SHAMBATECH (SMART FARM MONITORING SYSTEM)

BY

WEREHIRE ANDRE KURIA	**I39/82623/2017**
OCHIENG' DAVID WAMBIA	**I39/82632/2017**
WANJOHI PAUL MUTHOMI	**I39/85829/2017**

A Project report submitted for examination in partial fulfilment of the requirements for Award of the Degree of Bachelor of Science in Microprocessor Technology and Instrumentation of the University of Nairobi.

2021

DECLARATION

We hereby declare that this project is our original work and has not been submitted elsewhere for examination, award of a degree or publication. Where other people's work has been used, this has properly been acknowledged and referenced in accordance with the University of Nairobi's requirements.

Werehire Andre Kuria
I39/82623/2017
Department of Physics
University of Nairobi

Signature.....
Date.....

Ochieng' David Wambia
I39/82632/2017
Department of Physics
University of Nairobi

Signature.....
Date.....

Paul Muthomi
I39/85829/2017sss
Department of Physics
University of Nairobi

Signature.....
Date.....

Mr. Mjomba A C Kale
Project Supervisor
Department of Physics
University of Nairobi

Signature.....
Date.....

TABLE OF CONTENTS

LISTS OF TABLES, FIGURES, ABBREVIATIONS AND SYMBOLS

Table of figures

List of tables

List of abbreviations

CU - Control unit
LoRa - Long Range module
IoT- Internet of things
GDP- Gross domestic product
SPI -Serial peripheral interface
SISO - Single Input Single Output
SIMO - Single Input Multiple output
MISO - Multiple Input Single Output
MIMO - Multiple Input multiple Output
SCK- serial clock
NSS -network and switching subsystem
GND - Ground
RSSI - Received signal strength indication
JSON - JavaScript Object Notation
API - Application Programmable Interface

ABSTRACT

The smart farm monitor is system of sensors that comprises of a combination of sensors, microprocessors and an IoT server to collect data from a farm, analyse and give feedback to the farmer through a user interface. The system also includes a disease detection system and an automated response system.

The project was divided into the following areas:

- The sensors node: comprising of a combination of sensors and a central microprocessor. These sensors include: Temperature sensors, humidity sensors and light intensity sensors. They are connected to the microprocessor that will collect the data and send it to the communication node.
- The communication node: this will facilitate the data transfer from the microprocessor to the cloud. We settled to using LoRa modules which would send data to the thinkspeak IoT server which will facilitate storage and processing of the collected data. From the sensors to the microprocessors, the connection is wired but from the microprocessor to the LoRa module to the thinkspeak server.
  In the server, we plan to data science algorithms that would be used for processing the received data and come up with predictions that would be useful to the farmer for improving his farming methods to better his yield.
- The user interface node: this is the mobile application that will be the gateway between the farmer and the system. From this app, the farmer would receive data collected from the farm in real-time from any location as long as there is internet connectivity to his device. The farmer would also receive analytics summary, do remote operations like activating sprinklers and have a surveillance feed to the farm.
- The automated and remote response node: This comprises of the sprayer nozzles, drip pipes, hosepipes and storage tanks. This system will have two separate tanks, one for water and another for the farm chemical such as pesticide, fungicide or insecticide. Water from the tank will be delivered to the drip pipes and the chemicals from the chemical tanks will be delivered to the sprayers. Flow of liquids from the tanks to their respective delivery systems is controlled by electronic water valves and pumps that are controlled from signals from the microcontroller.
- He disease detection system: this will incorporate use of near infrared spectroscopy where the pants and produce would be scanned and analysed for diseases. This would be a tertiary addition to tis project since its quite a complex system that needs to developed intricately with high precision.

Due to the circumstances brought by the coronavirus pandemic, we settled with coming up with a proof of concept with the workable areas since collaboration was difficult considering for most of the part, we were developing this project from our homes. Some of the functions like the disease detection system, automated and remote response nodes and implementing data science algorithms were put on hold.

Aside from these challenges, we have a working system that collects data, sends it to the IoT server and conveys it back to the farmer through a mobile app of our own design.

INTRODUCTION

Agriculture is the backbone of many African economies but most of the farmers are still using traditional methods of farming. If tackled correctly, the income and GDP from agriculture can be increased by introducing 21st century technologies into the farming process. This integration has been done successively in first world countries like Netherlands and the USA. Compared to countries in Africa, most farmers are small scale farmers who still practice traditional farming methods which doesn't yield much.

The good thing is that many farmers have adopted greenhouses and better irrigation systems in their farms but still we can not compete with other countries that have better technologies and control systems. Since IoT have been introduced in many aspects of our life, such as smart watches and smart home appliances, its suitable to include IoT technology in Agriculture to closely monitor the crop. IoT technology has vastly improved over the last few years so the equipment is cheaper, more durable and the performance capabilities of some systems are off the charts. Integrating these technologies in our farms, especially small scale since majority of African farmers are small scale will be highly beneficial to the farmers and in the end, our economic growth.

PROBLEM STATEMENT

Agriculture is the main economic activity for many people in sub-Saharan Africa and more importantly it is essential for survival of the human species. Most of these countries are developing countries thus farmers still practice crude traditional farming methods. Crop yield is hindered by conditions such as drought, pests, diseases and lack of resources to shift to modern methods of farming that better crop yield and have factors that counteract attacks from pests and diseases. Monitoring of the state of crops is also another problem that farmers face especially in large farms. Timely identification of irregularities in the crop health for instance would prevent spread of a crop disease or pests since the farmer would apply the necessary precautions like fumigating or uprooting of weeds. This timely response is made difficult by physical monitoring by the farmer hence farmers are losing their crops due to late response. Another instance is when a particular section of a farm is not receiving enough water making crops wilt. This is not easily noticeable by use of the human eye hence integrating humidity sensors would change that.

Those are just a few of the difficulties farmers face that would otherwise be fixed by integrating technology into farming in these developing countries.

OBJECTIVES

The main objective is to incorporate IoT technology into small farms and greenhouses. We want to make the farm monitor using cheap readily available components such as sensors which would not be detrimental to the environment and to the cultivated crops.

The tasks we wanted to achieve by the end of this project were:

- Successfully connect sensors for data collection to a central microprocessor or computer.
- To come up with a communication model for transferring data between microprocessors and the think speak IoT server.
- Successfully upload data to the think speak IoT server where it would be readily available to the farmer and other authorised user through any device that internet connection and for easier processing.
- Apply data science techniques and algorithms on the collected data to come up with projections that would be useful to the farmer to come up with better farming practices to better his yield.
- Creating automated methods for easing farm maintenance such as automated watering and application of fertilisers and pesticides.
- Incorporating a farm surveillance system in which the farmer could use to view his crop in real time.
- Incorporating a disease detection system that could ease the farmer in maintaining healthy crop.
- Creating a website and a mobile application in which the farmer can receive real-time information recorded from the farm anywhere. This information will include measurements recorded from the farm, video surveillance and analytics. Remote operations will be included in these interfaces where the farmer can remotely activate crop maintenance mechanisms such as irrigation sprinklers and chemical sprayers.
  The mobile app will also indicate if there are issues in the farm like if there is insufficient soil water or the temperatures are too hot and the recommended actions.

