SOFE4610 - Final Project Report

Dec 6, 2022

Name	Student ID
Manreet Kaur	**100766207**
Ivan Bisol	**100701735**
Gutu Shiferaw	**100767090**

Table of Contents

Brief description of the application

dropTemp aims to enter the market of smart hygro/thermometers while offering an open-source, small form factor alternative at a competitive price. The dropTemp system starts with the edge devices known as “drops” that read an area’s temperature and humidity level using built in sensors. In our initial development/deployment cycle, drops will utilize the OSOYOO basic board and the Temperature & Humidity sensor we were provided with in our learning kits. Further development would likely result in using something akin to the Arduino Nano 33 IoT to reduce cost and form factor even further. These drops will be powered by either battery power or hard-wired with USB or an AC outlet depending on what fits the user’s needs the best.

One or more of these drops will then connect to a central processing unit that can be installed anywhere within the user’s house. Again in our initial development/deployment cycle, this central hub will use the Raspberry Pi 3 we were provided, however further development would aim to implement a cheaper and smaller alternative, possibly akin to the Raspberry Pi Zero 2 W. The central hub will periodically poll the drops for their current readings and report those readings to a cloud processing web server.

Once the web server receives the results of the central hub's information, it will then process and store the reading data into a database that keeps track of the past 21 days of readings from that specific hub. Users will be able to log in to a website from their laptop, PC, or mobile browser to then view and save their data in either .xls or .xml format.

Use Cases

Use Case ID	Description
UC-1	User wants to track temperature and humidity data of certain rooms within their house.
UC-2	User wants to track same stats of certain containers within their house.
UC-3	User wants to view historical data of their sensors to adjust thermostat settings accordingly.
UC-4	User wants to seek out potential discrepancies in climate across their house.

Application logical architecture with descriptions of the components

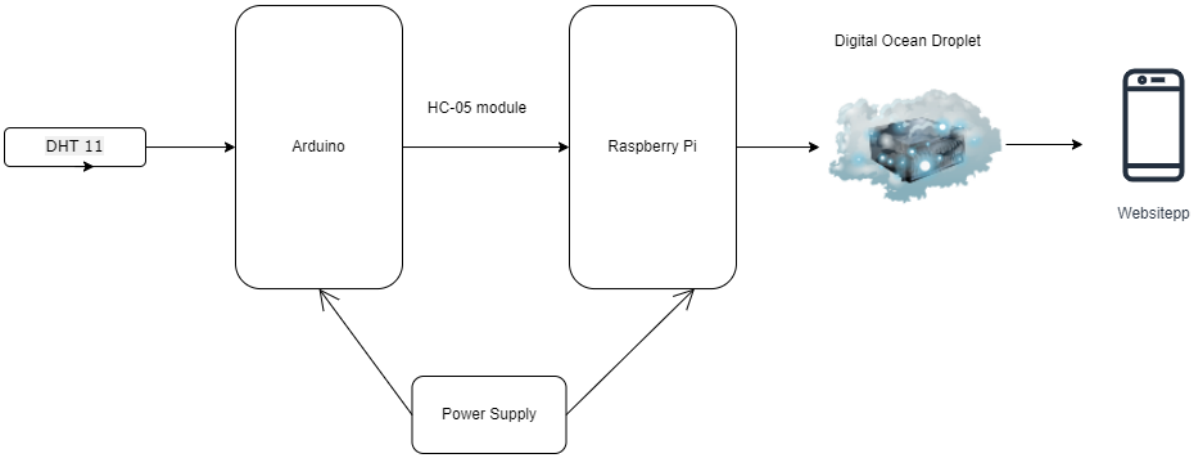

Figure 1: Logical Architecture

Sensing and Power

Component	Justification
DHT11	We decided to utilize the DHT11 as it is a low-cost temperature sensor that provides relatively accurate readings that strikes a good balance between cost and performance.

Data communication

Component	Justification
HC-05	Arduino Uno does not come with in-built bluetooth, hence we are using HC-05 bluetooth module that is acting as a bridge for the connection between the Arduino Uno and Raspberry Pi 4. This bluetooth module can switch modes between receiving and transmitting data.

Edge Computing

Component	Justification
Raspberry Pi 4	The Raspberry Pi is utilized as our far edge computer to collect and compile data received from the sensing device before periodically transmitting it to our Droplet for further analysis. While other Raspberry models may be more suitable for the task, the development team has decided to utilize the resources given to them.
Arduino Uno	The Arduino Uno is used to collect data using the DHT11 and encode it for transmission over BLE using the HC-05 to the Raspberry Pi. As with the Raspberry Pi, there may be more suitable Arduino models for the task.

Computing and analytics

Component	Justification
DigitalOcean Droplet	We chose to use DigitalOcean for cloud computing because that is what the development team is familiar with.

Security

Component	Justification
Log-in portal	The log-in portal would help secure each user's data and personalize the device to our users.

