foreach ($result as $row)
{
$data[] = $row;
}

$db->close();

echo json_encode($data);
?>
```

Water Monitoring: Latest Entry

```
<?php

header('Content-Type: application/json');
$db = mysqli_connect("localhost", "root", "", "teamlapia_database");

$query2 = "SELECT * FROM sensor_data ORDER BY ID DESC LIMIT 1";

$result2 = $db->query("$query2");

$data2 = [];
foreach ($result2 as $row)
{
$data2[] = $row;
}
$db->close();

echo json_encode($data2);
?>
```

Weight Prediction: Whole Set

```
<?php

header('Content-Type: application/json');

$db = mysqli_connect("localhost", "root", "", "teamlapia_database");

if (!$db) {
   echo "Error: Unable to connect to MySQL." . PHP_EOL;
   echo "Debugging errno: " . mysqli_connect_errno() . PHP_EOL;
   echo "Debugging error: " . mysqli_connect_error() . PHP_EOL;
   exit;
}

$query = "SELECT * FROM weight_prediction";
$result = $db->query("$query");
$data = [];
foreach ($result as $row)
{
$data[] = $row;
}

$db->close();
```

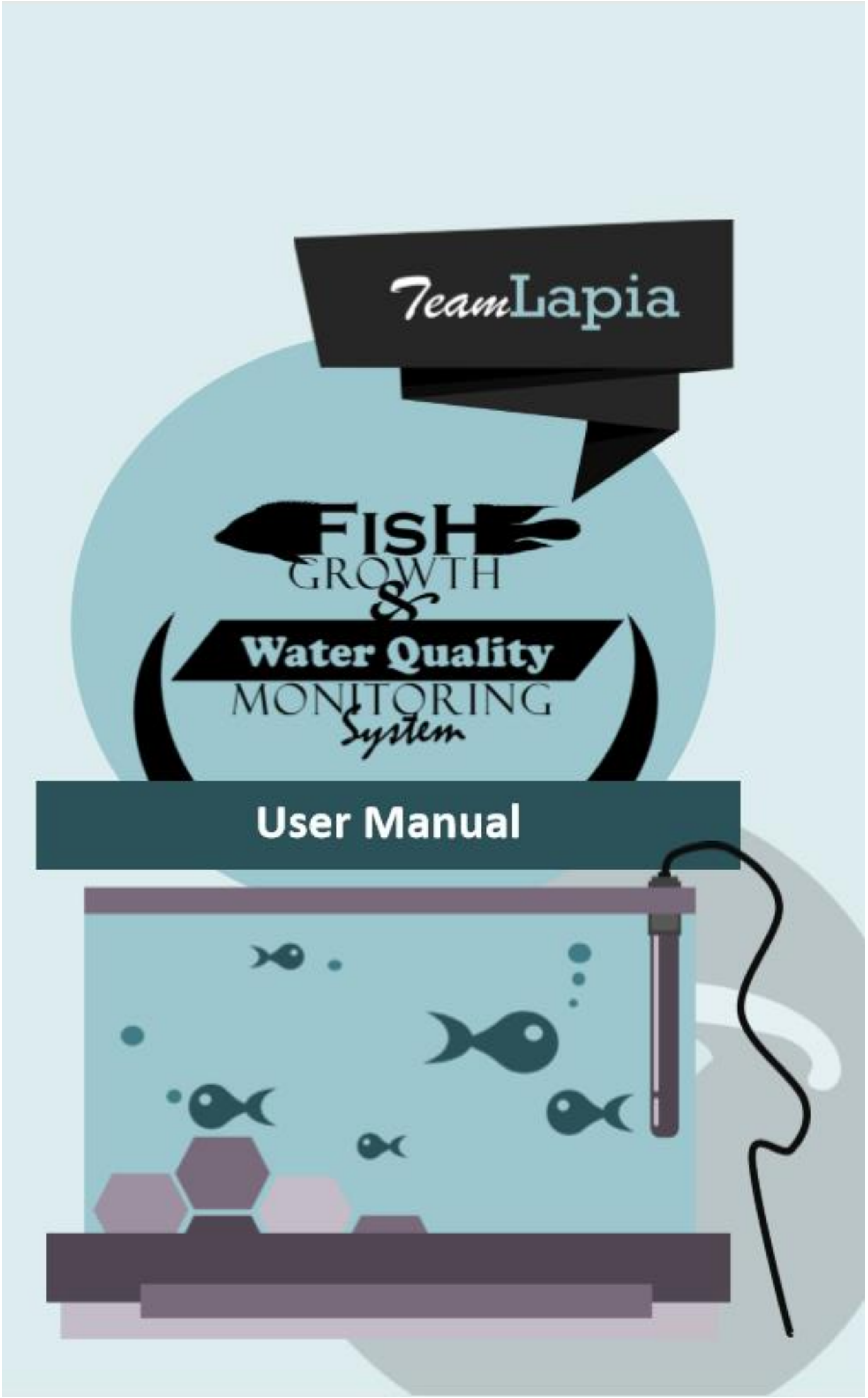
```
echo json_encode($data);
?>
```

Weight Prediction: Latest Entry

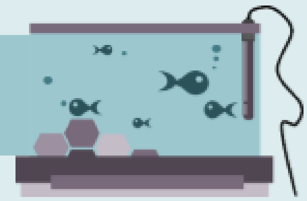
```
<?php

header('Content-Type: application/json');
$db = mysqli_connect("localhost", "root", "", "teamlapia_database");

$query2 = "SELECT * FROM weight_prediction ORDER BY ID DESC LIMIT 1";

$result2 = $db->query("$query2");

$data2 = [];
foreach ($result2 as $row)
{
$data2[] = $row;
}
$db->close();

echo json_encode($data2);

