

PROJECT DEVELOPMENT PHASE

DELIVERY OF SPRINT 4

Date	14th November 2022
Team ID	PNT2022TMID35860
Project Name	Project – Personal Assistance for senior citizens who are self-reliant
Team members	K.Gurubaran B.Mejalin Arno J.Vinothagan R.Arunkumar

SPRINT IV: Generating voice commands and Alert system

Outline of Sprint 4

This sprint delivery document contains the following,

- 1)Developing Python code to implement Text to speech service
- 2)Alert system via email by Ubidots notification
- 3)Updation of nodes in the node-red platform

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Registration. Creation of IBM services like NodeRED, Cloudant DB, TTS Service and design of IoT system	USN-1	As a user, I must be able to login to the IBM platform	2	High	Gurubaran, Arunkumar
Sprint-2	Web UI. Creation of Web UI using NodeRED service	USN-2	As a user, I must be able to update the medicine details in the web UI	2	High	Vinothagan, Mejalin Arno
Sprint-3	Software implementation. Developing Python code to retrieve data from cloudant db to send that data to IoT device	USN-3	As a user, I must be push the details to the IoT device	2	High	Gurubaran, Mejalin Arno
✓ Sprint-4	Final demonstration and user testing. Generating voice commands using IBM Text to Speech service	USN-4	As a user, I must be able hear the medicine name which is to be taken at the appropriate time and check its accuracy	2	High	Vinothagan, Arunkumar

1)Developing Python code to implement Text to speech service

a) Text to Speech piece of code

```
text2speech.py - C:\Users\HP\LAPTOP-U8QGQJFE\Downloads\text2speech.py (3.7.0)
File Edit Format Run Options Window Help
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound

authenticator = IAMAuthenticator('jHG72RxBEzpJDs4vkDt6DySXoaJu9hylvn0hjE_p-F0g')
text_to_speech = TextToSpeechV1(
    authenticator=authenticator
)

text_to_speech.set_service_url('https://api.au-syd.text-to-speech.watson.cloud.ibm.com/instances/74dcleed-1e64-4f57-ba4a-2031a8f39d85')
with open('med.mp3', 'wb') as audio_file:
    audio_file.write(text_to_speech.synthesize('Take Crocin 50 mg Now', voice='en-US_AllisonV3Voice', accept='audio/wav').get_result().content)

print("playing")
playsound('med.mp3')
```

Using ibm_watson library, we are going to implement text to speech service

The final code:

```
script.py - C:\Users\HP\LAPTOP-U8QGQJFE\Downloads\script.py (3.7.0)
File Edit Format Run Options Window Help

import json
from ibm_watson import TextToSpeechV1
from ibm_cloud_sdk_core.authenticators import IAMAuthenticator
from playsound import playsound
prevMedicine = ''
currMedicine = ''
while True:
    req=requests.get("http://169.51.206.114:32641/remainder")
    value=req.json()
    try:
        prevMedicine = value['command']
        ORG= "jchm38"
        DEVICE_TYPE ="MR"
        DEVICE_ID ="2019504037"
        TOKEN ="(!xRUci*BCpeso-rk"
        server = ORG + ".messaging.internetofthings.ibmcloud.com";
        pubTopic1 = "iot-2/evt/medicine/fmt/string"
        pubTopic2 = "iot-2/evt/ph/fmt/json"
        pubTopic3 = "iot-2/evt/turb/fmt/json"
        #pubTopic3 = "iot-2/evt/wf/fmt/json"

        authMethod = "use-token-auth";
        token = TOKEN;
        clientId = "d:" + ORG + ":" + DEVICE_TYPE + ":" + DEVICE_ID;
        if currMedicine != prevMedicine:
            mqttc = mqtt.Client(client_id=clientId)
            mqttc.username_pw_set(authMethod, token)
            mqttc.connect(server, 1883, 60)
            mqttc.publish(pubTopic1,json.dumps(value))
            print("Published Successfully!")
            authenticator = IAMAuthenticator('jHG72RxBEzpJDs4vkDt6DySXoaJu9hylvn0hjE_p-F0g')
            text_to_speech = TextToSpeechV1(
                authenticator=authenticator
            )

            text_to_speech.set_service_url('https://api.au-syd.text-to-speech.watson.cloud.ibm.com/instances/74dcleed-1e64-4f57-ba4a-2031a8f39d85')
            with open('try.mp3', 'wb') as audio_file:
                audio_file.write(text_to_speech.synthesize('Please Take'+ ' '+value['command']+' '+'+tablet now', voice='en-US_MichaelExpressive', accept='audio/wav').get_result().content)

            print("playing")
            #playsound('try.mp3')

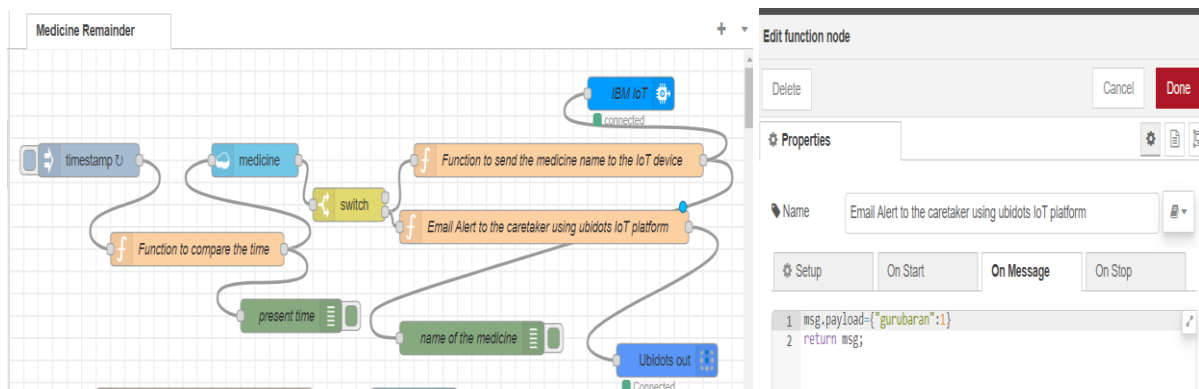
            currMedicine = prevMedicine
        except Exception as error:
            print(error.args[0])
            print("Error!")
mqttc.loop forever()
```