Deployment diagram with an associated description of the deployed platforms and entities deployed

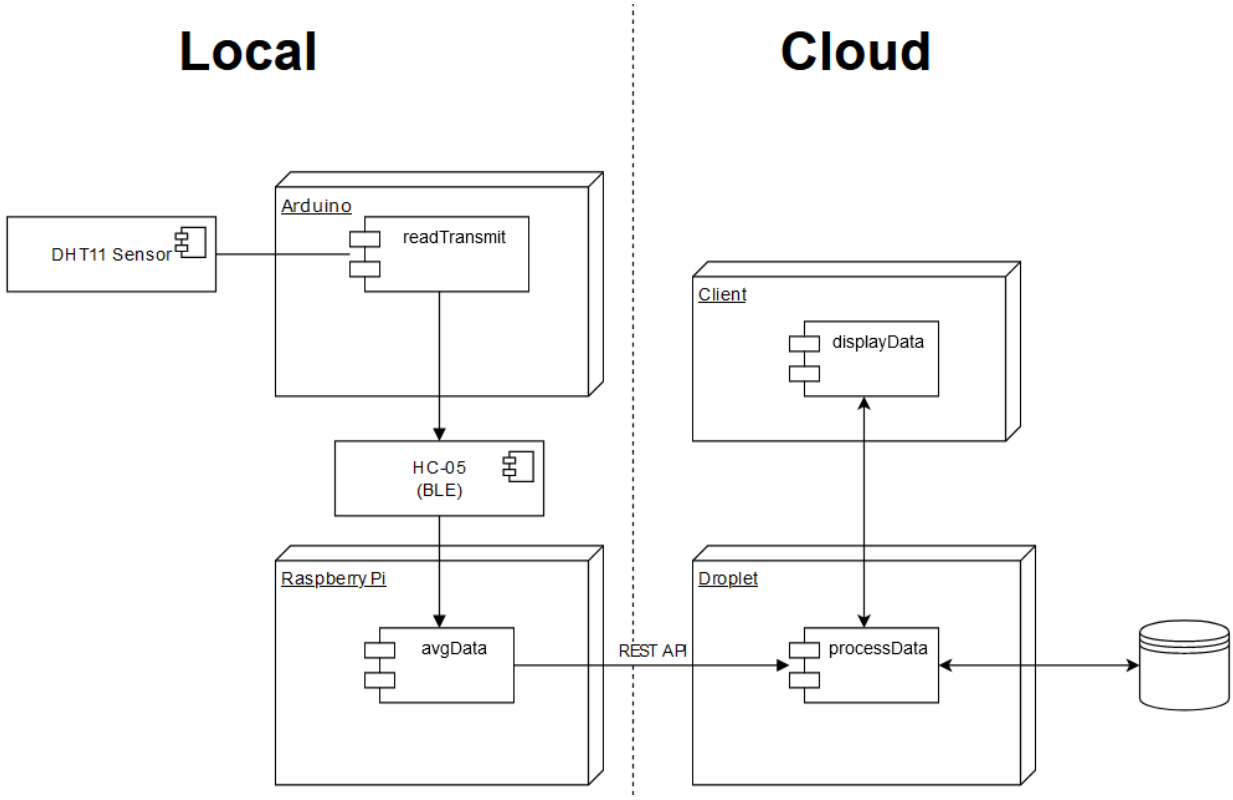

Figure 2: Deployment Diagram

Module	**Purpose**
readTransmit	Arduino periodically (approx. every 2 seconds) reads the temperature and humidity data from the DHT11 before concating them into a string which is then encoded into a char array which is then transmitted over BLE using the HC-05.
avgData	The Raspberry Pi periodically (approx. every 2 seconds as well) reads the next incoming line from the HC-05 and decodes array to receive the temperature and humidity. These values are added to a running total which is then averaged every 30 repetitions (approx. every 61-62 seconds). Averaged values are transmitted to our Digital Ocean Droplet, using a REST endpoint, for further processing.
processData	The goal of processData is to efficiently return data spreads of

Module	Purpose
	temp/humidity data for various timescales (say 10 min, 1 hr, 1 day, and 1 week for now) based on the request passed from displayData.
displayData	Client facing website that allows users to login and present them with a UI to select which values they want reported.