?>
```

APPENDIX B

System User Manual

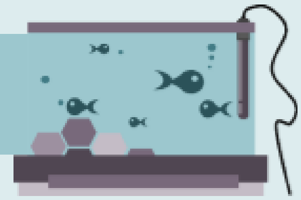

TeamLapia
FISH
GROWTH
&
Water Quality
MONITORING
System
User Manual

Panimula



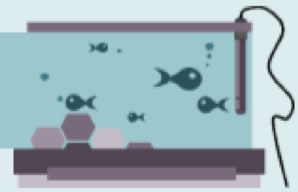
Ang **TeamLapia** System ay binubuo ng anim iba't-ibang klase ng sensors (pH, Turbidity, Temperature, ORP, Dissolved Oxygen at Salinity) at dalawang camera na tinatawag na Stereo-vision Camera. Lahat ng anim na sensors ay nakalubog sa tubig ng aquarium. Nakakabit ang anim na sensors sa Arduino kung saan maaaring makita ang mga impormasyon sa bawat isang sensors. Ang gamit ng pH Sensor ay pagkuha ng Acidity ng tubig. Kusang itatama ng systema ang Acidity sa gabay ng pH Sensor sa pamamagitan ng paglalagay ng Sodium Carbonate. Ang gamit ng Temperature Sensor ay pagkuha ng temperatura ng tubig at may isang device ang system upang mapalamig o mapainit ang tubig ayon sa tamang kailangan ng mga isda. Ang Turbidity, ORP, Dissolved Oxygen at Salinity ay naitatama sa pamamagitan ng pagpapalit ng tubig na kusang napapalitan kung hindi na tama ang bilang na kanyang pinapakita.

Paano Gamitin



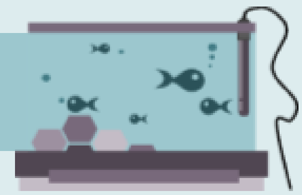
1. Ihanda ang lahat ng electronic at electrical devices na ikakabit sa aquarium. Mahalagang isa-isang icheck kung mayroong mga sirang linya ng kuryente na maaring magdulot ng pagkasunog. Gamitin ang naaayon na boltahe, 220VAC para sa mga ilaw at pump, 5VDC naman para sa Arduino at 12VDC naman para sa RaspberryPi.
2. Iset-up ang mga device na naaayon sa disenyo nito. Siguraduhing nakakakabit ng maigi ang mga device sa Arduino at RaspberryPi na makikita sa ilalim ng aquarium. Ikabit and Stereo Camera na nasa loob ng itim na kahon sa USB port ng RaspberryPi.
3. Siguraduhing malinis ang paligid ng aquarium at kung maaari ay takpan ang paligid nito upang di maapektuhan ng liwanag ang imahe na makukuha ng camera.
4. Matapos masigurong gumagana ng maayos ang lahat ng devices, maaaring mamonitor ang kalagayan nito sa aming website na www.teamlapia.com.

Mga Alituntunin at Gabay



1. Ingatan ang lahat ng electronic at electrical devices. Iwasang mabagsak o mabasa at panatilihing malinis ito hangga't maaari.
2. Iwasang magtanggal ng tubig habang nakasaksak ang Temperature Heater. Maaari itong masira at magdulot ng pagkasunog.
3. Iwasang malubog sa tubig ang mga linya ng kuryente ng mga sensors na posibleng magdulo ng short circuit sa buong set-up.
4. Gamitin ang itinakdang mga power supply sa bawat device na ginagamit sa system.
5. Iwasang mabasa ang mga nasa ilalim ng aquarium tulad ng Arduino system. Hindi ito maaaring mabasa lalo na kapag nakasaksak.
6. Maaaring manu-manong linisin ang aquarium. Siguraduhing gumamit ng dechlorinized na tubig sa paglinis at pagpalit ng tubig.

Pagpapanatili



1. Maaaring linisin ang aquarium sa pamamagitan ng pagsalok sa dumi nito na naiipon na sa aquarium. Punasan din ang labas ng aquarium upang mapanatiling malinaw ang kuha ng camera sa mga isda sa loob ng aquarium.
2. Siguraduhing gamitan ng pang dechlorinize ang tubig na ipapalit sa drum tuwing ito ay mauubos upang maging naaayon ang tubig sa isda.
3. Maaari ding icheck ang mga linya ng kuryente para masiguro na maayos ito at maiwasan ang short circuit sa setup.
4. Linisin ang tip o dulo ng mga sensor gamit ang distilled water upang mapanatiling tama ang mga nasusukat nitong pamantayan sa tubig.
5. Punasan ang lens ng camera ng malinis na basahan kung may alikabok o duming namumuo rito.

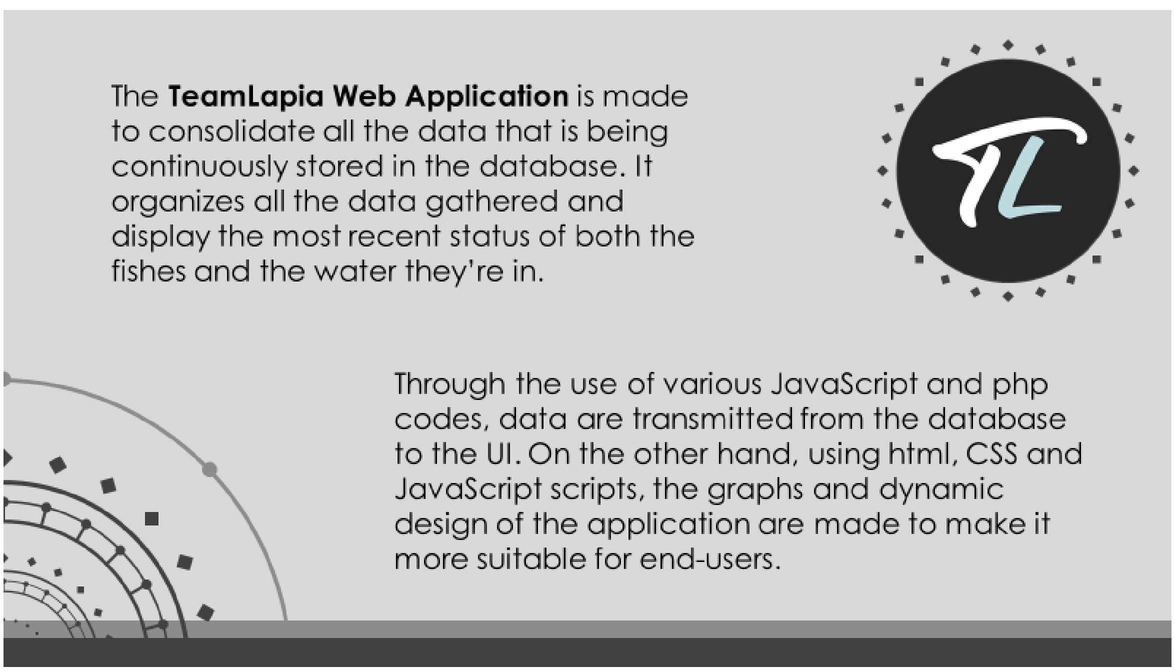
4

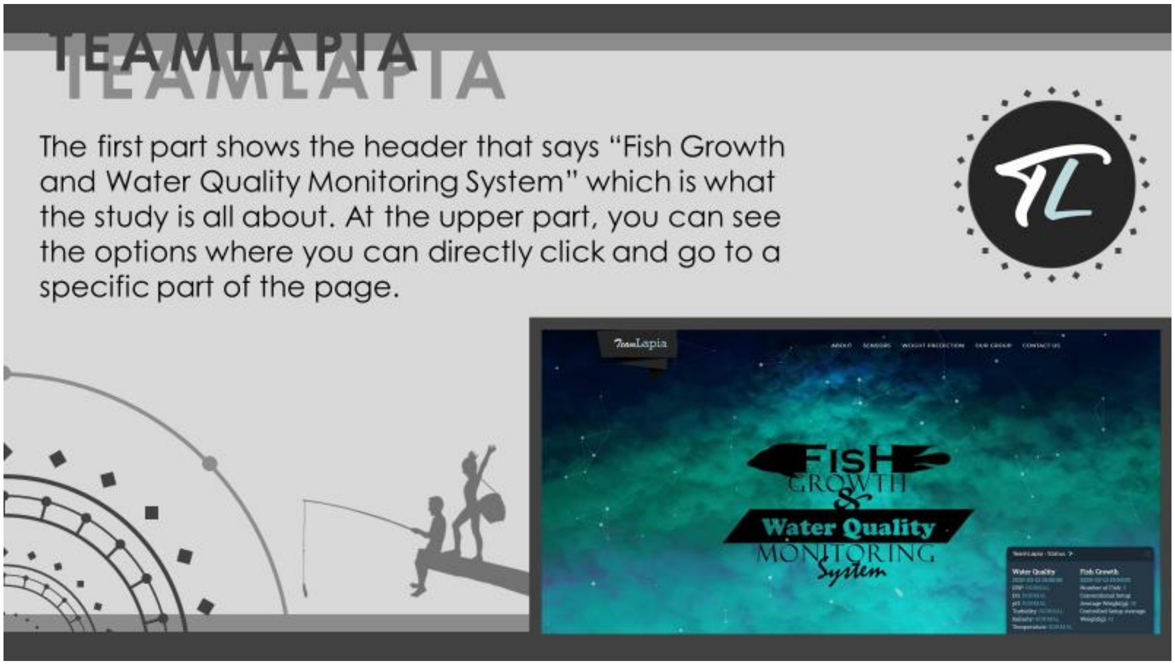
Kung may katanungan at problema tungkol sa aquaculture setup, maaaring ipagbigay alam sa amin gamit ang aming email o facebook page na makikita sa likod ng manual.

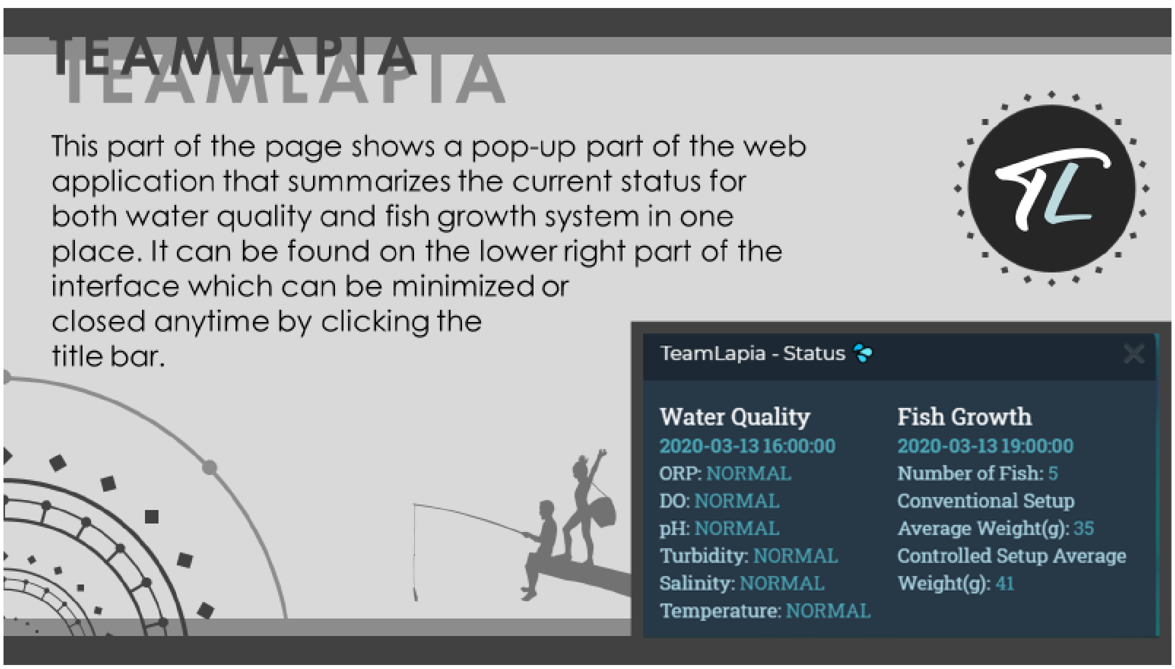
APPENDIX C

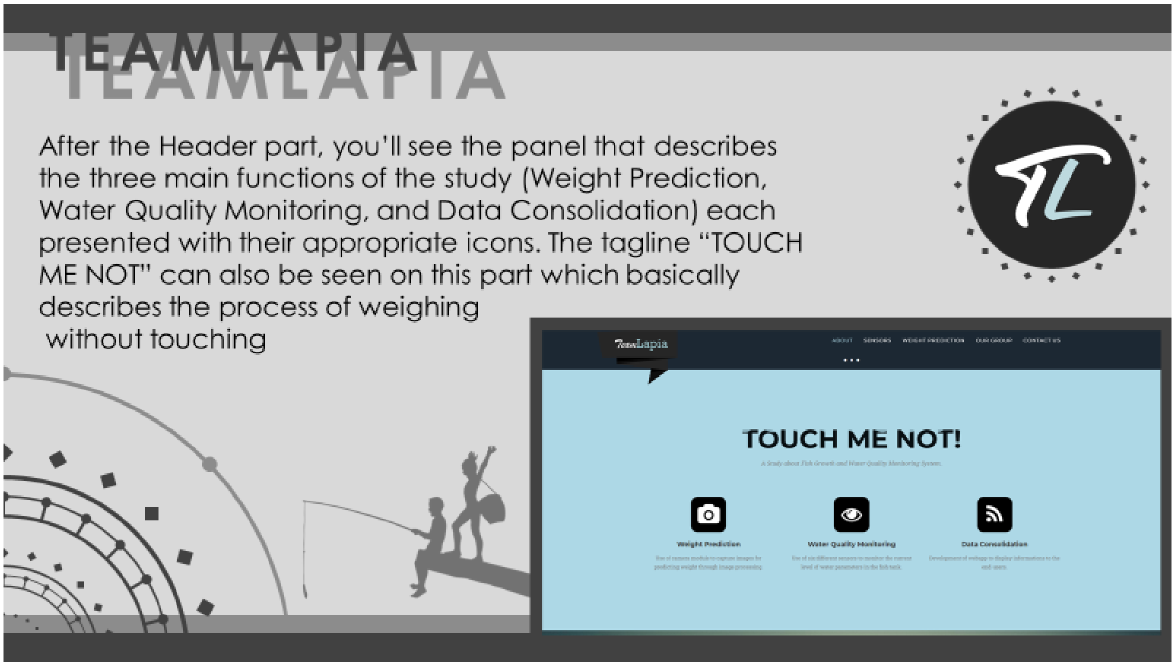
Web Application User Guide

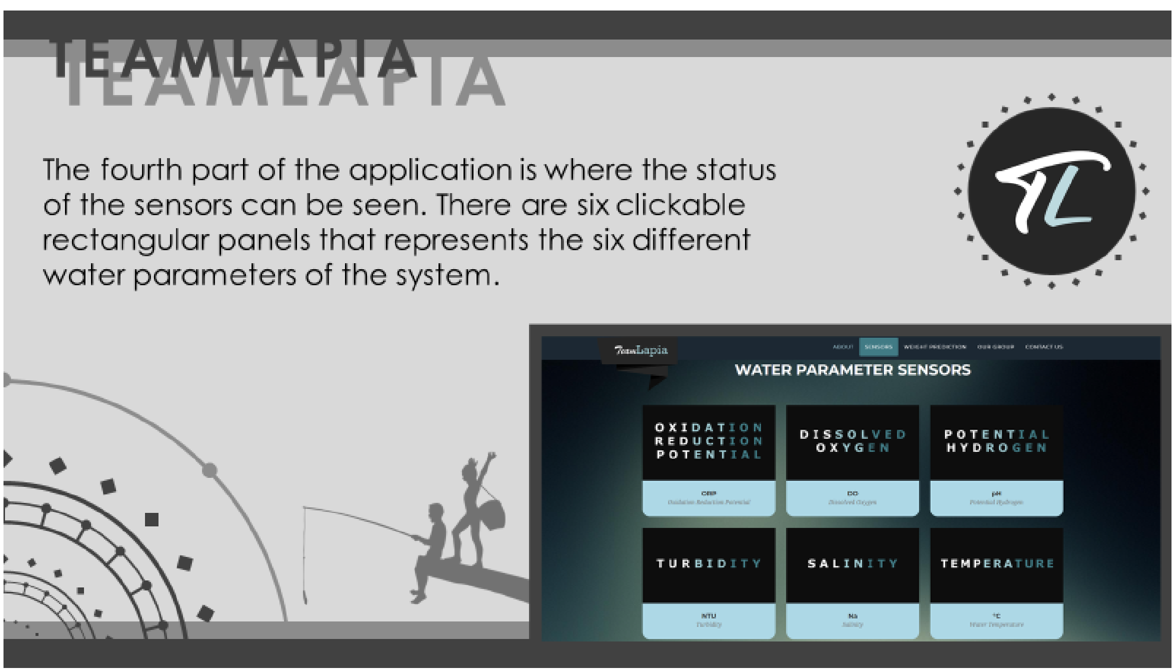

FISH GROWTH AND WATER QUALITY MONITORING SYSTEM FOR AN INTENSIVE AQUACULTURE SETUP USING RASPBERRY PI VIA LORAWAN IOT PROTOCOL
WEB APPLICATION
USER GUIDE