If successful, we would have created a system that would be an assistant to the farmer, lighting up the load so as to concentrate on improving the quality and yield of the crop.

We plan on using small, cheap and lightweight components. Improvement in IoT has made it possible to have the components we require thus the overall cost of the system should be reasonable to the small-scale farmer.

The system would also be scalable to meet the user's need, so that the farmer can set the ideal settings he wants depending on the cultivated crop.

LITERATURE REVIEW

IoT devices nowadays use a variety of various technologies to support their communications, but none of them are really ideal for the purpose and application of today. Wi-Fi is everywhere at the moment but it uses a lot of energy and transmits lots of data, whilst this is great it isn't such a perfect solution for IoT devices that don't have as much energy at their disposal or wish to send small amounts of data. There are also limitations in the modulation techniques used and as such access points can only handle a handful of devices at once.
Bluetooth devices allow local communication but have very limited range in version 4.0. They also require too much power. Even newer Bluetooth Low Energy devices still consume much more power than is necessary. Up to recently the best available technology on the market was considered to be ZigBee low power modules that transmit over greater distances and at low transfer rates, usually a few kilometres in clear path.

LoRa technology was developed by a company called Semtech and it is a new wireless protocol designed specifically for long-range, low-power communications. LoRa stands for Long Range Radio and is mainly targeted for M2M and IoT networks. This technology will enable public or multi-tenant networks to connect a number of applications running on the same network.
LoRa Alliance was formed to standardize LPWAN (Low Power Wide Area Networks) for IoT and is a non-profit association which features membership from a number of key market shareholders such as CISCO, activity, Microchip, IBM, STMicro, SEMTECH, Orange mobile and many more. This alliance is key to providing interoperability among multiple nationwide networks.
Each LoRa gateway has the ability to handle up to millions of nodes. The signals can span a significant distance, which means that there is less infrastructure required, making constructing a network much cheaper and faster to implement.

MATERIALS METHODOLOGY

HARDWARE IMPLEMENTATION

In this section we divide our hardware sections into -

1. The node -this is the component where sensor data would be collected.

2. The gateway – this is where sensor data collected from the nodes is transferred to web server or an IoT server.

For both components we would use LoRa technology to transfer data from the node to the gateway.

We used the SX1276 LoRa chip (image below)

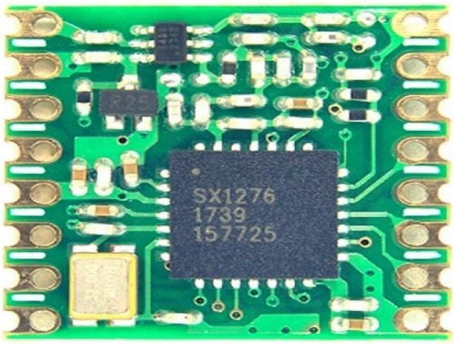


Figure 1: SX1276 LoRa chip

This module has the frequency of 868mhz bounded to it, thus we then needed to create an antenna for this module, thus using the formula:

c = h x f

where:

c – speed of light

h – wavelength

f – frequency

868 MHz band is thus 299.792.458 / 868.000.000 = 34,54 cm. Half of this is 17.27 cm and a quarter is 8.63 cm.

We then used 10-ohm wire cables of length 8.63 cm to have antenna for both the node and the gateway as shown below

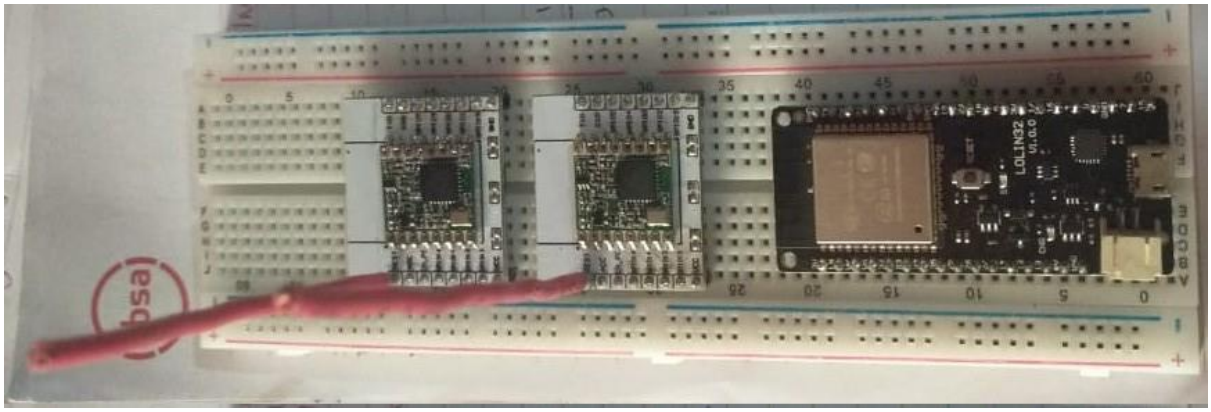


Figure 2: the node and gateway

The node and the gateway both contained the ESP8266 modules for communication with the lora chips which uses SPI protocol for communication between chip to chip.
The following table shows the connections used between the SX1276 and ESP8266 both at the gateway and the node.

SX1276	NodeMCU
MISO	D6
MOSI	D7
SCK	D5
NSS	D8
RESET	D2
DIO0	D1
3.3V	3.3V
GND	GND

Table 1:connections used between the SX1276 and ESP8266 both at the gateway and the node.