Sequence diagrams that reflect the use cases

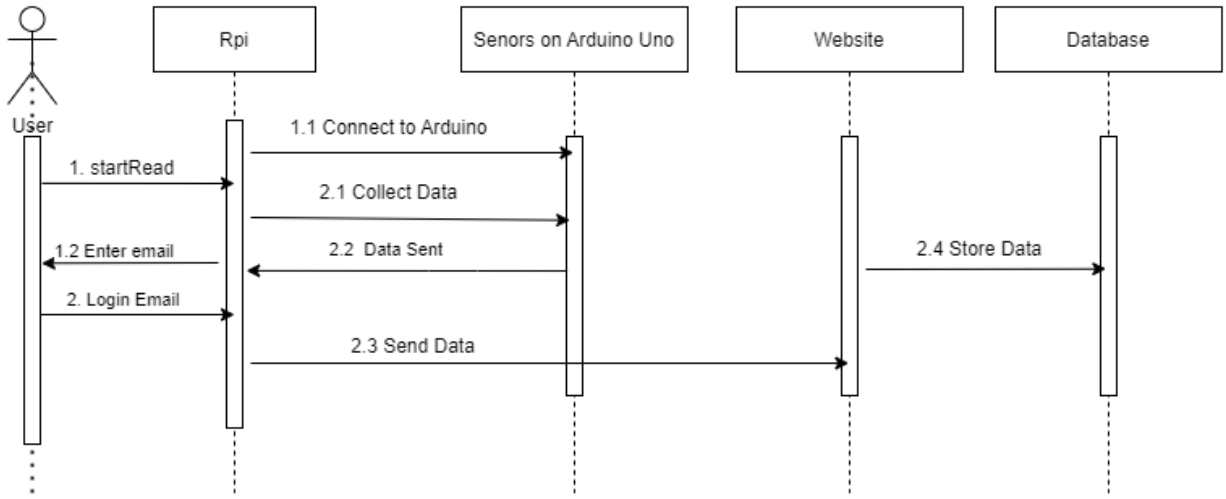

Figure 3: Sequence diagram of Use Case 1

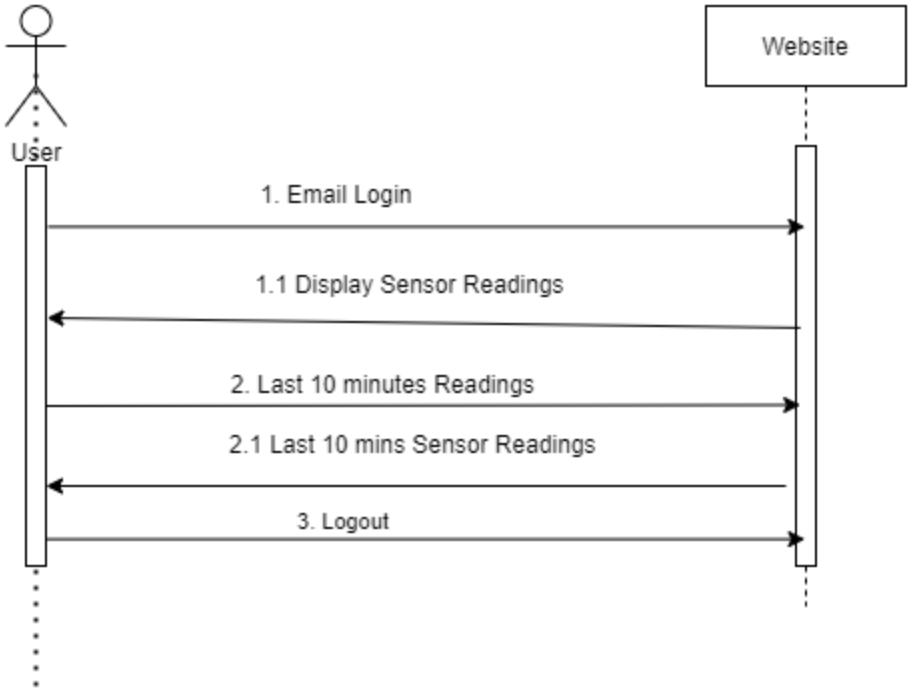

Figure 3: Sequence diagram of Use Case 3

Acceptance Test :

Login in to https://www.droptemp.online/ using email testing@pi.com to view below results :

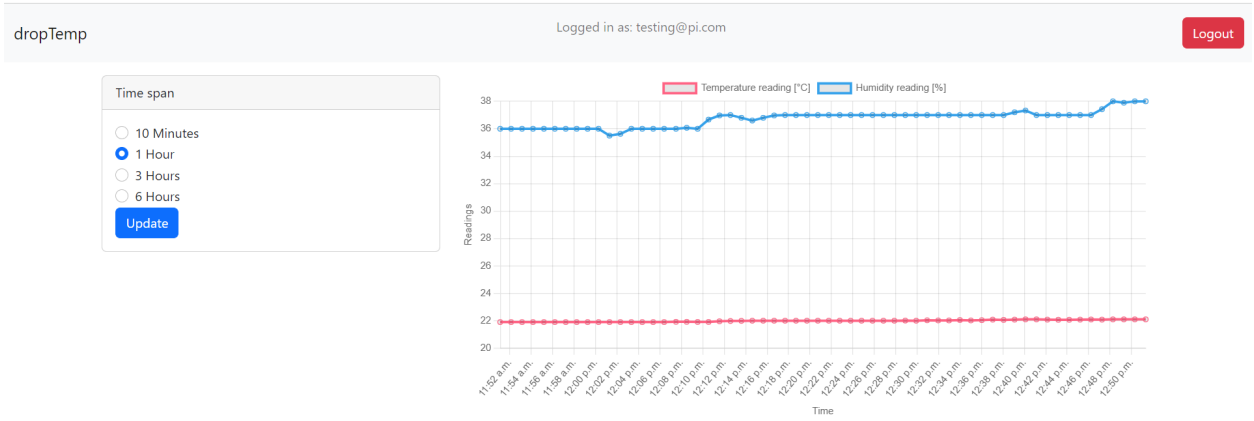

Figure 4: dropTemp Website