# Implementation

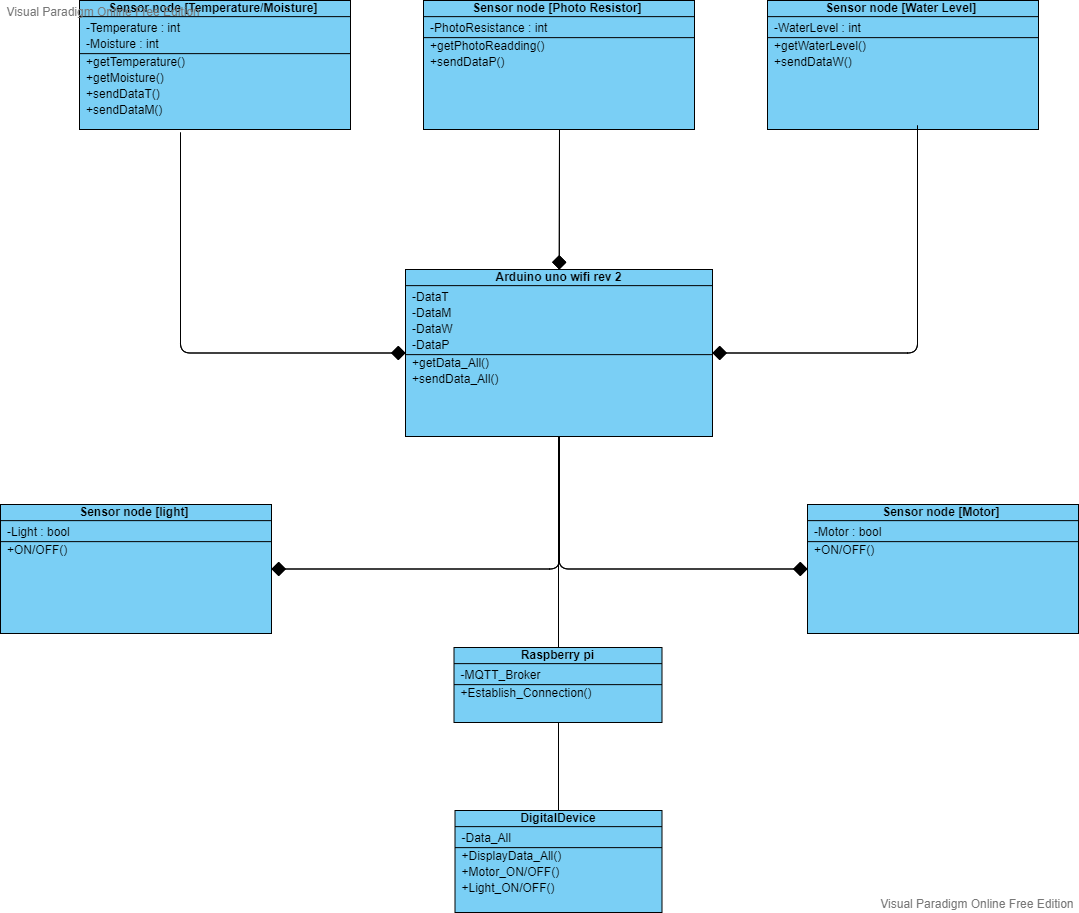
*Describe the static structure of the environment.*

*Provide a class diagram for this purpose and briefly explain the classes or modules.*

*Describe the use case(s) of your environment*

To describe the implementation of the project, first we will explain the structure or our project. It is constructed around 4 key elements :

* Sensor node [Publisher] : A node that consist of a Sensor such as humidity sensor that is connected to an arduino wifi rev 2. This node sends information to the server.
* Sensor node [Subscriber] : A node that consist of a Sensor such as Motor sensor that is connected to an arduino wifi rev 2. This node receives information for the Server.
* Raspberry PI [MQTT broker] : A raspberry pi is used to act as the mqtt broker that serves the connection between the publisher and the subscriber.
* Digital device [ Subscriber/Publisher] : A digital device such as mobile phone or a laptop that access to the web services. This device would be used to receive the information from the Sensor nodes and will let the user preform some action based on that information.

The following is the class diagram for our Implementation. 

We started our implementation by working on the Sensor nodes. The team decided that it would be best to work on an individual sensor node that could publish the data and then integrate the other nodes to our project.

We stared with installing and Arduino IDE and setup the arduino wifi rev 2 module. Once the setup was complete we used DHT11 sensor to send data to a web server. we used several libraries suchs as WiFiNINA, SPI, DHT. We also made "Arduino\_secrets.h" that would store the SSID and PASS for the network, thus making it more modular. At this point of development we were not using the Raspberry pi. The implemented code can be found in the appendix.

The next step of our project was to set up the Rasberry pi as the MQTT broker. For this we started with installing VNC viewer. VNC is free remote desktop application that helps you access a remote machine ( in our case Raspberry pi) and control it with existing hardware of your system. Once we set up our environment, we started with our development process by installing MOSQUITTO Broker on the Raspberry pi to suppor the MQTT protocol. Mosquitto is a message broker. Once this setup is complete, one can used the following line of code to Publish or Subscribe. To publish : “mosquitto -d mosquitto\_pub -d -t testTopic -m” and to subscribe : “Mosquitto\_sub -d -t testTopic -m”.

It is to be noted that we used our wifi hotspot to host the server for our project.

Our application could potentially use all digital platforms thnat have access to web services but we as a team decided to go with phone application to use it as a publisher/subscriber.

We are using test.mosquitto.org It hosts a publicly available Eclipse Mosquitto MQTT server/broker. With this we configured our application such that it could receive the data from the publisher nodes and send the data to the subscriber nodes. The implementation of this as well as some pictures of our hardware simulation can be found can be found in appendix.

We then proceeded to set up the subscriber node. We Initially decide to start with setting up the Motor sensor node to show the implementation but due to lack of compatible hardware and time, we decided to used a LED that could be turned ON/OFF representing the lighting system of our greenhouse prototype. With this we completed our project implementation. The following section describes the different use cases of our project.