Thus, the node circuit would look like:

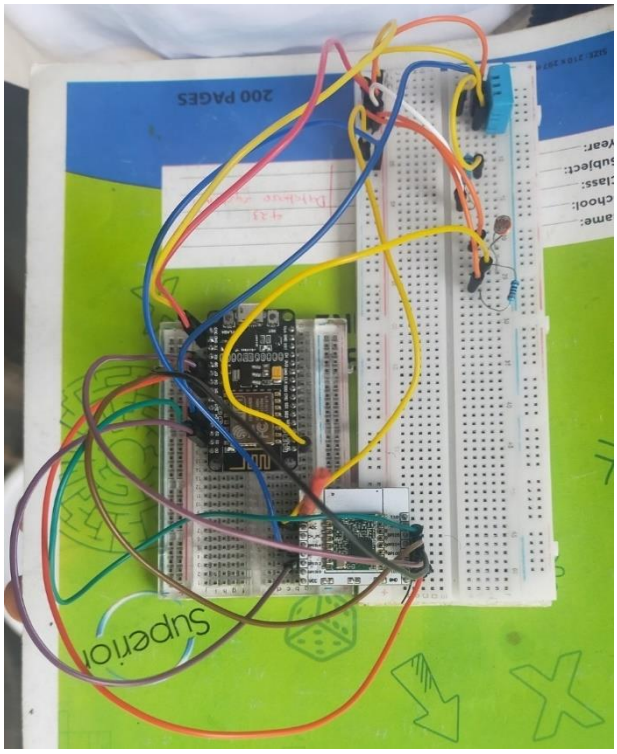


Figure 3: the node circuit

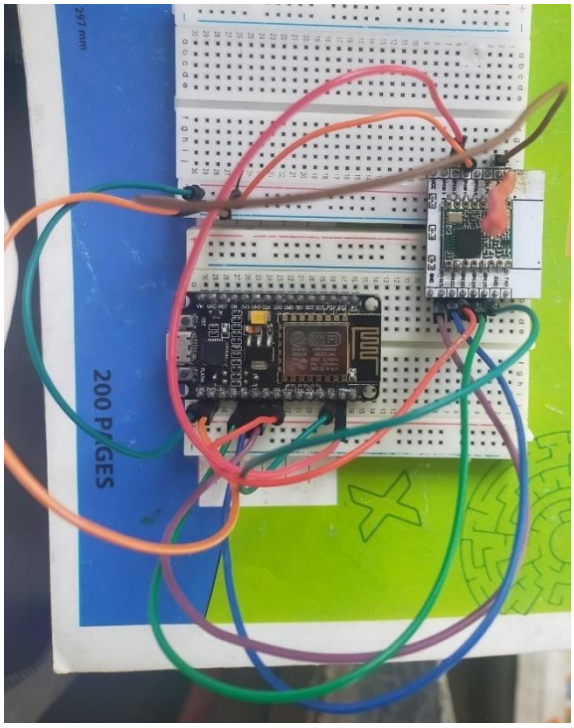


Figure 4: the gateway circuit

The next step was to configure both circuits to communicate with each other. Communication was set that both circuits would do so in half duplex mode. As the node circuit gets the sensor data it sends it to the gateway circuit, the sensor circuit then receives the data and send another packet of data saying it received the data.

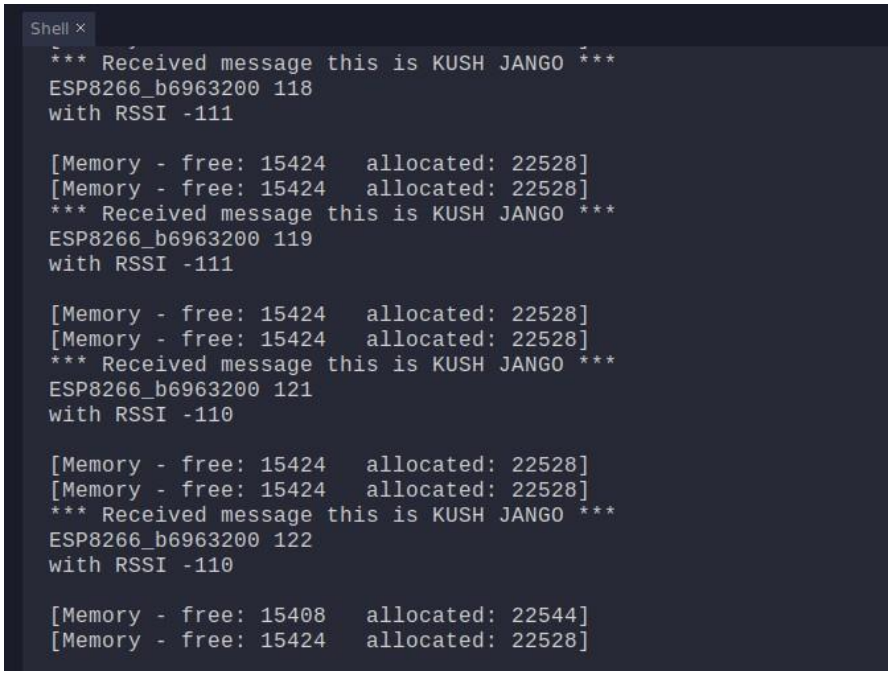


Figure 5: the node shell

The node shell prints out the message received from the gateway, the name of the gateway it has received from, the message count and RSSI which measures the strength of the signal.

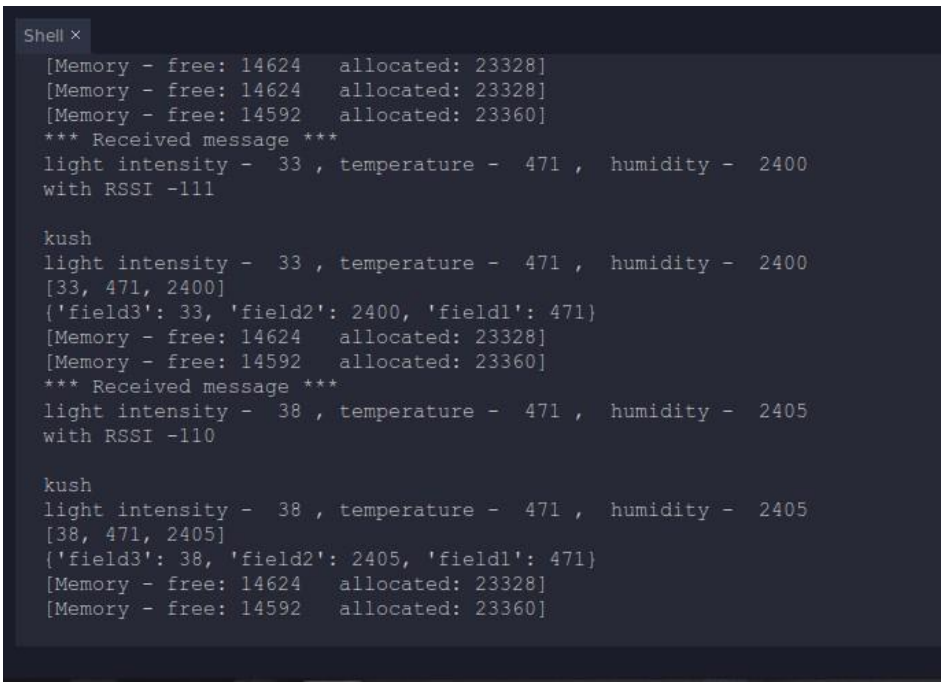


Figure 6: the gateway shell

In the gateway the data packets received include the message that the node received its packets from the node was received, also the sensor data sent from the node circuit, and also the RSSI.