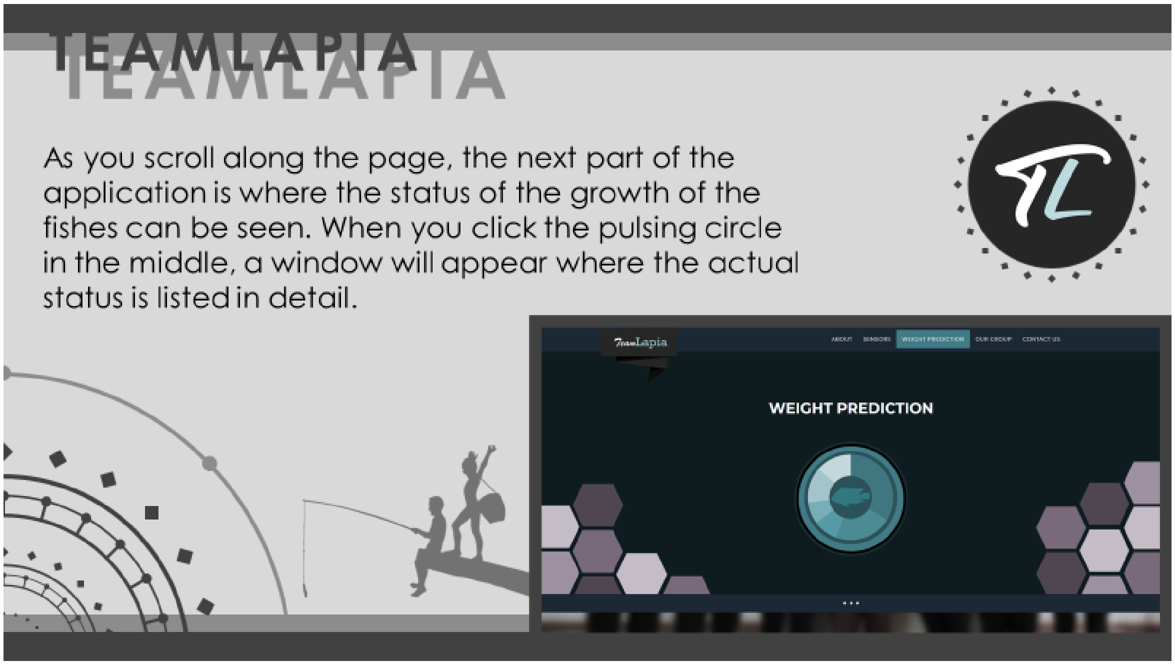

The **TeamLapia Web Application** is made to consolidate all the data that is being continuously stored in the database. It organizes all the data gathered and display the most recent status of both the fishes and the water they're in.
TL
Through the use of various JavaScript and php codes, data are transmitted from the database to the UI. On the other hand, using html, CSS and JavaScript scripts, the graphs and dynamic design of the application are made to make it more suitable for end-users.

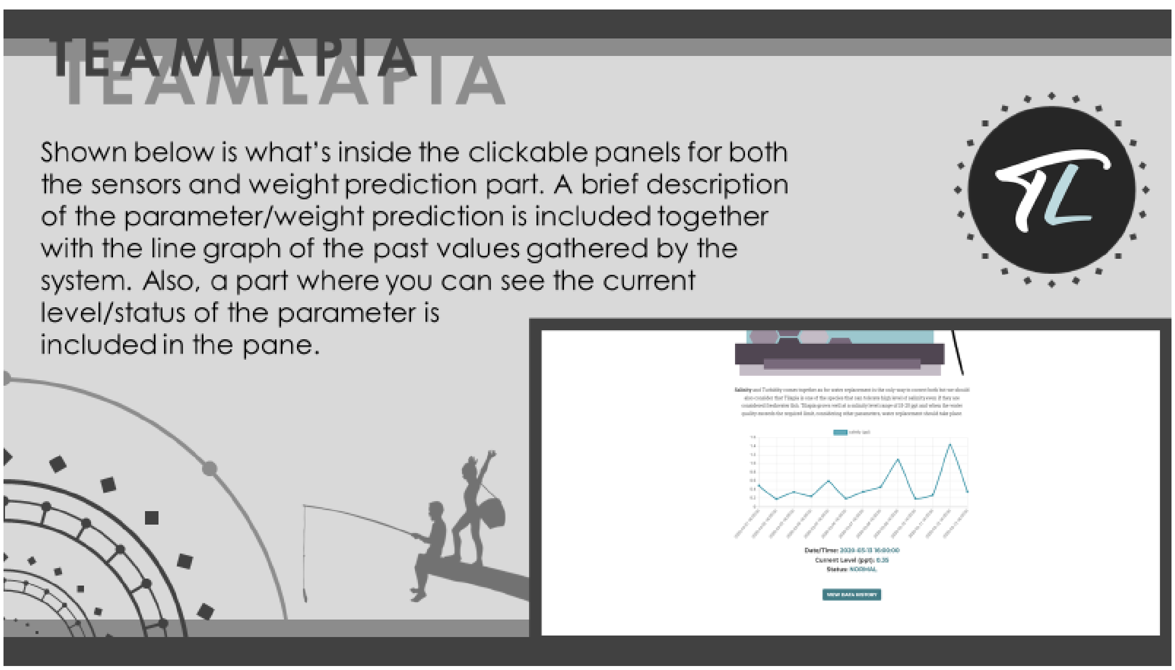

TEAMLAPIA
The first part shows the header that says "Fish Growth and Water Quality Monitoring System" which is what the study is all about. At the upper part, you can see the options where you can directly click and go to a specific part of the page.
TeamLapia
FISH
GROWTH
&
Water Quality
MONITORING
System

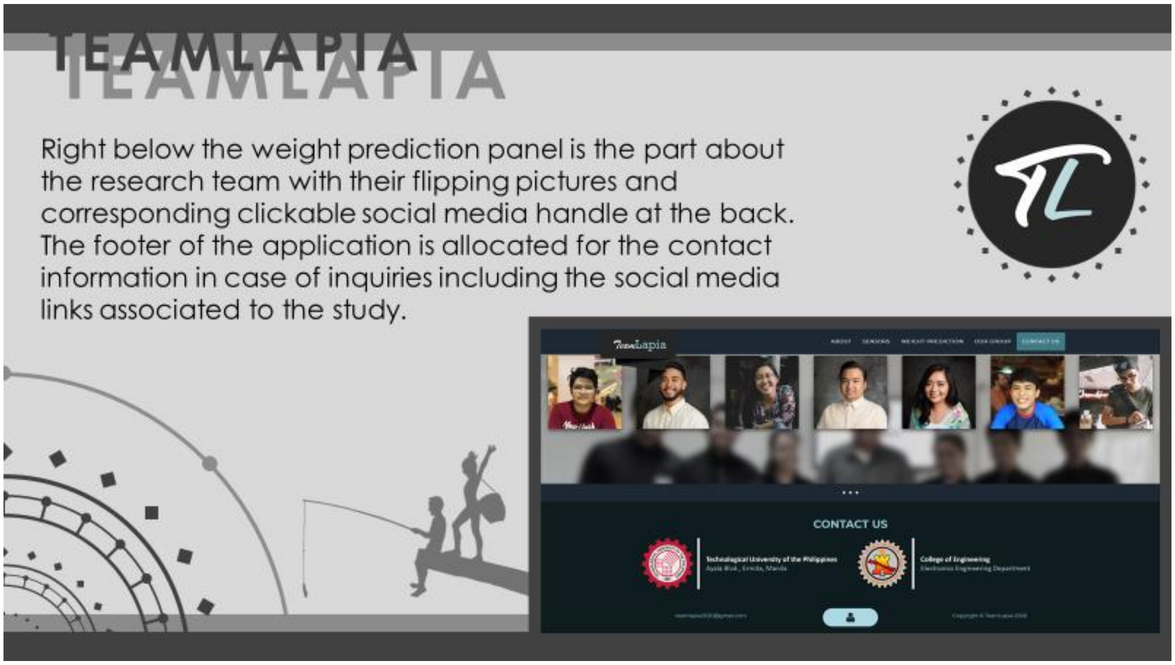

TEAMLAPIA
This part of the page shows a pop-up part of the web application that summarizes the current status for both water quality and fish growth system in one place. It can be found on the lower right part of the interface which can be minimized or closed anytime by clicking the title bar.
TeamLapia - Status
Water Quality
2020-03-13 16:00:00
ORP: NORMAL
DO: NORMAL
pH: NORMAL
Turbidity: NORMAL
Salinity: NORMAL
Temperature: NORMAL
Fish Growth
2020-03-13 19:00:00
Number of Fish: 5
Conventional Setup
Average Weight(g): 35
Controlled Setup Average
Weight(g): 41

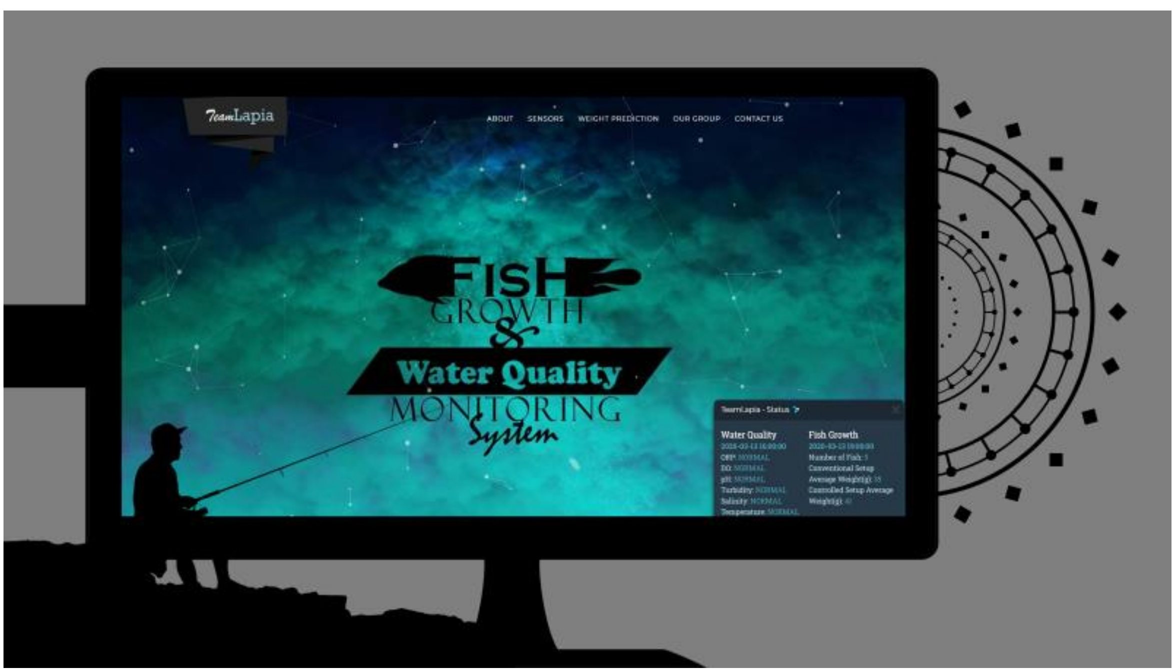

TEAMLAPIA
After the Header part, you'll see the panel that describes the three main functions of the study (Weight Prediction, Water Quality Monitoring, and Data Consolidation) each presented with their appropriate icons. The tagline "TOUCH ME NOT" can also be seen on this part which basically describes the process of weighing without touching
TeamLapia
TOUCH ME NOT!
Weight Prediction
Water Quality Monitoring
Data Consolidation


TEAMLAPIA
The fourth part of the application is where the status of the sensors can be seen. There are six clickable rectangular panels that represents the six different water parameters of the system.
TeamLapia
WATER PARAMETER SENSORS
OXIDATION REDUCTION POTENTIAL
ORP
DISSOLVED OXYGEN
DO
POTENTIAL HYDROGEN
pH
TURBIDITY
NTU
SALINITY
Na
TEMPERATURE
°C


TEAMLAPIA
As you scroll along the page, the next part of the application is where the status of the growth of the fishes can be seen. When you click the pulsing circle in the middle, a window will appear where the actual status is listed in detail.
ABOUT
SENSORS
WEIGHT PREDICTION
OUR GROUP
CONTACT US
WEIGHT PREDICTION


TEAMLAPIA
Shown below is what's inside the clickable panels for both the sensors and weight prediction part. A brief description of the parameter/weight prediction is included together with the line graph of the past values gathered by the system. Also, a part where you can see the current level/status of the parameter is included in the pane.


TEAMLAPIA
Right below the weight prediction panel is the part about the research team with their flipping pictures and corresponding clickable social media handle at the back. The footer of the application is allocated for the contact information in case of inquiries including the social media links associated to the study.
TeamLapia
CONTACT US
Technological University of the Philippines
College of Engineering


TeamLapia
ABOUT SENSORS WEIGHT PREDICTION OUR GROUP CONTACT US
FISH
GROWTH
&
Water Quality
MONITORING
System
TeamLapia - Status
Water Quality
Fish Growth

APPENDIX D

Bill of Materials

Bill of Materials

Material	Quantity	Cost
Gravity: Analog Dissolved Oxygen Sensor	1	12,000.00
Gravity:Analog Turbidity Sensor	1	1,249.50
Gravity: Analog Electric Conductivity Sensor	1	4,999.75
Waterproof Temperature Sensor DS18B20	1	129.75
Industrial Analog pH Sensor	1	3,199.75
ORP Analog Meter	1	4,997.50
Raspberry Pi Model B	1	5,377.50
Arduino Mega 2560	1	699.75
LoRa IoT Development Kit	1	5,931.60 + SF: 1,774.41
