iPot - T11

Aaron Vuong - 101023182
Edmond Chow - 100883365
Kevin Johnson - 101077070
Zubaer Ahmed - 101001925

Motivation

Growing plants is often seen as a very rewarding process. By giving care and nurture to your plants you are rewarded by seeing them bloom. Although this is quite gratifying it is a process that can take months of consistent and diligent care, which interferes with most people's busy lives. Many individuals have to worry about work, school, kids, and other unexpected events that could come up throughout their lives. Day to day activities are often not structured, as they are often changing to meet goals and obstacles for ones day.

Perhaps a student has to study all day at the library or maybe a mother has to take care of a sick child. It is quite easy to forget about caring for ones plants when other responsibilities take priority. Even with enough time to water plants, being able to garden is a talent that must be earned through hard work, patience and dedicated learning [1].

There are multiple factors that go into growing plants and accounting for each one while also navigating a busy life is too much for the average person.

Objective

The objective of this project is to develop a system that enables the general population to effortlessly grow and sustain their plants. Our objective will be achieved through the use of software and hardware capabilities.

Goals

With the use of an iPot growing a plant can become quite easy. The iPot will be able to water plants based on specific hydration needs of said plant, give optimal amounts of sunlight and monitor the temperature required to grow. The installation process requires the user to input the most optimal sunlight, temperature and moisture level into the app and this will optimize the plant growth. With this the plant will be able to grow and thrive with minimal help from the user to make growing flowers, herbs and vegetables easy and fun.

Scope

Plant profiles are the optimal solution to growing plants. Plant profiles however, require a substantial amount of knowledge of various plants which runs into databasing problems. In order to provide the best results, we realized that we would allow the user to predefine the benchmarks for the temperature, moisture and solar intake. Individuals will therefore be able to grow any plant of their liking by customizing the iPot to their plants.

Design Solution

The integrity of the system requires that all components communicate and operate in the intended manner that will be configured. *Figure 1* below provides a breakdown of the various components and peripherals that are relevant to the iPot.

The PI which is labelled as the *Back End* will control the overarching communications going on between the database and the Arduino via wired UDP and GPIO respectively. The database will be installed on the PI labelled *Front End* , which will also handle communication to the mobile app via wireless UDP. The Arduino will handle the flow of data from the various sensors. As seen from Figure 1, the sensor data that circulate the system will from the Moisture sensor, Temperature sensor and Solar Light sensor. The sensors will provide various data to the system and will be connected to the Arduino via GPIO. The watering system that exists is connected similarly to the Arduino via GPIO. The watering system will be activated when certain user inputted benchmarks are reached by the system.

The *Front End* Pi is responsible for the database. The database will have two tables. The first table will store the various records of sensor data that is captured from the system. At varying times during the runtime of the system, the sensor data will create records on the database table for moisture, sunlight and temperature. The second table on the database will store the user inputted benchmarks for the moisture, sunlight and temperature.

