

Project Development Phase
Project Development Delivery of Sprint 1

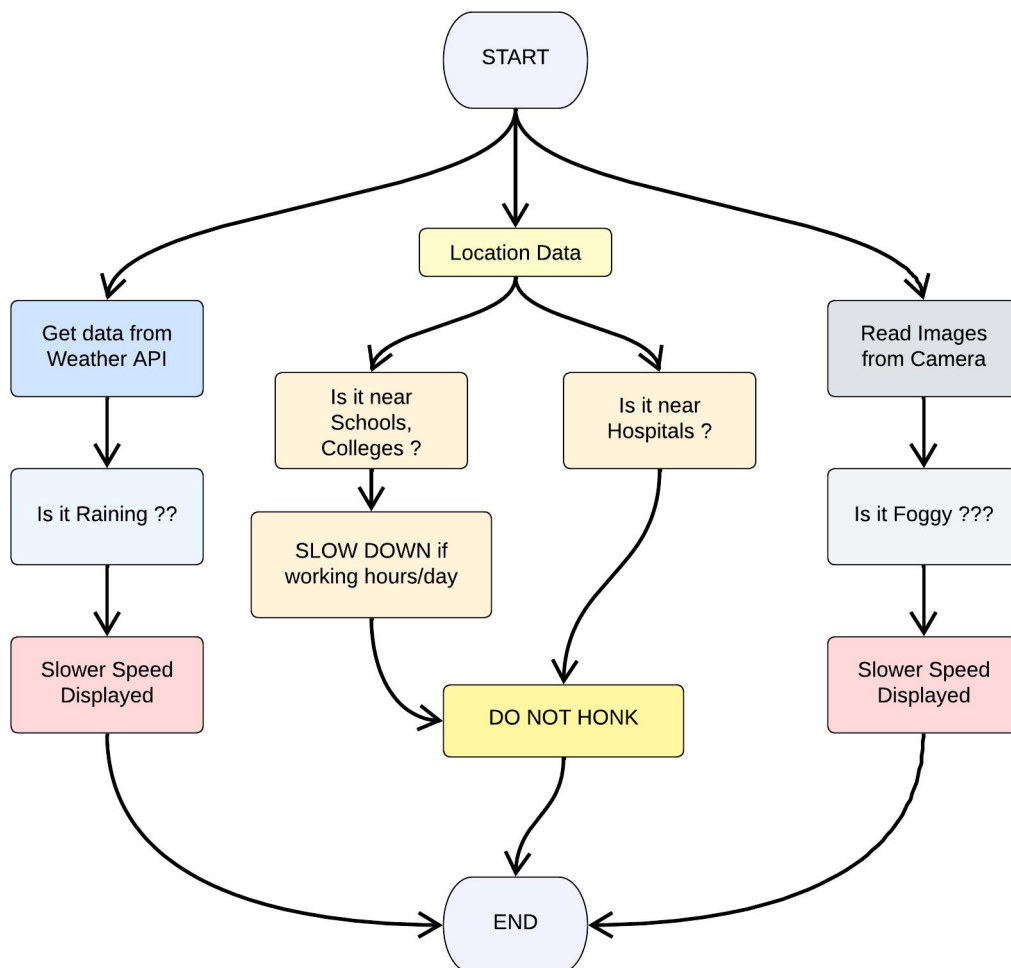
Date	07 September 2022
Team ID	PNT2022TMID35120
Project Name	Signs with smart connectivity for Better road safety
Maximum Marks	4 Marks

Signs with smart connectivity for Better road safety

Sprint Goals :

1. Create and initialize accounts in various public APIs like OpenWeather API
2. Write a Python program that outputs results given the inputs like weather and location.

Code Flow :



Program Code :

weather.py

```

# Python code
import requests as reqs
def get(myLocation, APIKEY):
    apiURL = f"https://api.openweathermap.org/data/2.5/weather?q={myLocation}&appid={APIKEY}"
    responseJSON = (reqs.get(apiURL)).json()
    responseObject = {
        "temperature": responseJSON['main']['temp'] - 273.15,
        "weather": [responseJSON['weather'][_]['main'].lower()
                     for _ in range(len(responseJSON['weather']))],
        "visibility": responseJSON['visibility']/100, # visibility in percentage where 10km is
        100% and 0km is 0%
    }
    if("rain" in responseJSON):
        responseObject["rain"] = [responseJSON["rain"][key] for key in responseJSON["rain"]]
    return(responseObject)

```

brain.py

This file is a utility function that returns only essential information to be displayed at the hardware side and abstracts all the unnecessary details. This is where the code flow logic is implemented.

```

# Python code
# IMPORT SECTION STARTS
import weather
from datetime import datetime as dt
# IMPORT SECTION ENDS
# ---
# UTILITY LOGIC SECTION STARTS
def processConditions(myLocation, APIKEY, localityInfo):
    weatherData = weather.get(myLocation, APIKEY)
    finalSpeed = localityInfo["usualSpeedLimit"]
    if "rain" not in weatherData:
        finalSpeed = localityInfo["usualSpeedLimit"] / 2
    finalSpeed = finalSpeed if weatherData["visibility"] > 35 else finalSpeed / 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 localityInfo["schools"]["activeTime"]]
doNotHonk=activeTime[0][0]<=now[0]<=activeTime[1][0]
andactiveTime[0][1]<=now[1]<=activeTime[1][1]
return({
"speed" : finalSpeed,
"doNotHonk" : doNotHonk
})
# UTILITY LOGIC SECTION ENDS