Solar Photovoltaic Panel	1	25,000.00
Solar Charge Controller	1	600.00
DC-DC Buck Converter	1	5,972.00
A4 Tech PC Camera	2	495.00
4-Channel Relay Module	1	200.00
DC 12V Water Pump	3	165.00
Servomotor	1	210.00
12x12in-Acrylic Board	1	125.00
260-Liters-Aquarium	2	3,000.00
Aquarium Stand	1	1,500.00
200-Liters-Blue Plastic Drum	1	600.00
Aerator	1	300.00
LED Strip 5m	2	249.75
Adaptor 9V 1A	3	124.75
5m-Clear Hose	1	25.00
Water Heater	1	450.00
Water Pipes	3	50.00
	TOTAL	**83,520.26**

APPENDIX E

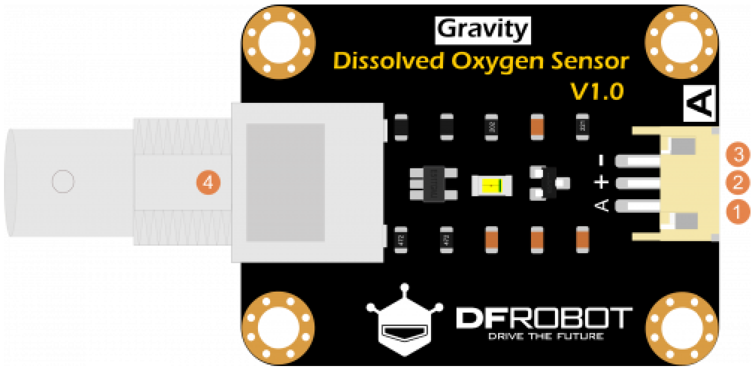
Data Sheets

DISSOLVED OXYGEN SENSOR

SPECIFICATION

- Dissolved Oxygen Probe
  - Type: Galvanic Probe
  - Detection Range: 0~20 mg/L
  - Temperature Range: 0~40 ℃
  - Response Time: Up to 98% full response, within 90 seconds (25℃)
  - Pressure Range: 0~50 PSI
  - Electrode Service Life: 1 year (normal use)
  - Maintenance Period:
    - Membrane Cap Replacement Period:
      - 1~2 months (in muddy water);
      - 4~5 months (in clean water)
    - Filling Solution Replacement Period: Once every month
  - Cable Length: 2 meters
  - Probe Connector: BNC
- Signal Converter Board
  - Supply Voltage: 3.3~5.5V
  - Output Signal: 0~3.0V
  - Cable Connector: BNC
  - Signal Connector: Gravity Analog Interface (PH2.0-3P)
  - Dimension: 42mm * 32mm/1.65 * 1.26 inches

BOARD OVERVIEW



No.	Label	Description
1	A	Analog Signal Output (0~3.0V)
2	+	VCC (3.3~5.5V)
3	-	GND
4	BNC	Probe Cable Connector

TEMPERATURE SENSOR

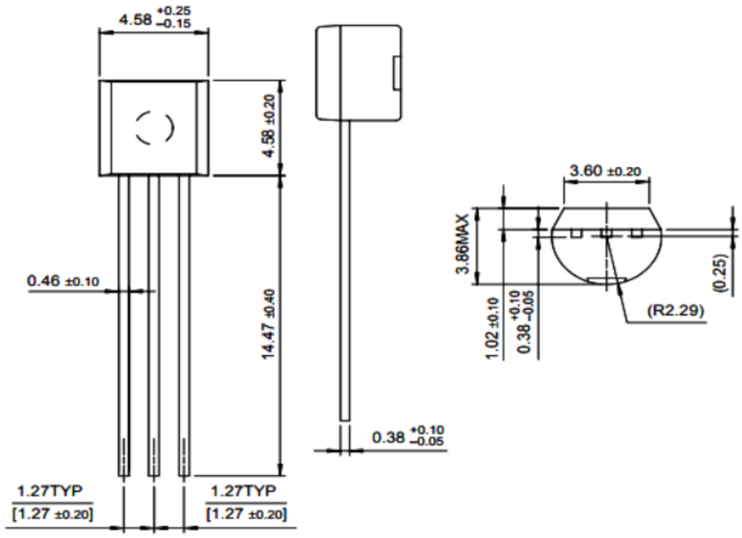
PIN CONFIGURATION

No:	Pin Name	Description
1	Ground	Connect to the ground of the circuit
2	Vcc	Powers the Sensor, can be 3.3V or 5V
3	Data	This pin gives output the temperature value which can be read using 1-wire method

SPECIFICATIONS

- Programmable Digital Temperature Sensor
- Communicates using 1-Wire method
- Operating voltage: 3V to 5V
- Temperature Range: -55°C to +125°C
- Accuracy: ±0.5°C
- Output Resolution: 9-bit to 12-bit (programmable)
- Unique 64-bit address enables multiplexing
- Conversion time: 750ms at 12-bit
- Programmable alarm options
- Available as To-92, SOP and even as a waterproof sensor

2-D MODEL



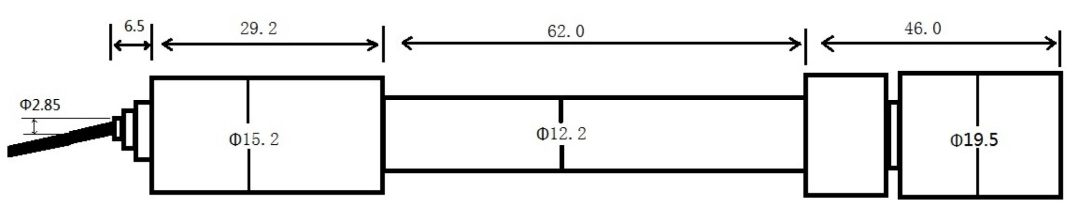
4.58 +0.25 −0.15
4.58 ±0.20
14.47 ±0.40
0.46 ±0.10
1.27TYP
[1.27 ±0.20]
1.27TYP
[1.27 ±0.20]
0.38 +0.10 −0.05
3.60 ±0.20
3.86MAX
1.02 ±0.10
0.38 +0.10 −0.05
(R2.29)
(0.25)

INDUSTRIAL ANALOG PH SENSOR

SPECIFICATION

- Module Power : 5.00V
- Module Size : 43mmx32mm（1.70″x1.26″)
- Measuring Range :0-14PH
- Measuring Temperature :0-60 ℃
- Accuracy : ± 0.1pH (25 ℃)
- Response Time : ≤ 1min
- Industry pH Electrode with BNC Connector
- PH2.0 Interface (3 foot patch)
- Gain Adjustment Potentiometer
- Power Indicator LED

DIMENSION

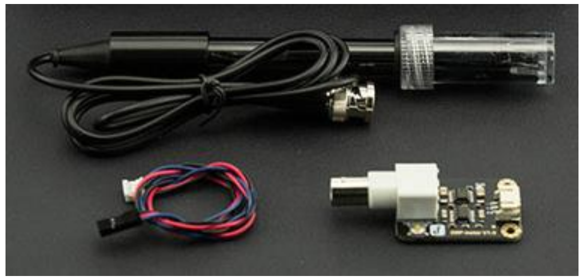


CHARACTERISTICS

The output of pH electrode is Millivolts, and the pH value if the relationship is shown as follows (25’C);

VOLTAGE （mV）	pH value	VOLTAGE （mV）	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

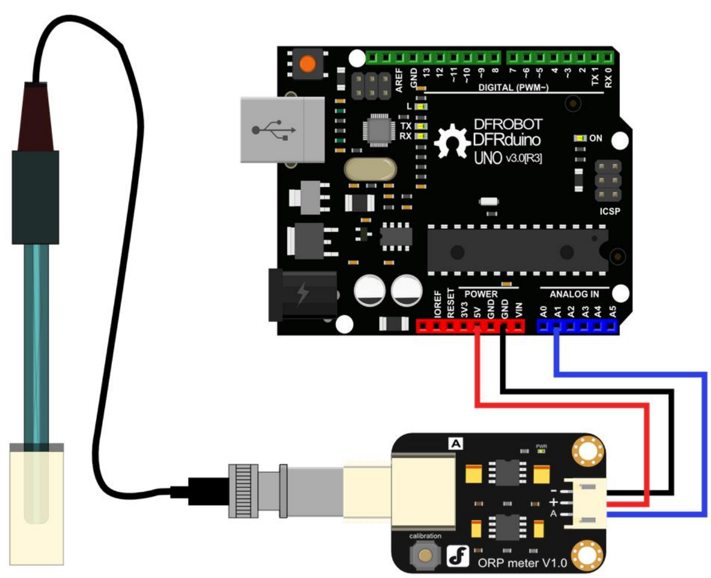
OXIDATION-REDUCTION POTENTIAL SENSOR



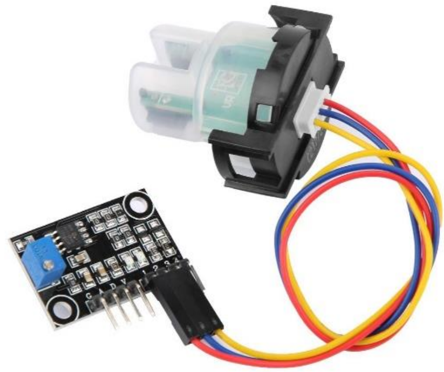
SPECIFICATION

- Module Power:+5.00V
- Module Size: 40mmX27mm(1.57"x1.06")
- Measuring Range:-2000mV—2000mV
- Suitable Temperature:5-70℃
- Accuracy:±10mv (25 ℃)
- Response Time:≤20sec
- ORP Probe with BNC Connector
- PH2.0 Interface(3 foot patch)
- Zero calibration button
- Power Indicator LED

CONNECTING DIAGRAM



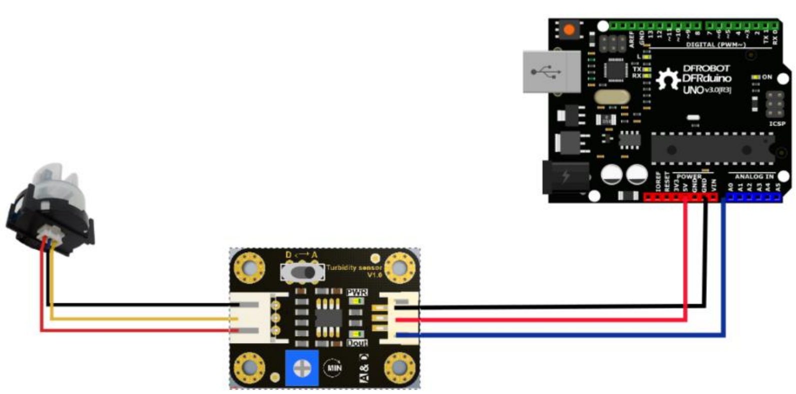
TURBIDITY SENSOR



SPECIFICATION

- Operating Voltage: 5V DC
- Operating Current: 40mA (MAX)
- Response Time : <500ms
- Insulation Resistance: 100M (Min)
- Output Method:
    - Analog output: 0-4.5V
    - Digital Output: High/Low level signal (you can adjust the threshold value by adjusting the potentiometer)
- Operating Temperature: 5℃~90℃
- Storage Temperature: -10℃~90℃
- Weight: 30g
- Adapter Dimensions: 38mm*28mm*10mm/1.5inches *1.1inches*0.4inches

CONNECTION DIAGRAM

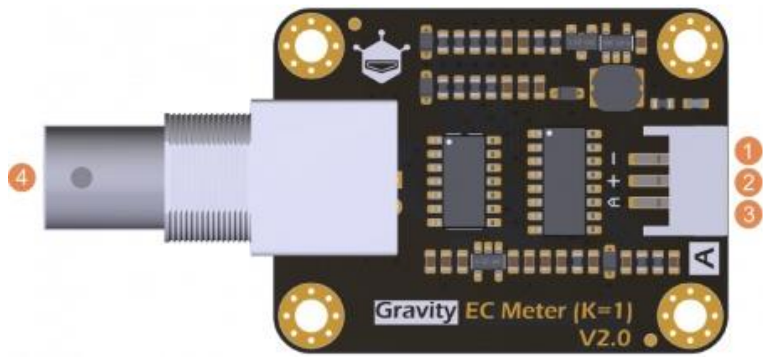


ELECTRIC CONDUCTIVITY SENSOR

SPECIFICATION