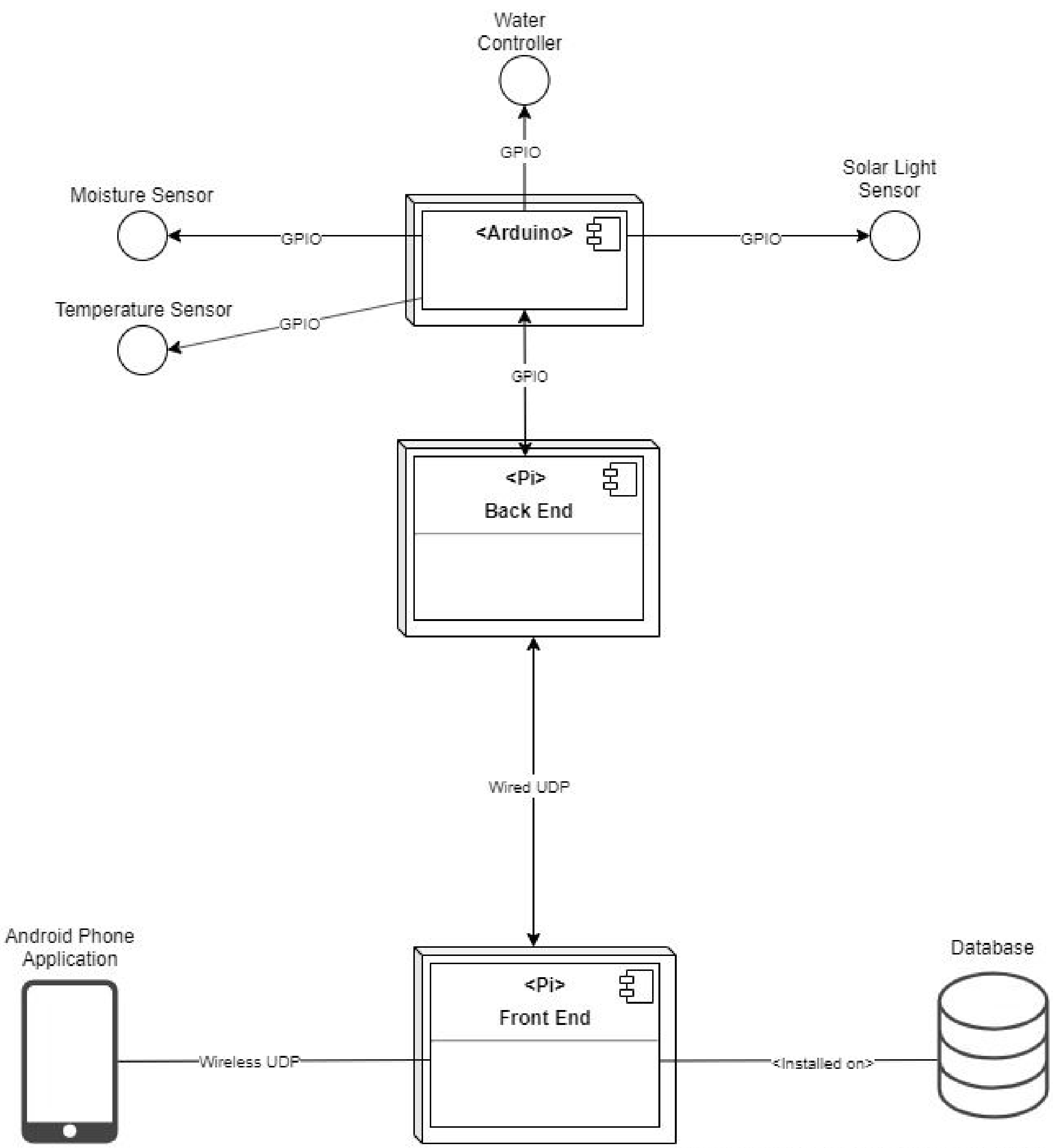


Figure 1: UML Diagram of System Breakdown

The diagram below (*Figure 2*) shows the communication between each component in the system when the moisture sensor detects that the plant is to be watered. Similarly to this diagram displaying the transfer of moisture data, the other sensors that will be providing temperature and solar lighting data will operate in the same manner. The moisture sensor communicated to the Arduino the information that there is not enough water (moisture) present in the iPot. This is relayed to the *Back End Pi* which gives this captured data to the *Front End* Pi to log into the database. Once the new moisture data has been compared to the benchmark entries supplied by the user, we see that the moisture levels are significantly lower. Due to this check that occurs at the Back End Pi we have determined that the iPot is to be watered. The request to water the plant is communicated to the Arduino which will water the plant through the use of the Hose system.