From there on the data now received by the gateway is now sent to the thingspeak server through connections to the internet. This therefore shows that only the gateway should be connected to the internet for sending functions.

Once data is sent to the thingspeak server it is visualized as follows:

Figure 7: Visualisation of the data sent to thinkspeak server

Communication

This unit will be responsible for the communication between the transducers in the farm and the control unit. Since the control unit will be situated indoors or wherever the user chooses, a wireless connection will be required for the transfer of data as it is more efficient cost wise and performance wise too.

We chose to use LoRa Modules for the communication due to their long range and small size.

Communication between the transducers(sensors) and the control unit - key to prevent having to manually collect the data from the sensors. The communication between these sensors and the control unit would be used to:

Collect physiological data (temperature, humidity)-This would come from the sensors to the CU.

Control regulators of the physical environment for example irrigation systems (made possible by the half duplex mode of the LoRa module).

This would help to implement necessary actions after data has been collected by the control unit from the sensors. For instance, for soil humidity levels way below the optimal conditions, the signal sent back to the farm triggers the turning on of the irrigation system in place.

The control of these external systems (Irrigation, temperature) is represented on our prototype by turning on and off of an LED from the control unit to the sensor circuit.

As the mechanism behind the working of the LoRa module has been covered in the literature review of this report, we now look at the data analytics section. After conducting research, we opted to use the website thingspeak as our analytics platform of choice. According to their official website, ThingSpeak™ is an IoT analytics platform service that allows you to aggregate, visualize and analyse live data streams in the cloud. This platform acts as our cloud service, hosting all data sent to it from the farm hence it helps the farmer receive data from the sensors in the farm in real time so any actions that need to be taken by the user are executed without delay, hence improving the crop health and bettering overall yield.

For ease of use, the user would not have to log in to the thingspeak website and check the data from there. That is solved by the phone application that integrates analytics, surveillance and any other remote farm tasks that the user would want to execute like turning on the tap for instance.

Communication between the CU and the ESP8266 microcontroller for the transmission of data is made possible by a python script stored in the microcontroller. The script basically automates the whole process of connecting first to the wireless network, storing the API key for Thing speak where all sensor data will be uploaded to and setting an interval of when to upload the data to thingspeak.

A snapshot of the script:

```
#!/bin/env
import network
import machine
import dht
import time
import urequests
import urandom
from machine import Pin , ADC

import esp
esp.osdebug(None)

import gc
gc.collect()

ssid = 'Enter your ssid'
password = 'enter password'

api_key = '64H59P3ABQ6IW6QW' #api key

station = network.WLAN(network.STA_IF) #connection

station.active(True)
station.connect(ssid, password)

def randint(min, max):
    span = max - min + 1
    div = 0x3fffffff // span
    offset = urandom.getrandbits(30) // div
    val = min + offset
    return val



while station.isconnected() == False:
  pass

print('Connection successful')
print(station.ifconfig())

dh = dht.DHT22(machine.Pin(14)) #sensor pin
ldr = ADC(0) # ldr is analogue config
```

Figure 8: a section of the communication script.

```
dh = dht.DHT22(machine.Pin(14)) #sensor pin
ldr = ADC(0) # ldr is analogue config


while True:

    dh.measure()
    time.sleep(10) #wait 10 seconds to update
    tem = dh.temperature() + randint(0 , 10) # have changing temp values
    hum = dh.humidity() + randint(0 , 25)
    light = ldr.read()
    #print(hum , tem , light)

    sensor_readings = {'field1':tem, 'field2':hum , 'field3':light} #the
    print(sensor_readings)

    request_headers = {'Content-Type': 'application/json'}

    request = urequests.post('http://api.thingspeak.com/update?api_key='
    #print(request.text)
    request.close()

```

Figure 9: continuation of the communication script

Once the data is sent to thingspeak, it is now available for analysis, for use by the android application and any other integral actions.

SOFTWARE IMPLEMENTATION

Android Application

The purpose for the android application for sending and receiving data between the farmer and the farm sensors. Since we were able to implement 3 sensors, temperature, humidity and light intensity, we developed and the android application that will receive this data and display it to the farmer. The Interpreted development environment used was Android Studio.

First, we listed the activities we want for the application, even if we were successful implementing the whole of the project on the sensors section. The main user interface activities are:

- Home/main activity
- Sensor's activity
- Surveillance activity
- Remote farm operations activity.

Since the sensors part of the project is the main focus and is the area, we were able to implement with successful results, we decided to concentrate more on receiving the sensor data and displaying the data to the farmer. The sensor information that will be displayed in the app is: date and time of measurement, the actual measurement value and the entry ID of the recording.

The initial stage of developing the app was creating user interfaces. All the interface/activities were created using XML as shown below.

