PROJECT DEVELOPMENT PHASE

PROJECT DEVELOPMENT – DELIVERY OF SPRINT 2

Date	4 November 2022
Team ID	PNT2022TMID15148
Project Name	Project – SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

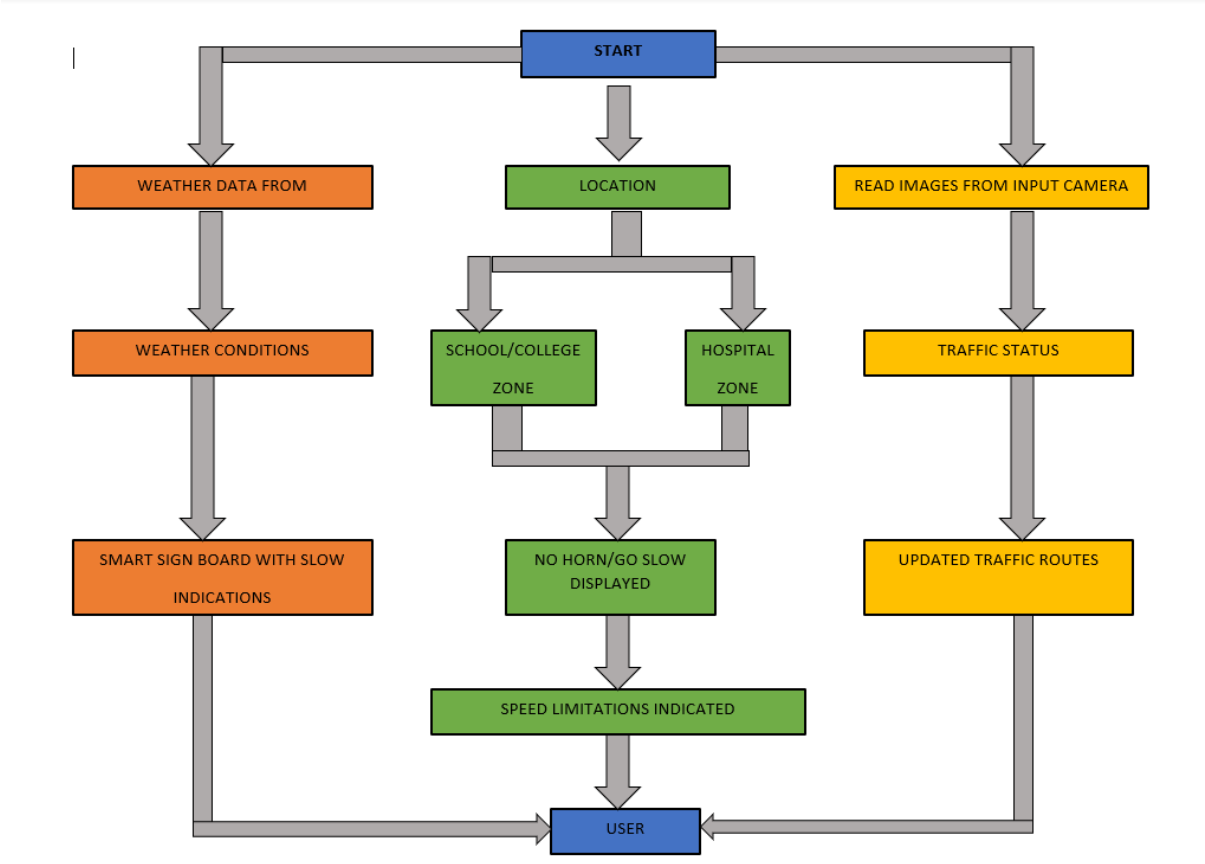
Project Development – Delivery of Sprint 2:

SIGNS WITH SMART CONNECTIVITY FOR BETTER ROAD SAFETY

Sprint Goals:

- Push data from local code to cloud

Data Flow:



Program Code:

- ✓ **(. /openweatherapi.py)**

This file contains a utility function that uses OpenWeatherAPI to retrieve the weather. It only gives certain results necessary inputs for the API results.

```
import requests as reqs

def get(myLocation,APIKEY):
    apiURL = f"https://api.openweathermap.org/data/2.5/weather?q={myLocati
on}&appid={APIKEY}"
    responseJSON = (reqs.get(apiURL)).json()
    returnObject = {
        "temperature" : responseJSON['main']['temp'] - 273.15,
        "weather" : [responseJSON['weather'][_]['main'].lower() for _ in r
ange(len(responseJSON['weather']))],
        "visibility" : responseJSON['visibility']/100,
    }
    if("rain" in responseJSON):
        returnObject["rain"] = [responseJSON["rain"][key] for key in respo
nseJSON["rain"]]
    return(returnObject)
```