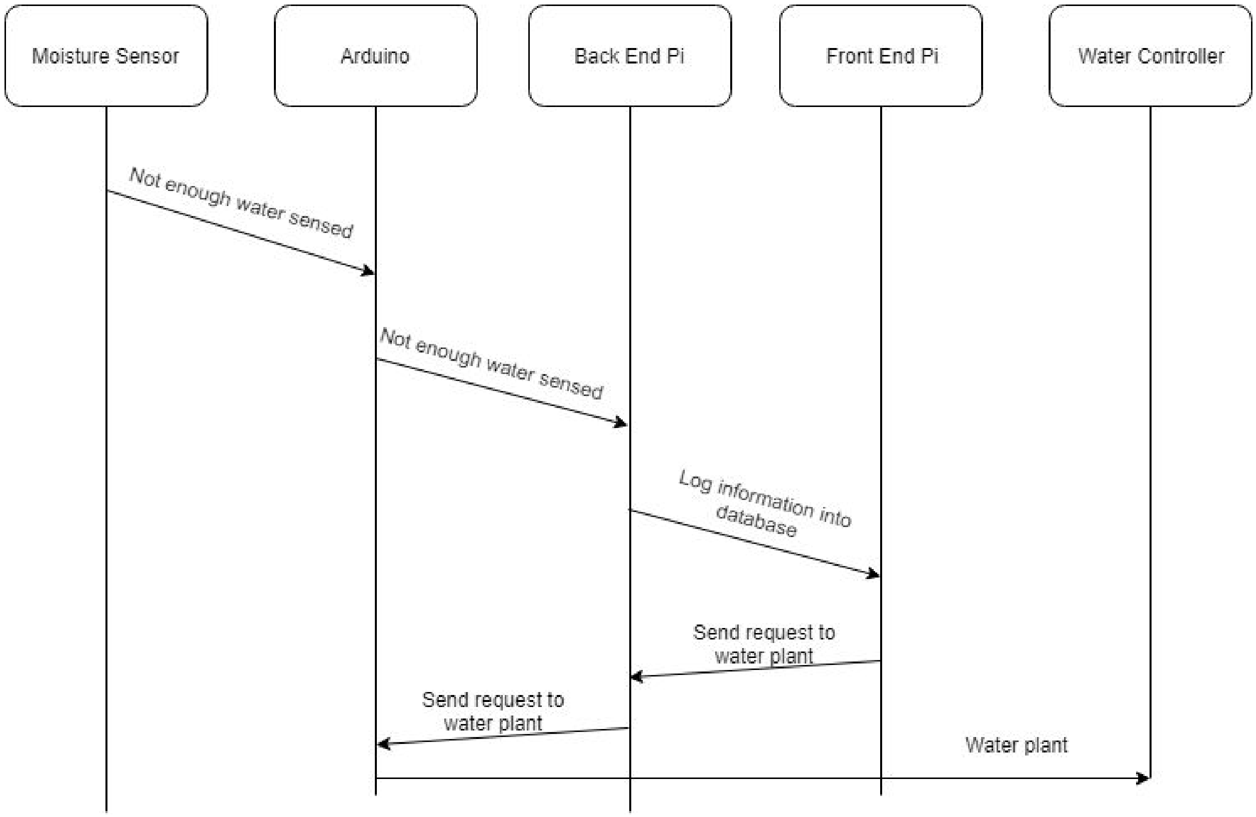


Figure 2: Sequence diagram showing the plant being watered

Equipment

Rasberry Pi x 2	Arduino x 1	Moisture Sensor x 1
Solar Sensor x 1	Temperature Sensor x 1	Watering Hose
Plant (Pot, Soil, Seed)		

Testing

All components for the iPot will be tested both separately (unit tested) and as a whole (integration tested).