Output

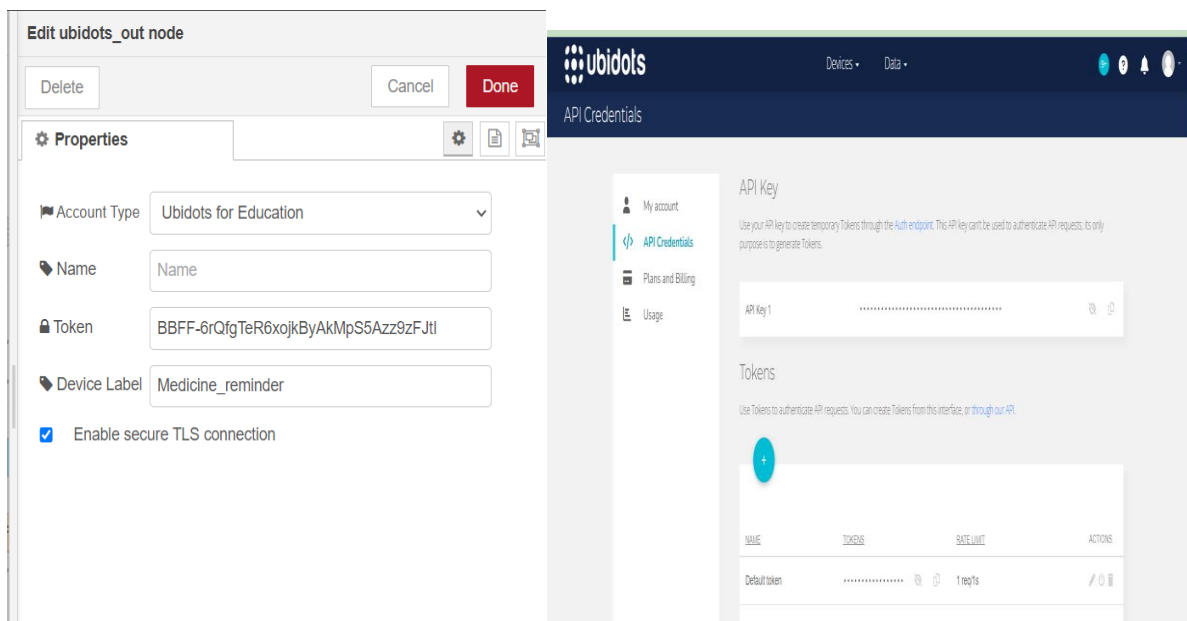
```
*Python 3.7.0 Shell*
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\HP.LAPTOP-U8QGQJFE\Downloads\script.py =====
Published Successfully!
playing
Published Successfully!
playing
```