- Signal Conversion Board (Transmitter) V2
    - Supply Voltage: 3.0~5.0V
    - Output Voltage: 0~3.4V
    - Probe Connector: BNC
    - Signal Connector: PH2.0-3Pin
    - Measurement Accuracy: ±5% F.S.
    - Board size: 42mm*32mm/1.65in*1.26in
- Electrical Conductivity Probe
    - Probe Type: Laboratory Grade
    - Cell Constant: 1.0
    - Support Detection Range: 0~20ms/cm
    - Recommended Detection Range: 1~15ms/cm
    - Temperature Range: 0~40°C
    - Probe Life: >0.5 year (depending on frequency of use)
    - Cable Length: 100cm

BOARD OVERVIEW



Num	Label	Description
1	-	Power GND(0V)
2	+	Power VCC(3.0~5.0V)
3	A	Analog Signal Output(0~3.4V)
4	BNC	Probe Connector

A4 – Tech (PK-365G) WEBCAM

SPECIFICATIONS

- Resolution: 480P 640*480
- Frame Rate: 30fps
- Built-in Mic: Single Mic
- Lens Type: Glass lens
- Focus Type: Fixed Focus*
- Output Format: MJPEG YUY2
- USB Transfer: USB 2.0 High-Speed
- Viewing Angle: 54 degrees
- System Requirements: Windows 7/ 8/ 8.1/ 10 or above

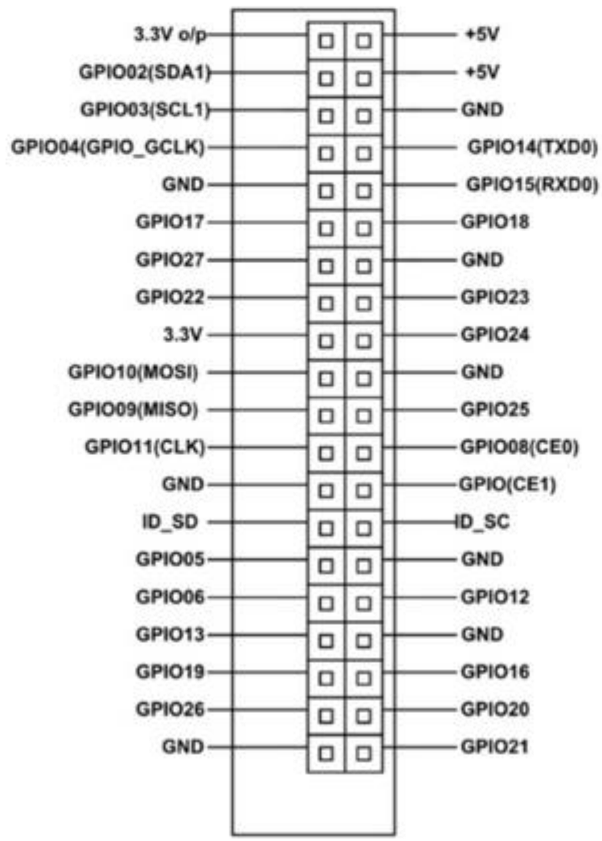
PRODUCT DIMENSION

RASPBERRY PI 3 MODEL B

SPECIFICATION

- Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
- 1GB RAM
- BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
- 100 Base Ethernet
- 40-pin extended GPIO
- 4 USB 2 ports
- 4 Pole stereo output and composite video port
- Full size HDMI
- CSI camera port for connecting a Raspberry Pi camera
- DSI display port for connecting a Raspberry Pi touchscreen display
- Micro SD port for loading your operating system and storing data
- Upgraded switched Micro USB power source up to 2.5A

PINOUT


3.3V o/p
+5V
GPIO02(SDA1)
+5V
GPIO03(SCL1)
GND
GPIO04(GPIO_GCLK)
GPIO14(TXD0)
GND
GPIO15(RXD0)
GPIO17
GPIO18
GPIO27
GND
GPIO22
GPIO23
3.3V
GPIO24
GPIO10(MOSI)
GND
GPIO09(MISO)
GPIO25
GPIO11(CLK)
GPIO08(CE0)
GND
GPIO(CE1)
ID_SD
ID_SC
GPIO05
GND
GPIO06
GPIO12
GPIO13
GND
GPIO19
GPIO16
GPIO26
GPIO20
GND
GPIO21

ARDUINO MEGA 2560

SPECIFICATION

- Microcontroller: **ATmega2560**
- Operating Voltage: 5V
- Input Voltage (recommended): 7-12V
- Input Voltage (limit) : 6-20V
- Digital I/O Pins: 54 (of which 15 provide PWM output)
- Analog Input Pins: 16
- DC Current per I/O Pin: 20 mA
- DC Current for 3.3V Pin: 50 mA+
- Flash Memory: 256 KB of which 8 KB used by bootloader
- SRAM: 8 KB
- EEPROM: 4 KB
- Clock Speed: 16 MHz
- LED_BUILTIN: 13
- Length: 101.52 mm
- Width : 53.3 mm
- Weight: 37 g

12V R385 Water Pump

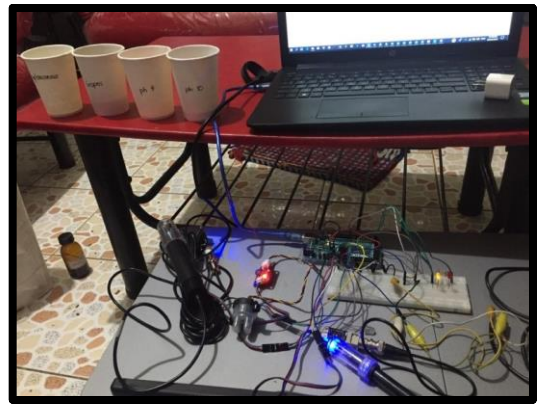
SPECIFICATION

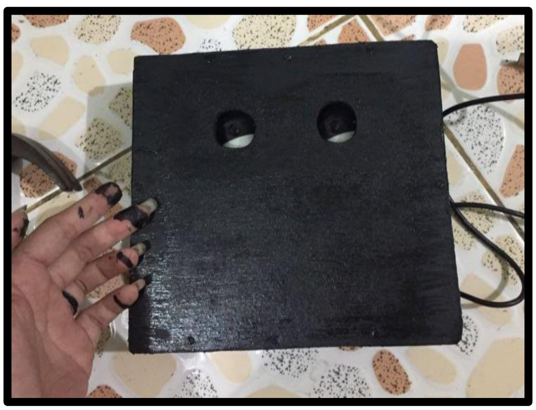
- Working voltage: DC12V
- Current work: 0.5-0.7A (6W power must achieve above)
- Traffic: 1.5-2L / Min (left and right)
- The maximum suction lift: 2 m
- Head: Vertical up to 3 meters
- Lifetime: up to 2500H
- Withstand water temperature: up to 80 ℃

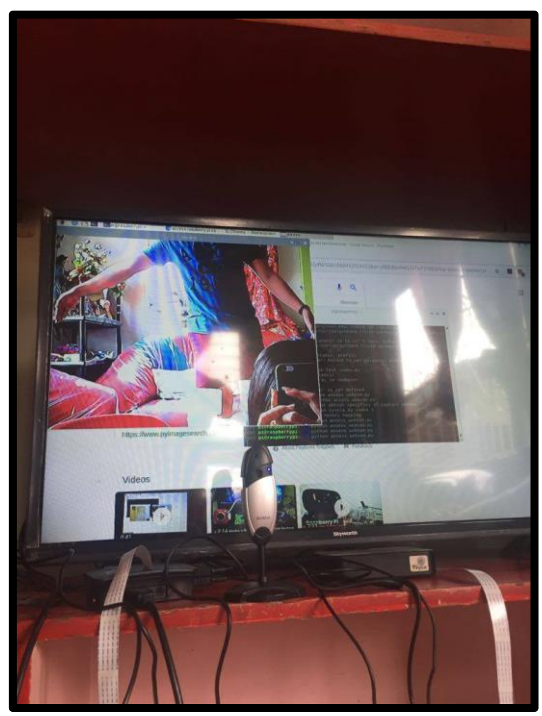
APPENDIX F

Project Documentation

1. Project Development

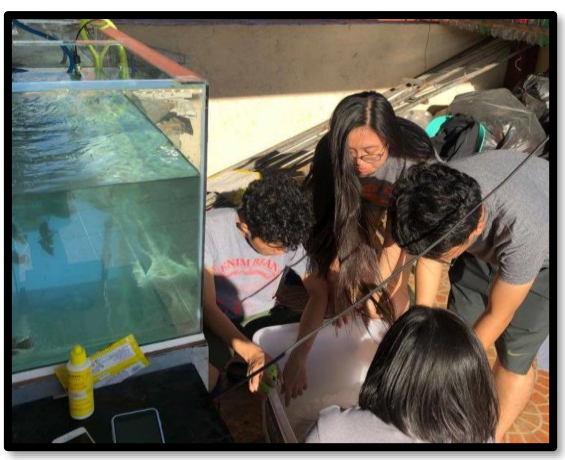


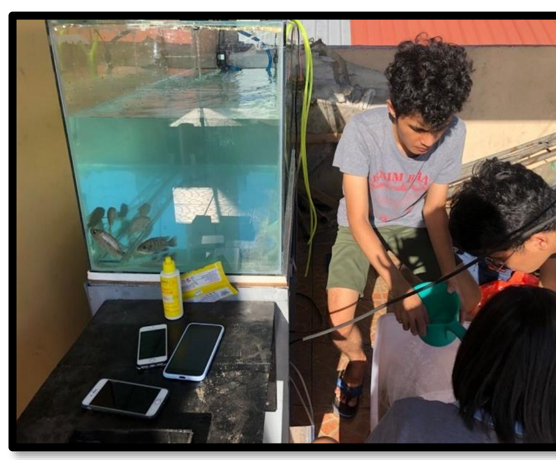




2. Meeting and Consulation with the Barangay Captain of Brgy. Pulang Lupa 1 Las Piñas City for deployment

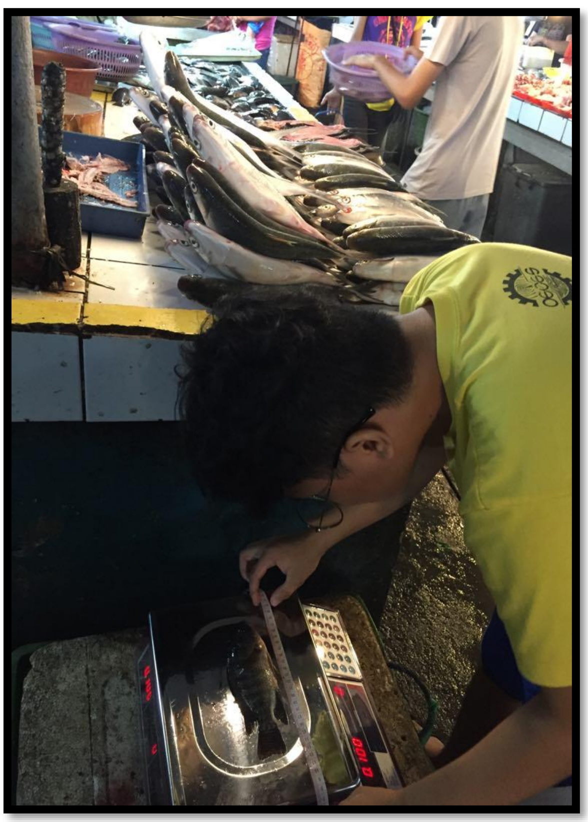
3. Assembling and Setting-up of Aquaculture System
(Brgy. Pulang Lupa 1, Las Piñas City Rooftop)





4. Procurement of Tilapia Fingerlings at Sto. Domingo, Laguna
(February 28, 2020)

5. Data Gathering at Tatawid Public Market, Malabon City



6. Data Gathering for Manual Growth Measurement

7. Proponents after Defense

Topic Defense

Title Defense

Progress Report

Pre-final Defense

APPENDIX G

Proponent’s Profile

#650-27 J.M. Templora St.,
Santulan, Malabon City
+63 956 302 2171
jatticamina@gmail.com

JATT D. ICAMINA

OBJECTIVE

To utilize my skills and abilities in the field of electronics engineering that offers professional growth while being resourceful, innovative and flexible as well as to practice my leadership skills and decision- making skills.

SKILLS

WORK

- C++ and Python Languages
- Machine Learning
- Computer Vision
- Electronic Workbenches
- Computing Software (MATLAB)
- Arduino