<u>Pot equipment</u>

1. Verify the accuracy of the soil humidity sensor installed in the pot to a tolerance of plus or minus 5%
2. Measure the temperature of the air around the plant to within 4 degrees celsius
3. That the accuracy of lux recorded by the solar sensor be accurate within 5 units
4. Determine that water, temperature and light can be controlled by the arduino
5. Using feedback from the sensors set the arduino to automatically adjust the control devices on the pot to maintain conditions of the pot to a pre programmed set-point so that the values measured never deviate more than 5% from the set point values in average room conditions

<u>Database</u>

1. Create a relational database that can store information about the set-points for a pot and its sensor data. The database in its current form should be able to hold the data of one pot and store up to the past 300 sensor logs from the pot
2. Be able to receive data from the android app such as set points and labels and update the corresponding entries in the database accordingly promptly taking no more than 1 minute to do so.
3. Database should be able to send the data from the table containing the sensor data concerning any given pot out over UDP

<u>Android App</u>

1. Create an app that allows the user to create a profile for a pot including name, and creating setpoints for soil humidity, and temperature for the pot.
2. App should be able to send changes to the label and the setpoints of the pot to the database and the backend PI using UDP

<u>Integrated Testing</u>

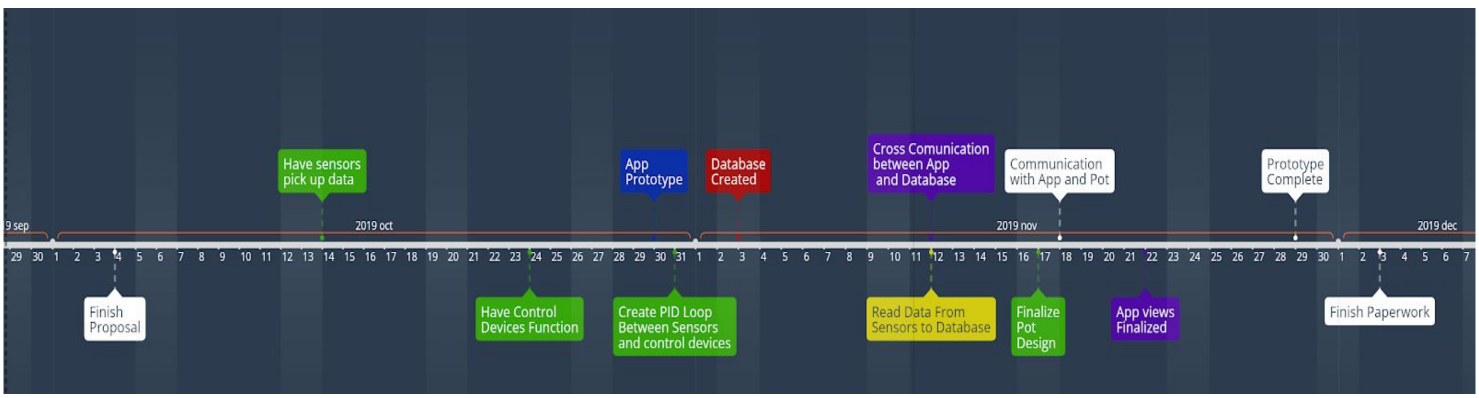
1. Be able to send sensor data collected by the pot to the database using UDP at regular intervals determined by settings determined using the android app. The time from data transfer to logging it into the database should take no more than one minute
2. Can update the set-point values for the pot using the android app, and have the pot begin moving conditions towards the new set point in no more than one minute
3. The app should be able to access the database and pull the information in tables regarding the sensor data for any given pot attached to their account and display it in both a graphical format and a numeric format. This process should take less than 2 minutes from when the request is first sent by the app.

Roles

As seen below in *Table 2: Task breakdowns*, various responsibilities are defined which takes into account the major aspects of this engineering project.

Responsibility	Team Member
Database	Aaron Vuong
Application	Zubaer Ahmed
Hardware/Sensors	Kevin Johnson
Communication	Edmond Chow

Milestones / Project Timeline



References

1. A. Overwatering, "Are You Sure that Plant Needs Water? 5 Signs of Overwatering | Teleflora Blog", *Teleflora Blog*, 2019. [Online]. Available: https://www.teleflora.com/blog/are-you-sure-that-plant-needs-water-5-signs-of-overwatering/. [Accessed: 19- Sep- 2019].