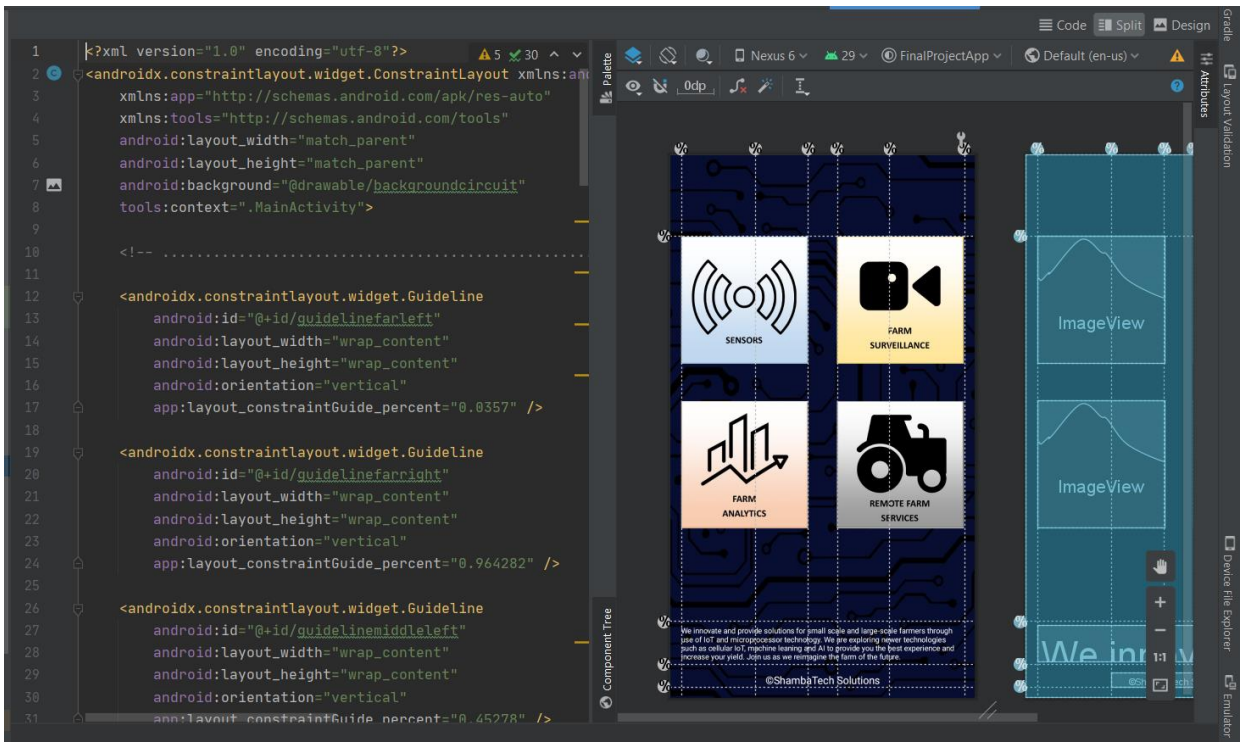


Figure 10: XML code and layout for the home activity

The Home activity was just 4 icons that navigate to the other activities. The icons are pictures that are clickable. These images were created using Microsoft publisher.

The main focus was the sensors activity. It was subdivided into three sections, each displaying its respective sensor information; temperature, humidity and light intensity. Swipe refresh was used for retrieving and refreshing data from the data source. The layout snippet for the sensor activity:

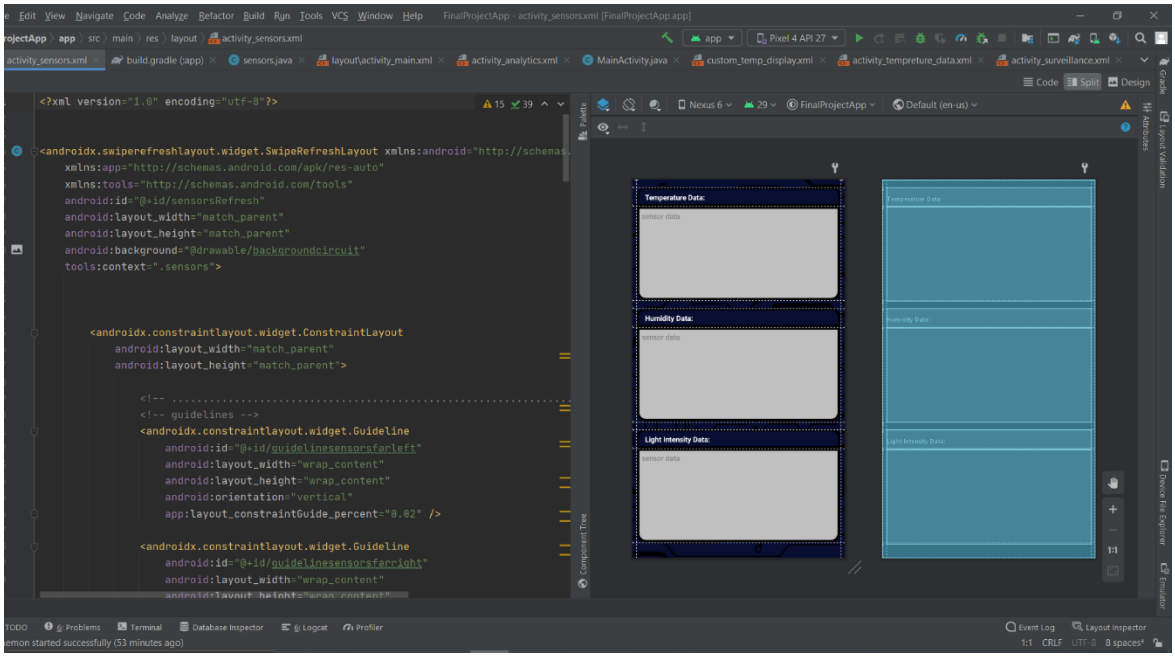
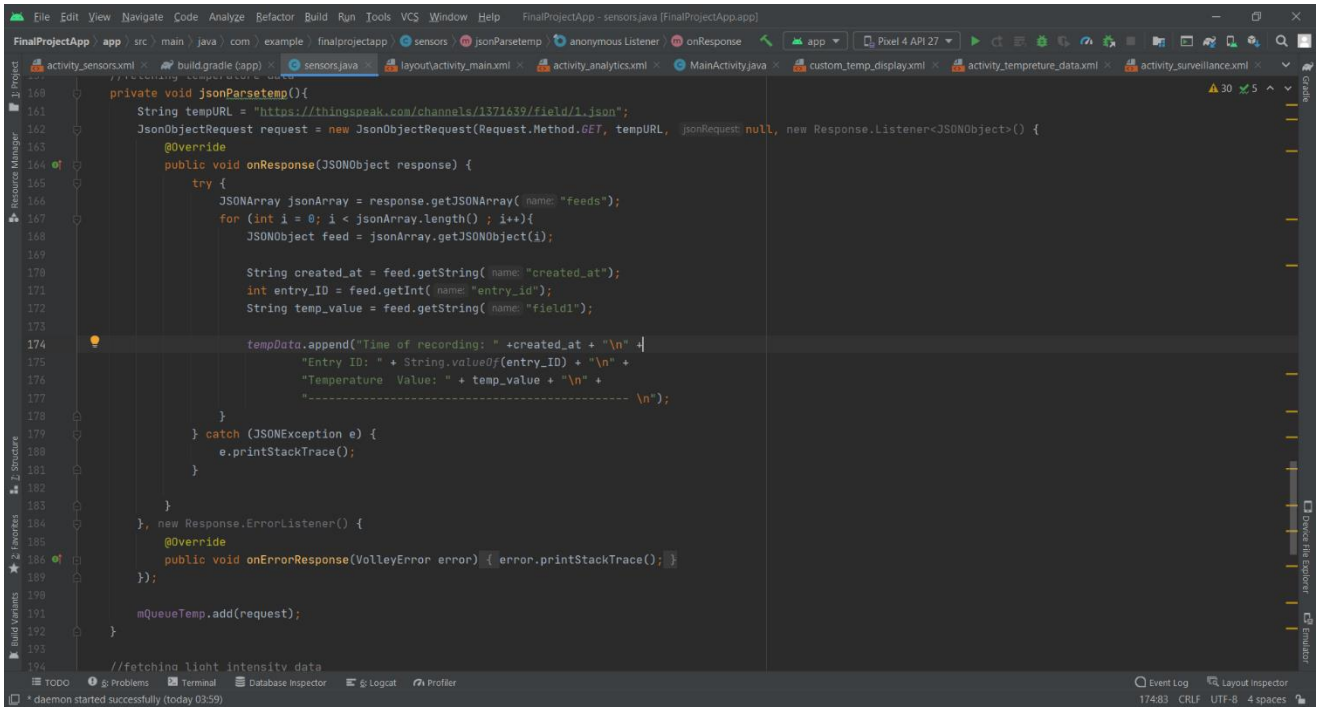


Figure 11: XML code and layout for the sensor's activity

Java classes for joining and adding capabilities to the user interface objects. The scrolling feature of the sensor display containers was also implemented using Java instead of a scrolling activity. Each user interface activity has its own java class for user ingterface implementation.