- Raspberry Pi

PERSONAL

- Fast learner and adaptability
- Complex problem solver
- Good communication skills

CHARACTER REFERENCE

ENGR. LEAN KARLO S. TOLENTINO
DIRECTOR
Office of the University Extension Services, TUP-Manila
+63 915 468 8227

ENGR. NILO M. ARAGO
COLLEGE SECRETARY
College of Engineering,
TUP-Manila
+63 915 468 8227

PRE-PROFESSIONAL EXPERIENCE

(APRIL 2019-AUGUST 2019)
FASTRAC: CATCH-ALL: Vision-Based Artificial Analytics Software Development for Traffic and Transport Applications
RESEARCH ASSISTANT
De La Salle University

AFFILIATIONS

(JUNE 2014-PRESENT)
ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS (OECES)
MEMBER
Bachelor of Science in Electronics Engineering
Technological University of the Philippines – Manila

(JUNE 2016-PRESENT)
INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES – MANILA STUDENT CHAPTER
MEMBER
Manila Student Chapter

(JUNE 2017-MARCH 2018)
INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES – MANILA STUDENT CHAPTER
BOARD OF DIRECTOR
Manila Student Chapter

(OCTOBER 2015-MARCH 2016)
UNIVERSITY STUDENT GOVERNMENT – TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES
OIC GOVERNOR – COLLEGE OF ENGINEERING STUDENT COUNCIL
Technological University of the Philippines – Manila

(JUNE 2015-OCTOBER 2015)
UNIVERSITY STUDENT GOVERNMENT – TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES
VICE GOVERNOR – COLLEGE OF ENGINEERING STUDENT COUNCIL
Technological University of the Philippines – Manila

EDUCATIONAL BACKGROUND

(2014-PRESENT)
TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA
BACHELOR OF SCIENCE IN ELECTRONICS ENGINEERING
Ayala Boulevard, Ermita, Manila

(2010-2014)
TINAJEROS NATIONAL HIGH SCHOOL
6th HONORABLE MENTION
B. Rivera St., Tinajeros, Malabon City

PROJECT STUDY

TeamLapia: An IOT-based Intensive Aquaculture Monitoring System with Water Quality Correction and Fish Weight Prediction. The system involves monitoring and acquiring real-time data of six different water quality parameters with automatic correction and acquiring fish weight through image processing and predictive analysis.

APOLO A. VILLANUEVA

842 M. Bernal Street, Daniel Fajardo, Las Piñas City
+63 998 196 3237
villanuevapolo3.14@gmail.com

OBJECTIVE

To utilize my skills and abilities in the field of electronics engineering that offers professional growth while being resourceful, innovative and flexible as well as to practice my leadership skills and decision-making skills.

SKILLS

- Web Development & Design
- Electronic Workbenches
- Computer Hardware Servicing
- Microsoft Office Applications
- Computing Software (MATLAB)
- Arduino
- Raspberry Pi
- Fast learner and adaptability
- Complex problem solver
- Good communication skills
- People-oriented

CHARACTER REFERENCE

ENGR. LEAN KARLO S. TOLENTINO
DIRECTOR
Office of the University Extension Services, TUP-Manila
+63 915 468 8227

ENGR. NILO M. ARAGO
COLLEGE SECRETARY
College of Engineering,
TUP-Manila
+63 915 468 8227

PRE-PROFESSIONAL EXPERIENCE

(APRIL 2019-MAY 2019)
SIGNALING AND TELECOMMUNICATIONS SECTION
INTERN TECHNICIAN
Light Rail Manila Corporation

AFFILIATIONS

(JUNE 2014-PRESENT)
ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS (OECES)
MEMBER
Bachelor of Science in Electronics Engineering
Technological University of the Philippines – Manila

(JUNE 2016-PRESENT)
INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES – MANILA STUDENT CHAPTER
MEMBER
Manila Student Chapter

EDUCATIONAL BACKGROUND

(2014-PRESENT)
TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA
BACHELOR OF SCIENCE IN ELECTRONICS ENGINEERING
Ayala Boulevard, Ermita, Manila

(2010-2014)
LAS PIÑAS NATIONAL HIGH SCHOOL
5th HONORABLE MENTION
Sultana Rd., Daniel Fajardo, Las Piñas City

(2004-2010)
MANUYO ELEMENTARY SCHOOL
Salutatorian
Tramo St., Manuyo Uno, Las Piñas City

PROJECT STUDY

TeamLapia: An IOT-based Intensive Aquaculture Monitoring System with Water Quality Correction and Fish Weight Prediction. The system involves monitoring and acquiring real-time data of six different water quality parameters with automatic correction and acquiring fish weight through image processing and predictive analysis.