- ✓ **(. /datatopublish.py)**

This programme logs data as well as pushing info to the cloud. The following webpage is set up to display the data from the IBM Cloud: **CLICK TO OPEN NODE RED DASHBOARD**

```
# Python code

# IMPORT SECTION STARTS

import wiotp.sdk.device # python -m pip install wiotp
import time

# IMPORT SECTION ENDS
# ---------------------------------------------
# API CONFIG SECTION STARTS

myConfig = {
    "identity" : {
        "orgId" : "epmoec",
        "typeId" : "testDevice",
        "deviceId" : "device0"
    },
    "auth" : {
        "token" : "?-KDXUPMvDo_TK2&b1"
```

```
    }
}

# API CONFIG SECTION ENDS
# ---------------------------------------------
# FUNCTIONS SECTION STARTS

def myCommandCallback(cmd):
    print("recieved cmd : ",cmd)


def logData2Cloud(location,temperature,visibility):
    client = wiotp.sdk.device.DeviceClient(config=myConfig,logHandlers=Non
e)
    client.connect()
    client.publishEvent(eventId="status",msgFormat="json",data={
        "temperature" : temperature,
        "visibility" : visibility,
        "location" : location
    },qos=0,onPublish=None)
    client.commandCallback = myCommandCallback
    client.disconnect()
    time.sleep(1)

# FUNCTIONS SECTION ENDS
```

- ✓ **(. /harddisp.py)**

This file is a utility function that abstracts all extraneous details and only returns the information that is necessary to be presented on the hardware side. The logic behind the code flow is put into action here.

```
from datetime import datetime as dt
from datatopublish import logData2Cloud as log2cloud

# IMPORT SECTION ENDS
# ---------------------------------------------
# UTILITY LOGIC SECTION STARTS
def processConditions(myLocation,APIKEY,localityInfo):
    weatherData = weather.get(myLocation,APIKEY)

    log2cloud(myLocation,weatherData["temperature"],weatherData["visibilit
y"])

    finalSpeed = localityInfo["usualSpeedLimit"] if "rain" not in weatherD
ata else localityInfo["usualSpeedLimit"]/2
```

```
    finalSpeed = finalSpeed if weatherData["visibility"]>35 else finalSpee
d/2

    if(localityInfo["hospitalsNearby"]):
        # hospital zone
        doNotHonk = True
    else:
        if(localityInfo["schools"]["schoolZone"]==False):
            # neither school nor hospital zone
            doNotHonk = False
        else:
            # school zone
            now = [dt.now().hour,dt.now().minute]
            activeTime = [list(map(int,_.split(":"))) for _ in localityInf
o["schools"]["activeTime"]]
            doNotHonk = activeTime[0][0]<=now[0]<=activeTime[1][0] and act
iveTime[0][1]<=now[1]<=activeTime[1][1]

    return({
        "speed" : finalSpeed,
        "doNotHonk" : doNotHonk
    })

# UTILITY LOGIC SECTION ENDS
```

- ✓ **(. /mcon.py)**

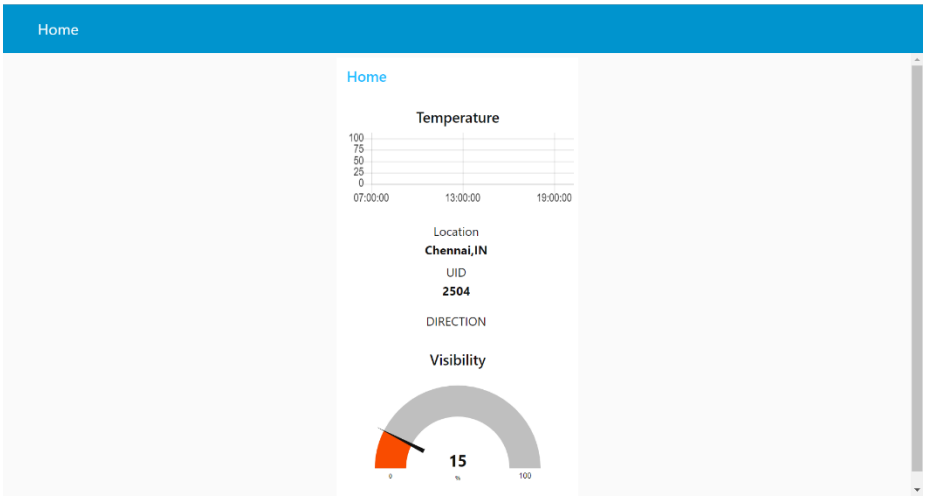
The programme that the microcontroller executes indefinitely. The output hardware display is changed dependent on the return result from each of the util functions called from other Python scripts.

