The Strawberi E-Tank

**

Group:

- M2

Members:

-Chase Badalato

-Chase Fridgen

-Haydn Brown

Problem Context and Motivation:

All pet owners can agree that they love having their pets around and always want to keep them happy and healthy. However, people tend to lead busy lives and cannot always be there to tend to a pets needs.

Aquariums and other stores/organizations are responsible for managing and ensuring the health and safety of a large number of animals at a time, and leaving this to humans can cause problems in organization and reliability. Allowing all the animals habitats to be managed simply through an interface ensures the quality of their care. And in the case of a legal issue where accountability needs to be held, a record of care can be produced.

Problem Statement:

The objective of this project is to make caring for an animal easier through the use of automation, remote controlled systems, data collection, and visualization. The remote systems allow a user to care for the animals needs at any time and any place by giving it food, water, changing its habitat temperature and having a visual to ensure its safety. The data collection and visualization allow the user to track habitat conditions and have a record of information if needed. It also allows the system to regulate itself in an event such as overheating.

All Functional Features:

- Give your pet food or water remotely
- Be able to check on your pet in real time with the HD Camera
- Keep track of your pets movements and sleep schedule
- Set the temperature to a temperature of your liking, or set the temperature to be the current live temperature of anywhere in the world
- Change the intensity of the heat lamp / LED from within your app

**https://www.iconfinder.com/icons/2274784/berry_food_fruit_health_strawberry_vitamin_icon

UML:

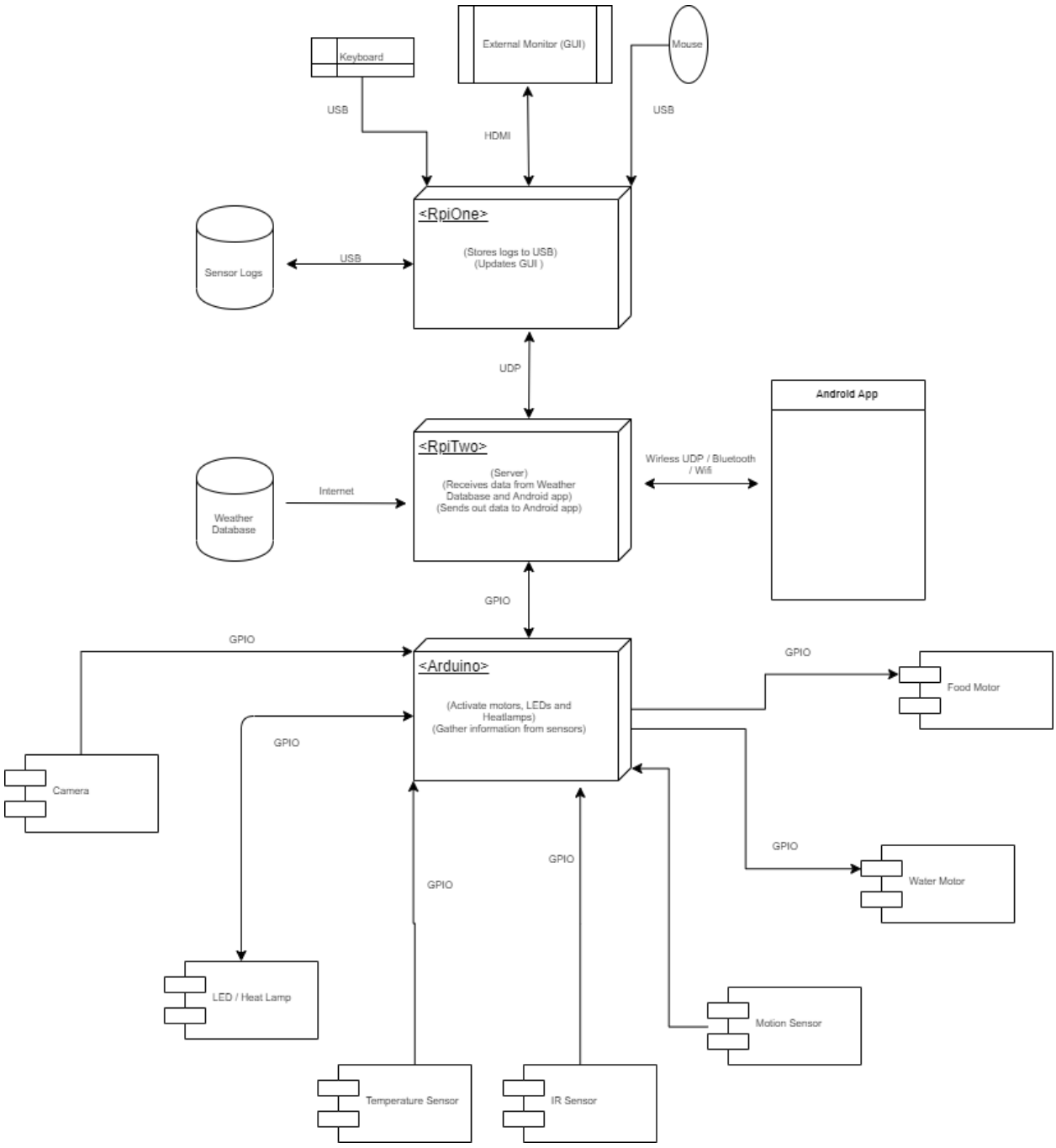


Figure 1: UML Diagram of Strawberi E-Tank

Sensors and Motors:

Motion and IR Sensor: Keep track of an animal's movements and with this data sleep schedule.

Temperature Sensor: Gather the current temperature of the room

Food and Water Actuators: These motors will be what actually dispenses the food into the pets food bowls

Camera: This will record a video of your pet that can be viewed in real time

LED / Heat Lamp: This will be the light source and heat source of the terrarium.

Technical Description:

There will be 3 microcomputers used. 2 Raspberry Pi's (Rpi1 and Rpi2) and an Arduino. Rpi1 will act as a database and a bridge between the GUI and the Rpi2 which acts as our server. The database is in the form of a USB stick that saves logs generated by the various sensors and exports them into JSON files. The Rpi2 will be the headless pi that uses wifi. It will connect to a weather website to gather information about the current temperature at a previously inputted location. The Arduino will deal with the input and output of the sensors, motors, and LEDs. It will be connected to all of these through GPIO. All of the information that is gathered will be sent to the Rpi2 to make decisions of what to do with the data. The Rpi2 will receive packets of information from the Android app and GUI, decide what the information means, and sends instructions to the Ardunio. The GUI and Android app are what the user will be using to communicate with the rest of the system.

General Layout:

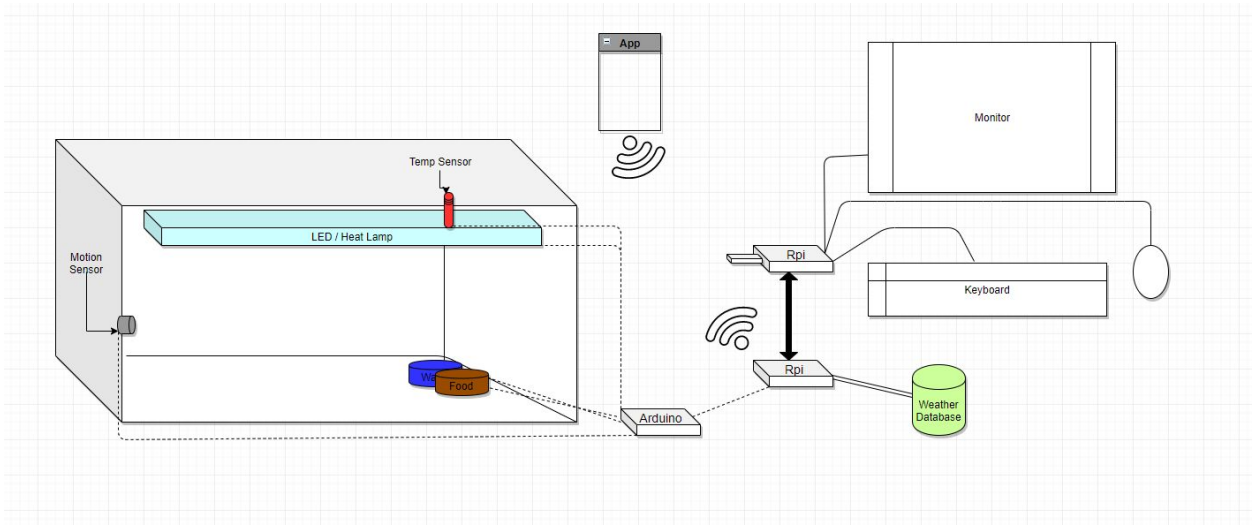


Figure 2: General Layout of Strawberi E-Tank

UML Class Diagram:

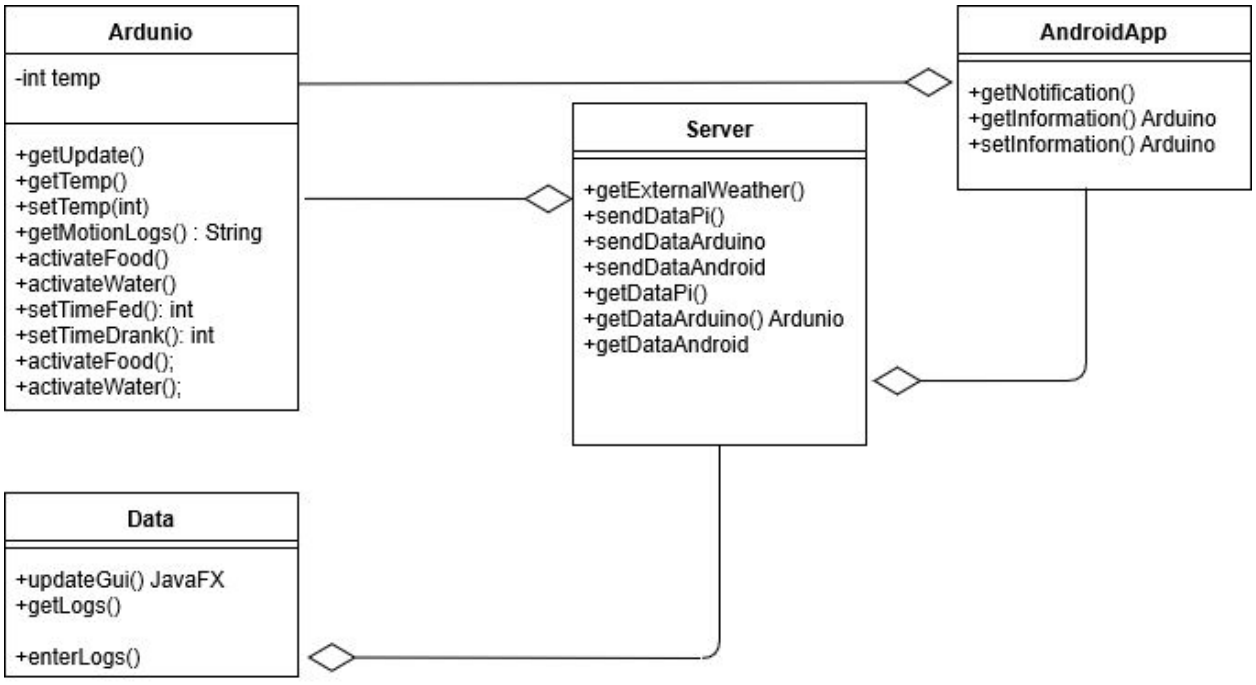


Figure 3: Basic UML Class Diagram of Strawberi E-Tank

Testing:

The three main components to our system are the arduino which operators the sensors, the Pi that sends data to and from the arduino, app, and other Pi, and finally the Pi that displays the interface and stores data obtained from the sensors. While these components will be tested both incrementally and iteratively, when a new feature is added to any of them it will be tested before more functionality is implemented. Once these components functions in isolation, they will be connected to form the whole system, which will again be tested to ensure its function.

Testing the Arduino:

- To test the motors that will dispense food and water we can use a button on a breadboard to make ensure they work and will be able to open/close
- To test the motion and IR sensors we will have a group member walk or wave their hand in front of the sensor to ensure it responds to the motion/IR given off
- To test the temperature sensor we will move the setup between varying environments, such as in a warm room and near an open window with cold air flowing over the sensor
- To test the camera, it will connected to a monitor so we can ensure that it outputs video correctly
- To test the LED/heat lamp we can simply use the setup from lab 3 where we attached it to the breadboard to ensure it turns on properly

Testing the Pi used for GUI:

- To test the GUI, we will have the application running on the Pi while it is connected to a monitor, keyboard, and mouse. We will ensure that all the buttons/input features react when used and that information is displayed for the user
- To test the database we will use sample records which we will either create or download, and using these we will upload them to the database and then fetch them
- The applications that run on this Pi (analyzing/processing our data) will be run initially without a connection to our database and GUI, to ensure they operate as desired