The most important function was receiving, displaying and refreshing data from the sensors. The measurements from the sensors are sent to the ThinkSpeak server from where the app will retrieve the information from. The code for data communication is in the sensors.java file. JSON API intergration was used for retrieveing data from thinkspeak IoT server since its easier and lightwight. The tempreture data retrieval code is shown in the next page.

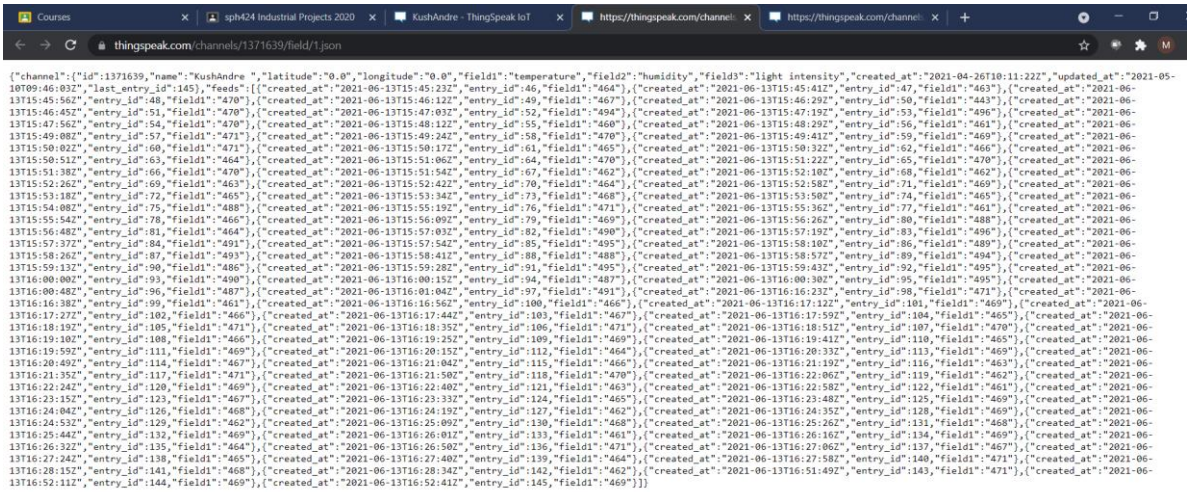


```
private void jsonParsetemp(){
    String tempURL = "https://thingspeak.com/channels/1371639/field/1.json";
    JsonObjectRequest request = new JsonObjectRequest(Request.Method.GET, tempURL, jsonRequest: null, new Response.Listener<JSONObject>() {
        @Override
        public void onResponse(JSONObject response) {
            try {
                JSONArray jsonArray = response.getJSONArray( name: "feeds");
                for (int i = 0; i < jsonArray.length() ; i++){
                    JSONObject feed = jsonArray.getJSONObject(i);

                    String created_at = feed.getString( name: "created_at");
                    int entry_ID = feed.getInt( name: "entry_id");
                    String temp_value = feed.getString( name: "field1");

                    tempData.append("Time of recording: " +created_at + "\n" +
                            "Entry ID: " + String.valueOf(entry_ID) + "\n" +
                            "Temperature  Value: " + temp_value + "\n" +
                            "---------------------------------------------- \n");
                }
            } catch (JSONException e) {
                e.printStackTrace();
            }

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) { error.printStackTrace(); }
    });

    mQueueTemp.add(request);
}

//fetching light intensity data
```

Figure 12: a section of the Java code for sensors activity showing the temperature data retrieval using JSON parsing

The JSON code that has the data is given by the thinkspeak platform. The code will read the URL on which the JSON code is located and parse the data contained in it then display it in the respective container in the activity.