```

main.py

The code that runs in a forever loop in the microcontroller. This calls all the util functions from other python files and based on the return value transduces changes in the output hardware display.

```

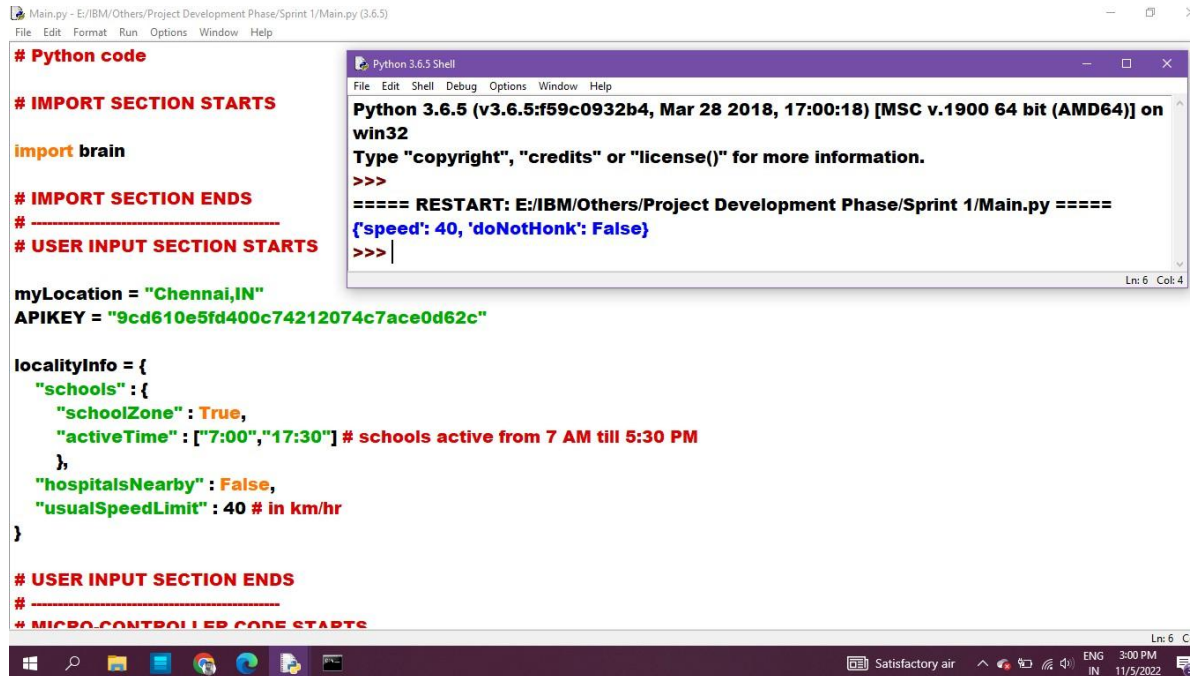
# Python code
# IMPORT SECTION STARTS
import brain
# IMPORT SECTION ENDS
# --- - - - - - - - - - - - - - - - - - - - - ---
# USER INPUT SECTION STARTS
myLocation="Chennai,IN"
APIKEY="9cd610e5fd400c74212074c7ace0d62c"
localityInfo= {
"schools" : {
"schoolZone" : True,
"activeTime" : ["7:00","17:30"] # schools active from 7 AM till 5:30 PM
},
"hospitalsNearby" : False,
"usualSpeedLimit" : 40# in km/hr
}
# USER INPUT SECTION ENDS
# --- - - - - - - - - - - - - - - - - - - - - ---
# MICRO-CONTROLLER CODE STARTS
print(brain.processConditions(myLocation,APIKEY,localityInfo))
'''
MICRO CONTROLLER CODE WILL BE ADDED IN SPRINT 2 AS PER OUR PLANNED
SPRINT SCHEDULE
'''
# MICRO-CONTROLLER CODE ENDS

```

Output :

Code Output

```
{'speed': 40, 'doNotHonk': False}
```



The screenshot shows a Python IDE with a file named 'Main.py' open. The code in the editor includes comments for different sections: '# Python code', '# IMPORT SECTION STARTS', '# IMPORT SECTION ENDS', '# USER INPUT SECTION STARTS', '# USER INPUT SECTION ENDS', and '# MICRO-CONTROLLER CODE STARTS'. The code defines a dictionary 'localityInfo' with keys for 'schools', 'hospitalsNearby', and 'usualSpeedLimit'. The 'schools' key has a nested dictionary with 'schoolZone' (True) and 'activeTime' (['7:00', '17:30']). The 'usualSpeedLimit' is set to 40 km/hr. The code also defines 'myLocation' as 'Chennai,IN' and 'APIKEY' as a long alphanumeric string. A Python 3.6.5 Shell window is open, showing the output of the code execution: 'Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32', followed by a prompt 'Type "copyright", "credits" or "license()" for more information.', and then the output of the code: '==== RESTART: E:/IBM/Others/Project Development Phase/Sprint 1/Main.py ====', followed by the dictionary {'speed': 40, 'doNotHonk': False}.

```
# Python code

# IMPORT SECTION STARTS
import brain
# IMPORT SECTION ENDS
# =====
# USER INPUT SECTION STARTS

myLocation = "Chennai,IN"
APIKEY = "9cd610e5fd400c74212074c7ace0d62c"

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
```

```
Python 3.6.5 (v3.6.5:f59c0932b4, Mar 28 2018, 17:00:18) [MSC v.1900 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: E:/IBM/Others/Project Development Phase/Sprint 1/Main.py ====
{'speed': 40, 'doNotHonk': False}
>>>
```