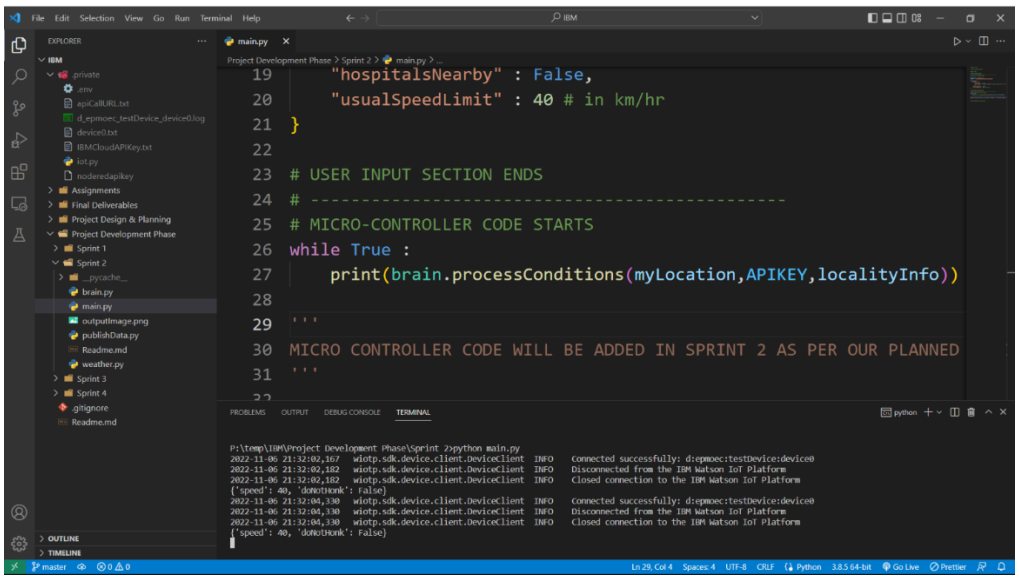
```
# Python code

# IMPORT SECTION STARTS

import harddisp

# IMPORT SECTION ENDS
# -----------------------------------------------
# USER INPUT SECTION STARTS

myLocation = "Chennai,IN"
APIKEY = "bf4a8d480ee05c00952bf65b78ae826b"

localityInfo = {
    "schools" : {
```

```
        "schoolZone" : True,
        "activeTime" : ["7:00","17:30"] # schools active from 7 AM till 5:
30 PM
        },
    "hospitalsNearby" : False,
    "usualSpeedLimit" : 40 # in km/hr
}

# USER INPUT SECTION ENDS
# ----------------------------------------------
# MICRO-CONTROLLER CODE STARTS
while True :
    print(brain.processConditions(myLocation,APIKEY,localityInfo))

# MICRO-CONTROLLER CODE ENDS
```

Output:

```
# Code Output
2022-11-06 21:38:33,452   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:epmoec:testDevice:device0
2022-11-06 21:38:33,452   wiotp.sdk.device.client.DeviceClient  INFO    Disconnected from the IBM Watson IoT Platform
2022-11-06 21:38:33,452   wiotp.sdk.device.client.DeviceClient  INFO    Closed connection to the IBM Watson IoT Platform
{'speed': 40, 'doNotHonk': False}
2022-11-06 21:38:35,631   wiotp.sdk.device.client.DeviceClient  INFO    Connected successfully: d:epmoec:testDevice:device0
2022-11-06 21:38:35,631   wiotp.sdk.device.client.DeviceClient  INFO    Disconnected from the IBM Watson IoT Platform
2022-11-06 21:38:35,631   wiotp.sdk.device.client.DeviceClient  INFO    Closed connection to the IBM Watson IoT Platform
{'speed': 40, 'doNotHonk': False}
.
.
.
... repeats every 1 sec
```




File Edit Selection View Go Run Terminal Help
IBM
EXPLORER
IBM
.private
.env
apiCallURL.txt
d_epmoec_testDevice_device0.log
device0.txt
IBMCloudAPIKey.txt
iot.py
noderedapikey
Assignments
Final Deliverables
Project Design & Planning
Project Development Phase
Sprint 1
Sprint 2
__pycache__
brain.py
main.py
outputImage.png
publishData.py
Readme.md
weather.py
Sprint 3
Sprint 4
.gitignore
Readme.md
OUTLINE
TIMELINE
main.py
Project Development Phase > Sprint 2 > main.py > ...
"hospitalsNearby" : False,
"usualSpeedLimit" : 40 # in km/hr
}
# USER INPUT SECTION ENDS
# --
# MICRO-CONTROLLER CODE STARTS
while True :
print(brain.processConditions(myLocation,APIKEY,localityInfo))
'''
MICRO CONTROLLER CODE WILL BE ADDED IN SPRINT 2 AS PER OUR PLANNED
'''
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
python
P:\temp\IBM\Project Development Phase\Sprint 2>python main.py
2022-11-06 21:32:02,167 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:epmoec:testDevice:device0
2022-11-06 21:32:02,182 wiotp.sdk.device.client.DeviceClient INFO Disconnected from the IBM Watson IoT Platform
2022-11-06 21:32:02,182 wiotp.sdk.device.client.DeviceClient INFO Closed connection to the IBM Watson IoT Platform
{'speed': 40, 'doNotHonk': False}
2022-11-06 21:32:04,330 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:epmoec:testDevice:device0
2022-11-06 21:32:04,330 wiotp.sdk.device.client.DeviceClient INFO Disconnected from the IBM Watson IoT Platform
2022-11-06 21:32:04,330 wiotp.sdk.device.client.DeviceClient INFO Closed connection to the IBM Watson IoT Platform
{'speed': 40, 'doNotHonk': False}
master
0 0
Ln 29, Col 4 Spaces: 4 UTF-8 CRLF Python 3.8.5 64-bit Go Live Prettier