A sample of the JSON code:




```
{"channel":{"id":1371639,"name":"KushAndre ","latitude":"0.0","longitude":"0.0","field1":"temperature","field2":"humidity","field3":"light intensity","created_at":"2021-04-26T10:11:22Z","updated_at":"2021-05-
10T09:46:03Z","last_entry_id":145},"feeds":[{"created_at":"2021-06-13T15:45:23Z","entry_id":46,"field1":"464"},{"created_at":"2021-06-13T15:45:41Z","entry_id":47,"field1":"463"},{"created_at":"2021-06-
13T15:45:56Z","entry_id":48,"field1":"470"},{"created_at":"2021-06-13T15:46:12Z","entry_id":49,"field1":"467"},{"created_at":"2021-06-13T15:46:29Z","entry_id":50,"field1":"443"},{"created_at":"2021-06-
13T15:46:45Z","entry_id":51,"field1":"470"},{"created_at":"2021-06-13T15:47:03Z","entry_id":52,"field1":"494"},{"created_at":"2021-06-13T15:47:19Z","entry_id":53,"field1":"496"},{"created_at":"2021-06-
13T15:47:56Z","entry_id":54,"field1":"470"},{"created_at":"2021-06-13T15:48:12Z","entry_id":55,"field1":"460"},{"created_at":"2021-06-13T15:48:29Z","entry_id":56,"field1":"461"},{"created_at":"2021-06-
13T15:49:08Z","entry_id":57,"field1":"471"},{"created_at":"2021-06-13T15:49:24Z","entry_id":58,"field1":"470"},{"created_at":"2021-06-13T15:49:41Z","entry_id":59,"field1":"469"},{"created_at":"2021-06-
13T15:50:02Z","entry_id":60,"field1":"471"},{"created_at":"2021-06-13T15:50:17Z","entry_id":61,"field1":"465"},{"created_at":"2021-06-13T15:50:32Z","entry_id":62,"field1":"466"},{"created_at":"2021-06-
13T15:50:51Z","entry_id":63,"field1":"464"},{"created_at":"2021-06-13T15:51:06Z","entry_id":64,"field1":"470"},{"created_at":"2021-06-13T15:51:22Z","entry_id":65,"field1":"470"},{"created_at":"2021-06-
13T15:51:38Z","entry_id":66,"field1":"470"},{"created_at":"2021-06-13T15:51:54Z","entry_id":67,"field1":"462"},{"created_at":"2021-06-13T15:52:10Z","entry_id":68,"field1":"462"},{"created_at":"2021-06-
13T15:52:26Z","entry_id":69,"field1":"463"},{"created_at":"2021-06-13T15:52:42Z","entry_id":70,"field1":"464"},{"created_at":"2021-06-13T15:52:58Z","entry_id":71,"field1":"469"},{"created_at":"2021-06-
13T15:53:18Z","entry_id":72,"field1":"465"},{"created_at":"2021-06-13T15:53:34Z","entry_id":73,"field1":"468"},{"created_at":"2021-06-13T15:53:50Z","entry_id":74,"field1":"465"},{"created_at":"2021-06-
13T15:54:08Z","entry_id":75,"field1":"488"},{"created_at":"2021-06-13T15:55:19Z","entry_id":76,"field1":"471"},{"created_at":"2021-06-13T15:55:36Z","entry_id":77,"field1":"461"},{"created_at":"2021-06-
13T15:55:54Z","entry_id":78,"field1":"466"},{"created_at":"2021-06-13T15:56:09Z","entry_id":79,"field1":"469"},{"created_at":"2021-06-13T15:56:26Z","entry_id":80,"field1":"488"},{"created_at":"2021-06-
13T15:56:48Z","entry_id":81,"field1":"464"},{"created_at":"2021-06-13T15:57:03Z","entry_id":82,"field1":"490"},{"created_at":"2021-06-13T15:57:19Z","entry_id":83,"field1":"496"},{"created_at":"2021-06-
13T15:57:37Z","entry_id":84,"field1":"491"},{"created_at":"2021-06-13T15:57:54Z","entry_id":85,"field1":"495"},{"created_at":"2021-06-13T15:58:10Z","entry_id":86,"field1":"489"},{"created_at":"2021-06-
13T15:58:26Z","entry_id":87,"field1":"493"},{"created_at":"2021-06-13T15:58:41Z","entry_id":88,"field1":"488"},{"created_at":"2021-06-13T15:58:57Z","entry_id":89,"field1":"494"},{"created_at":"2021-06-
13T15:59:13Z","entry_id":90,"field1":"486"},{"created_at":"2021-06-13T15:59:28Z","entry_id":91,"field1":"495"},{"created_at":"2021-06-13T15:59:43Z","entry_id":92,"field1":"495"},{"created_at":"2021-06-
13T16:00:00Z","entry_id":93,"field1":"490"},{"created_at":"2021-06-13T16:00:15Z","entry_id":94,"field1":"487"},{"created_at":"2021-06-13T16:00:30Z","entry_id":95,"field1":"495"},{"created_at":"2021-06-
13T16:00:48Z","entry_id":96,"field1":"487"},{"created_at":"2021-06-13T16:01:04Z","entry_id":97,"field1":"491"},{"created_at":"2021-06-13T16:16:23Z","entry_id":98,"field1":"471"},{"created_at":"2021-06-
13T16:16:38Z","entry_id":99,"field1":"461"},{"created_at":"2021-06-13T16:16:56Z","entry_id":100,"field1":"466"},{"created_at":"2021-06-13T16:17:12Z","entry_id":101,"field1":"469"},{"created_at":"2021-06-
13T16:17:27Z","entry_id":102,"field1":"466"},{"created_at":"2021-06-13T16:17:44Z","entry_id":103,"field1":"467"},{"created_at":"2021-06-13T16:17:59Z","entry_id":104,"field1":"465"},{"created_at":"2021-06-
13T16:18:19Z","entry_id":105,"field1":"471"},{"created_at":"2021-06-13T16:18:35Z","entry_id":106,"field1":"471"},{"created_at":"2021-06-13T16:18:51Z","entry_id":107,"field1":"470"},{"created_at":"2021-06-
13T16:19:10Z","entry_id":108,"field1":"466"},{"created_at":"2021-06-13T16:19:25Z","entry_id":109,"field1":"469"},{"created_at":"2021-06-13T16:19:41Z","entry_id":110,"field1":"465"},{"created_at":"2021-06-
13T16:19:59Z","entry_id":111,"field1":"469"},{"created_at":"2021-06-13T16:20:15Z","entry_id":112,"field1":"464"},{"created_at":"2021-06-13T16:20:33Z","entry_id":113,"field1":"469"},{"created_at":"2021-06-
13T16:20:49Z","entry_id":114,"field1":"467"},{"created_at":"2021-06-13T16:21:04Z","entry_id":115,"field1":"466"},{"created_at":"2021-06-13T16:21:19Z","entry_id":116,"field1":"463"},{"created_at":"2021-06-
13T16:21:35Z","entry_id":117,"field1":"471"},{"created_at":"2021-06-13T16:21:50Z","entry_id":118,"field1":"470"},{"created_at":"2021-06-13T16:22:06Z","entry_id":119,"field1":"462"},{"created_at":"2021-06-
13T16:22:24Z","entry_id":120,"field1":"469"},{"created_at":"2021-06-13T16:22:40Z","entry_id":121,"field1":"463"},{"created_at":"2021-06-13T16:22:58Z","entry_id":122,"field1":"461"},{"created_at":"2021-06-
13T16:23:15Z","entry_id":123,"field1":"467"},{"created_at":"2021-06-13T16:23:33Z","entry_id":124,"field1":"465"},{"created_at":"2021-06-13T16:23:48Z","entry_id":125,"field1":"469"},{"created_at":"2021-06-
13T16:24:04Z","entry_id":126,"field1":"468"},{"created_at":"2021-06-13T16:24:19Z","entry_id":127,"field1":"462"},{"created_at":"2021-06-13T16:24:35Z","entry_id":128,"field1":"469"},{"created_at":"2021-06-
13T16:24:53Z","entry_id":129,"field1":"462"},{"created_at":"2021-06-13T16:25:09Z","entry_id":130,"field1":"468"},{"created_at":"2021-06-13T16:25:26Z","entry_id":131,"field1":"468"},{"created_at":"2021-06-
13T16:25:44Z","entry_id":132,"field1":"469"},{"created_at":"2021-06-13T16:26:01Z","entry_id":133,"field1":"461"},{"created_at":"2021-06-13T16:26:16Z","entry_id":134,"field1":"469"},{"created_at":"2021-06-
13T16:26:32Z","entry_id":135,"field1":"464"},{"created_at":"2021-06-13T16:26:50Z","entry_id":136,"field1":"471"},{"created_at":"2021-06-13T16:27:06Z","entry_id":137,"field1":"467"},{"created_at":"2021-06-
13T16:27:24Z","entry_id":138,"field1":"465"},{"created_at":"2021-06-13T16:27:40Z","entry_id":139,"field1":"464"},{"created_at":"2021-06-13T16:27:58Z","entry_id":140,"field1":"471"},{"created_at":"2021-06-
13T16:28:15Z","entry_id":141,"field1":"468"},{"created_at":"2021-06-13T16:28:34Z","entry_id":142,"field1":"462"},{"created_at":"2021-06-13T16:51:49Z","entry_id":143,"field1":"471"},{"created_at":"2021-06-
13T16:52:11Z","entry_id":144,"field1":"469"},{"created_at":"2021-06-13T16:52:41Z","entry_id":145,"field1":"469"}]}
```


Figure 13: A sample of JSON code containing temperature data

Since the sensors continue to record new measurements, they are recorded into the JSON code and to refresh the data displayed in the app, simple down swipe refresh is all that's needed.