Testing the headless Pi:

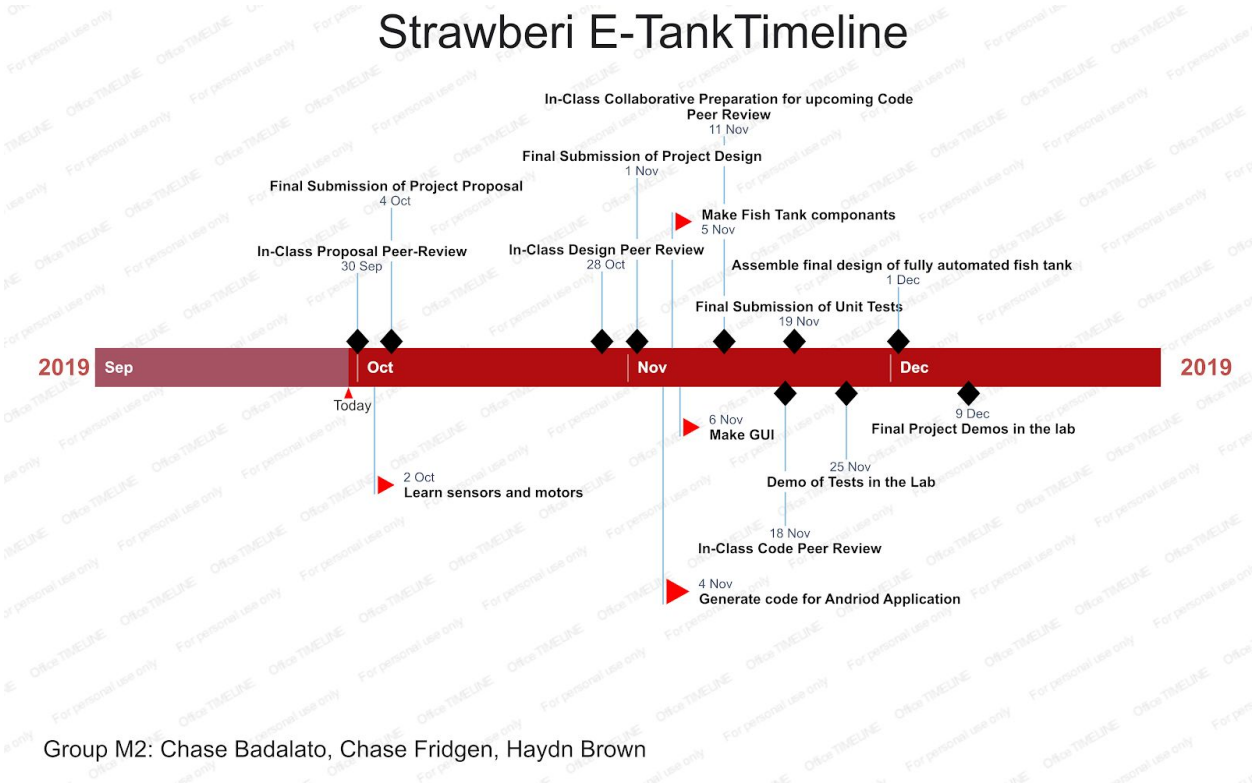
- To test the application that fetches weather data from a website, we can ping the weather website that we choose to make sure that the Rpi is receiving data packets with no loss

- To test the Pi's connection to the android app, a simply feature can be implemented to send a message from the app and have the Pi output a message to a monitor
- Tests for the connectivity between the other components will be done as a whole system test, as they require the arduino and other Pi

Testing the whole system:

- To test the connectivity between the headless Pi and arduino, a simple command can be sent to blink an LED
- To test the connectivity between the headless Pi and the Pi with the GUI, a message can be sent which will be displayed on the GUI
- To test that sensor data can be captured and stored, a single sensor such as the temperature sensor will be used to capture input, send to the headless Pi, then sent to the next Pi which will store the information in the database
- To test that the sensors can be controlled through the app, a feature will be implemented to control the motors. The app will connect to the headless Pi which will relay a command to the arduino the move the actuators

Timeline:



Group M2: Chase Badalato, Chase Fridgen, Haydn Brown