PERSONAL INFORMATION

Date of Birth, Age: November 6, 1997, 22
Gender: Male
Height: 5'4
Weight: 46 kg.
Religion: Roman Catholic
Civil Status: Single
Citizenship: Filipino

B15 L14 Pinatubo St., Dolmar Golden Hills 2, Deparo, Caloocan City
+63 945 620 3197
giandearroz@outlook.com

OBJECTIVE

To utilize my skills and abilities in the field of Electronics Engineering industry that offers professional growth while being resourceful, innovative, and flexible, as well as to practice my leadership skills and decision-making skills, and to be a part of a team whose development I can contribute to as a member

SKILLS

- Knowledgeable at Electronics and Communication
- Knowledgeable at MATLAB
- Knowledgeable at Octave
- Able to adapt effectively to challenging and emergency situation
- Technical Skills (Soldering and Troubleshooting
- Computer Skills (Microsoft Offce)

CHARACTER REFERENCE

ENGR. LEAN KARLO S. TOLENTINO
DIRECTOR
Office of the University Extension Services, TUP-Manila
+63 915 468 8227

ENGR. TIMOTHY M. AMADO
ECE FACULTY
College of Engineering,
TUP-Manila
+63 932 867 2868

GIAN CARLO D SOBREVILLA

PRE-PROFESSIONAL EXPERIENCE

(APRIL 2019- JUNE 2019)
SIGNALING AND TELECOMMUNICATIONS
INTERN TECHNICIAN
LIGHT RAIL MANILA CORPORATION

AFFILIATIONS

(JUNE 2014-PRESENT)
ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS (OECES)
MEMBER
Bachelor of Science in Electronics Engineering
Technological University of the Philippines – Manila

(JUNE 2016-PRESENT)
INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES – MANILA STUDENT CHAPTER
MEMBER
Manila Student Chapter
(JUNE 2017-MARCH 2018)

EDUCATIONAL BACKGROUND

(2014-PRESENT)
TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA
BACHELOR OF SCIENCE IN ELECTRONICS ENGINEERING
Ayala Boulevard, Ermita, Manila

(2010-2014)
NOVALICHES HIGH SCHOOL
1 Lakandula St., TS Cruz Subdivision, Novaliches, Quezon City

SAN AGUSTIN ELEMENTARY SCHOOL
3rd HONORABLE MENTION
1 Lakandula St., TS Cruz Subdivision, Novaliches, Quezon City

PROJECT STUDY

TeamLapia: An IOT-based Intensive Aquaculture Monitoring System with Water Quality Correction and Fish Weight Prediction. The system involves monitoring and acquiring real-time data of six different water quality parameters with automatic correction and acquiring fish weight through image processing and predictive analysis.

JOHN BENJAMIN E. NAVARRO

Blk 300 Mendiola St.,
Alabang, Muntinlupa City
+63 928 017 3370
orravanjb@gmail.com

OBJECTIVE

To secure an engineering position where I can utilize my experience by optimizing processes to improve efficiency and accuracy when applicable.

SKILLS

WORK
- Python
- Electronic workbenches
- Fire Detection and Alarm System (FDAS)
- Computing Software (MATLAB)
- Arduino
- Raspberry Pi

PERSONAL
- Critical Thinking Skills
- Active listening
- Adaptability

CHARACTER REFERENCE

ENGR. LEAN KARLO S. TOLENTINO
DIRECTOR
Office of the University Extension Services, TUP-Manila
+63 915 468 8227

ENGR. NILO M. ARAGO
COLLEGE SECRETARY
College of Engineering,
TUP-Manila
+63 915 468 8227

PRE-PROFESSIONAL EXPERIENCE

(MAY 2019-JUNE 2019)
Fire Detection and Alarm System Design
STUDENT TRAINEE
Net Pacific

AFFILIATIONS

(JUNE 2017-PRESENT)
ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS (OECES)
MEMBER
Bachelor of Science in Electronics Engineering
Technological University of the Philippines – Manila

(JUNE 2019-PRESENT)
INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES – MANILA STUDENT CHAPTER
MEMBER
Manila Student Chapter

EDUCATIONAL BACKGROUND

(2014-PRESENT)
TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA
BACHELOR OF SCIENCE IN ELECTRONICS ENGINEERING
Ayala Boulevard, Ermita, Manila

(2010-2014)
MUNTINLUPA NATIONAL HIGH SCHOOL
University Rd, Poblacion, Muntinlupa City

PROJECT STUDY