Here are snapshots of the sensor and home activities working in an android device:

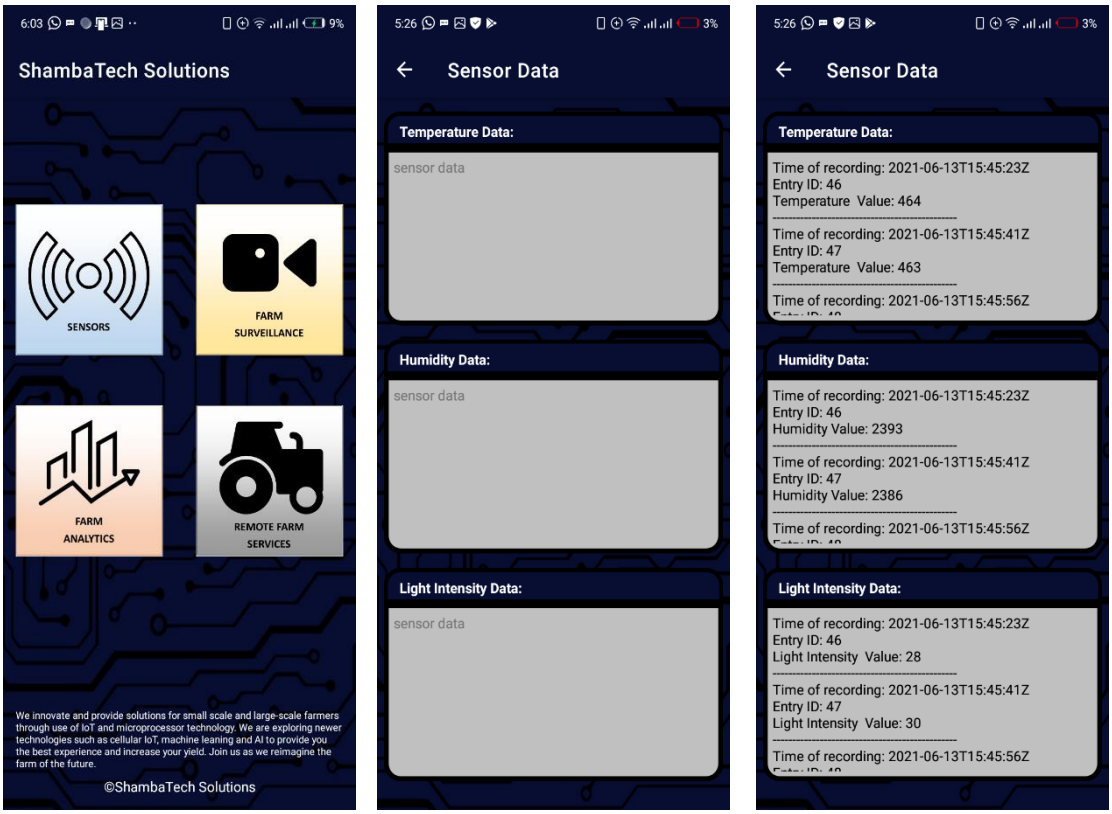


Figure 14: The home activity, the sensors activity without data and the sensors activity with data, respectively.

For connectivity, the android manifest file was modified, adding permission for internet and Wi-Fi accessibility:

The farm surveillance, farm analytics and remote farm operations activities do not have functional features but their links were included in the home activity to show how the app would look like if it was fully functional.

For the surveillance activity, 4 video containers were inserted but they are not functional since there aren't any cameras included in the proof of concept. The farm analytics activity was created but it is empty since the data processing models and algorithms were not implemented in the IoT server. The remote farm operations activity wasn't created so the its link in the home page will not redirect to an activity. These activities weren't developed fully because of the situation in which we were doing our projects since it was during the pandemic which made it difficult to implement all the objectives if the project.

The mobile app however does implement the main objective, creating a user-friendly interface between the farmer and the sensors located in the farm.

CONCLUSIONS AND RECOMMENDATIONS

The project was a success, data was transferred from a sensor node, to a gateway network, to the thingspeak network and finally posted to an IoT app. As the main vision of our project was to collect data from the farm and be able to visualize it, that route was achieved. The other goal was to have automation, that is from the app one would be able to send instructions for automation in the farm. For example, if data collected from the farm says it's too hot, a farmer would like to turn on water sprinklers in the farm thus sending such instructions via the app. We believe this is the next step for the project as communication is half duplex between the lora chips thus a reverse of data from the gateway to the node can be fully established.

Further steps from the project would to now instead of using the esp8266 board for the implementation of the node circuit and the gateway circuit, we would design boards that are only suited for their functionality that is a gateway to only have gateway functions like being able to accommodate multiple nodes and a node circuit that would only be applicable for sending data.

In large scale applications, a common gateway can be developed to handle heavy amounts of data collected from several farms that are sharing the platform. This can be a provided service where we install all the system in their farms and the user interfaces, the app and the desktop program. Data processing methods can be developed to process the collected data and provide them with predictions and recommendations. This would create a collaborative environment between farmers and the system developers which would positively impact the overall yield and quality of produce that comes from a common agricultural community.

In the future, we plan to continue to develop this project from a proof of concept to a fully working system. There have been great strides in 5G and we want to integrate it into this project. The sensors in the farm can be wireless and connected straight to a cloud. Cellular IoT has proven to be a useful tool since the existing infrastructure used in cellular communication for data transfer.

REFERENCES

1. Kazarinoff, P. (2021). Building an IoT Server with Flask and Python.

   https://pythonforundergradengineers.com/search.html?q=IoT

2. Makerfabs-He. (2020, June 16). Lora Gateway Based on MicroPython ESP32. Hackster.Io.

   https://www.hackster.io/Makerfabs01/lora-gateway-based-on-micropython-esp32-

   0aa32b#toc-step-6--lorawan-gateway-5

3. Scharler, H. (2017, December 20). Learn How to Build a Custom Android App for a ThingSpeak IoT Project. Hans on IoT. https://blogs.mathworks.com/iot/2017/12/20/learn-how-to-build-a-custom-android-app-for-a-thingspeak-iot-project/