

Report Date: 07/01/2022

To: [ematson@purdue.edu](mailto:ematson@purdue.edu), [ahsmith@purdue.edu](mailto:ahsmith@purdue.edu), [lhiday@purdue.edu](mailto:lhiday@purdue.edu), [lee3450@purdue.edu](mailto:lee3450@purdue.edu)

From: IEEE

- Sungjin Park ([huitseize@chungbuk.ac.kr](mailto:huitseize@chungbuk.ac.kr))
- Gayoung Yeom ([gayoung@hufs.ac.kr](mailto:gayoung@hufs.ac.kr))
- Dayeon Won ([aakk9350@kw.ac.kr](mailto:aakk9350@kw.ac.kr))
- Haegyeong Im ([fine632@soongsil.ac.kr](mailto:fine632@soongsil.ac.kr))
- Minji Kim ([minzyk0729@jejunu.ac.kr](mailto:minzyk0729@jejunu.ac.kr))

## Summary

The front-end team applied lots of libraries for functions such as sliders, and graphs. The API of the backend server was modified as the sensor of the weather station was changed. The connection between the back-end server and the front-end web server was made. The back-end and front-end servers were dockerized. After that dockerizing, kubernetes cluster was created on GCP(Google Cloud Platform). The network team completed testing weather sensors and started to requesting API to Open Weather map for gathering real-time data.

## What IEEE completed this week:

- Front-end team
  - Applying the graph library called Apexcharts with dummy data
  - Customizing active marker in map component
    - It was made using icons that are presented data types.
  - Setting the location and footer section
  - Creating setting component
    - Setting component means modal which can set the range of data value and it was made applying the slider library called rc-slider.
  - Adding the toggle button which can show the slider of graphs
    - Using the react-slick, the graphs can be presented by two types.
      - One is a slide and the other is a collective view.
  - Modularization Axios library for connecting Backend server
  - Testing connecting API using customized HTTP modules
  - Creating Dockerfile
    - Building the image and running on [localhost](#) for a test.
    - Pushing the image in Docker Hub to make it everyone can access it.
- Back-end team
  - Developing the back-end server
    - The network between the back-end server and front-end was connected.
    - API which retrieves stations by using zip codes was developed.
    - Zip code was added to the station's data structure.
    - The entity of the sensors was changed as the sensors were changed.
  - Testing API with the front-end web server
    - CORS error was solved and the front-end server was connected successfully.
  - Dockerizing the back-end server
    - Dockerfile was created.
    - The back-end server was dockerized.
- Kubernetes team
  - Building kubernetes cluster environment
    - Backend kubernetes pools and pods were created.

- Frontend kubernetes pools and pods were created.
  - A SQL instance was created. But, Now There are issues with the connection between Backend nodes and a SQL instance.
- Network team
  - Trying to request API from Open weather map server
    - The original plan was to use the Weather Cloud platform to collect weather data.
    - However, the Open weather map which can provide free weather information API more easily was found and selected for getting real-time weather data.
    - Therefore, requesting API has been tried this week using ESP32 with Arduino.
  - Testing UV sensor
    - According to the last team meeting, the fact that the weather cloud platform can not provide soil moisture and soil temperature was checked.
    - In order to set a data set in the same condition, the type of sensor was changed. The changed sensor are as follows
      - The wind direction sensor was not necessary, so it was removed.
      - Instead, the UV light sensor has been added.
      - The soil temperature sensor and Soil moisture sensor were replaced with just temperature and humidity.
  - Transferring Arduino sensor code to ESP LoRa 32
    - The structure of the Arduino board and the ESP board is different, so the code should be changed.

### **Things to do by next week**

- Front-end team
  - Connecting all APIs
  - Containerizing after testing with APIs
  - Refactoring some components such as Map and Graph
- Back-end team
  - The environmental values will be injected by using Dockerfile.
  - The back-end server will be run on the Google Cloud Platform(GCP).
- Kubernetes team
  - Solving the issue with the connection between backend nodes and a SQL instance.
  - Setting detailed values in kubernetes deployment.
- Network team
  - The real weather station will be operated in front of K-square front yard.
  - Getting real-time API from open weather map to ESP32
  - The Weather station data and real-time data will be sent to the LoRa gateway.

### **Problems or challenges:**

- Changing the data type of receiving from the weather cloud
  - Soil temperature, soil moisture, wind speed, UV test were finished as a real sensor, but the data type that can be received from the weather cloud does not have a soil temperature, soil moisture, and change to temperature, and humidity.
- CORS error between the back-end server and the front-end web server occurred.
  - CORS error occurred when the domain between servers is different.
  - The origin of the front-end web server was added to CORS policy in the back-end server.

## References