TeamLapia: An IOT-based Intensive Aquaculture Monitoring System with Water Quality Correction and Fish Weight Prediction. The system involves monitoring and acquiring real-time data of six different water quality parameters with automatic correction and acquiring fish weight through image processing and predictive analysis.

29 P. Gabriel St. Sipac-Almacen, Navotas City
+63 966 711 9247
cellinedepedro@yhaoo.com

OBJECTIVE

To associate my knowledge and skills acquired from the university in the field of Electronics Engineering Industry

SKILLS

- MATLAB and OCTAVE Computing Software
- Electronic Workbenches
- Computer Hardware Servicing
- Microsoft Office Applications
- Adobe Photoshop and Illustrator
- Arduino
- Raspberry Pi
- Flexibility and adaptability
- Collaboration
- Good communication skills
- People-oriented

CHARACTER REFERENCE

ENGR. LEAN KARLO S. TOLENTINO
DIRECTOR
Office of the University Extension Services, TUP-Manila
+63 915 468 8227

ENGR. NILO M. ARAGO
COLLEGE SECRETARY
College of Engineering, TUP-Manila
+63 915 468 8227

CELLINE P. DE PEDRO

PRE-PROFESSIONAL EXPERIENCE

(MAY 2019-JUNE 2019)
ENGINEERING DEPARTMENT
INTERN TECHNICIAN
TelTrends International Corporation

AFFILIATIONS

(JUNE 2014-PRESENT)
ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS (OECES)
MEMBER
Bachelor of Science in Electronics Engineering
Technological University of the Philippines – Manila

(JUNE 2016-PRESENT)
INSTITUTE OF ELECTRONICS ENGINEERS OF THE PHILIPPINES – MANILA STUDENT CHAPTER
MEMBER
Manila Student Chapter

EDUCATIONAL BACKGROUND

(2014-PRESENT)
TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA
BACHELOR OF SCIENCE IN ELECTRONICS ENGINEERING
Ayala Boulevard, Ermita, Manila

(2010-2014)
ARELLANO UNIVERSITY – MALABON
Gov. Pascual St., Malabon City

(2004-2010)
Tangos Elementary School
J. Pascual St., Navotas City

PROJECT STUDY

TeamLapia: An IOT-based Intensive Aquaculture Monitoring System with Water Quality Correction and Fish Weight Prediction. The system involves monitoring and acquiring real-time data of six different water quality parameters with automatic correction and acquiring fish weight through image processing and predictive analysis.

PERSONAL INFORMATION

Date of Birth: October 22, 1997
Gender: Female
Height: 5'0
Weight: 53 kg.
Religion: Methodist
Civil Status: Single
Citizenship: Filipino

Block 4 Lot 31 Ph2 Villa de Primarosa, Buhay na Tubig, Imus, Cavite
+63 9219622624
luigisalvacion@gmail.com

OBJECTIVE

Hardworking and passionate Electronics Engineering graduate seeking to use my superior knowledge in Electronics to effectively serve your company while gaining valuable work experience and knowledge.

SKILLS

WORK

• Circuit Designing
• Electronic Workbenches
• MATLAB
• Railway Signaling and Low Voltage Applications
• PLCs (Programmable Logic Circuits)

PERSONAL

• Critical Thinking Skills
• Good communication skills

CHARACTER REFERENCE

ENGR. LEAN KARLO S. TOLENTINO
DIRECTOR
Office of the University Extension Services, TUP-Manila
+63 915 468 8227

LUIGI JAMES D. SALVACION

PRE-PROFESSIONAL EXPERIENCE

(APRIL 2019 – JUNE 2019)

Signaling and Telecommunications Section
INTERN TECHNICIAN
Light Rail Manila Corporation (LRT 1)
Andrews Ave., Pasay City

AFFILIATIONS

(JUNE 2014-PRESENT)

ORGANIZATION OF ELECTRONICS ENGINEERING STUDENTS (OECES)
MEMBER
Bachelor of Science in Electronics Engineering
Technological University of the Philippines – Manila

EDUCATIONAL BACKGROUND

(2014-PRESENT)

TECHNOLOGICAL UNIVERSITY OF THE PHILIPPINES - MANILA
BACHELOR OF SCIENCE IN ELECTRONICS ENGINEERING
Ayala Boulevard, Ermita, Manila

(2010-2014)

PASAY CITY SCIENCE HIGH SCHOOL
2888 Vergel St., Pasay City

(2006-2010)

IMUS PILOT ELEMENTARY SCHOOL
Nueno Ave., Imus, Cavite

PROJECT STUDY

TeamLapia: An IOT-based Intensive Aquaculture Monitoring System with Water Quality Correction and Fish Weight Prediction. The system involves monitoring and acquiring real-time data of six different water quality parameters with automatic correction and acquiring fish weight through image processing and predictive analysis.