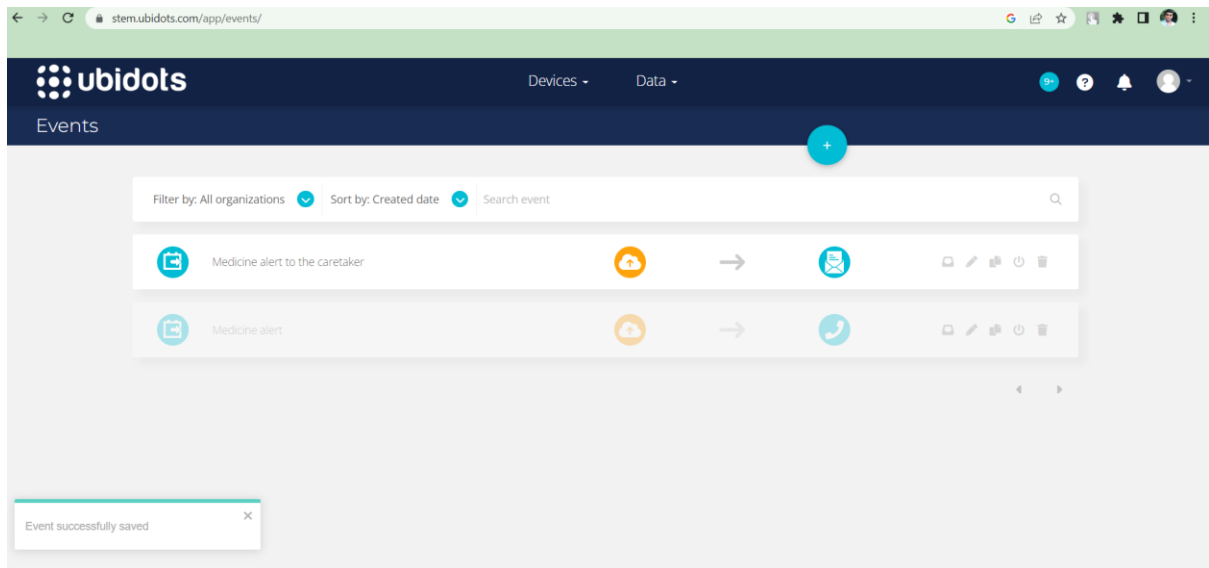
2) Alert system via email by Ubidots notification

By updating the node red flow by adding ubidots out node and creating a function

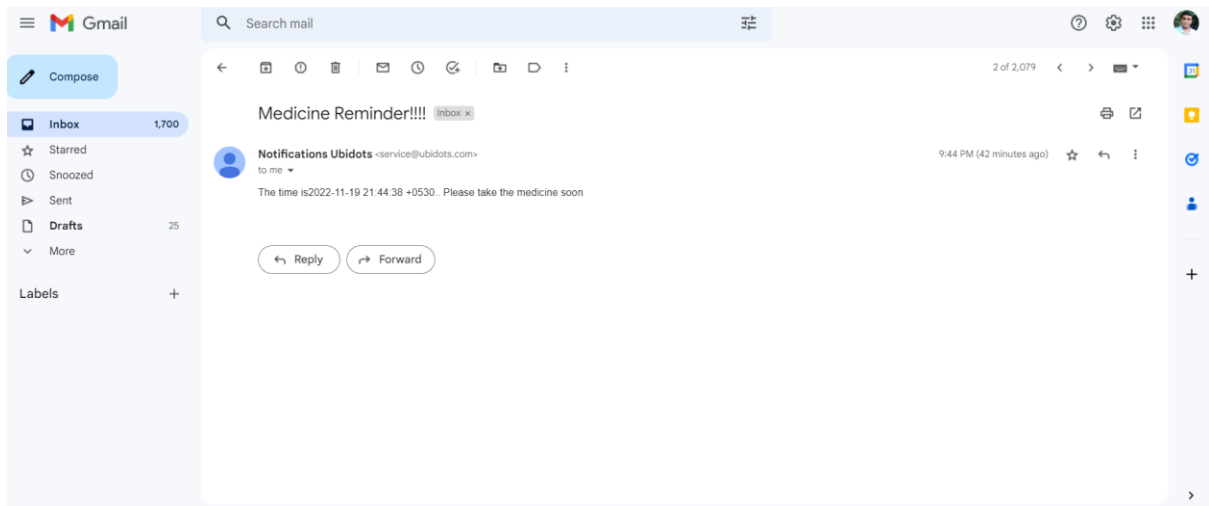


Ubidots node edit and web platform screenshots

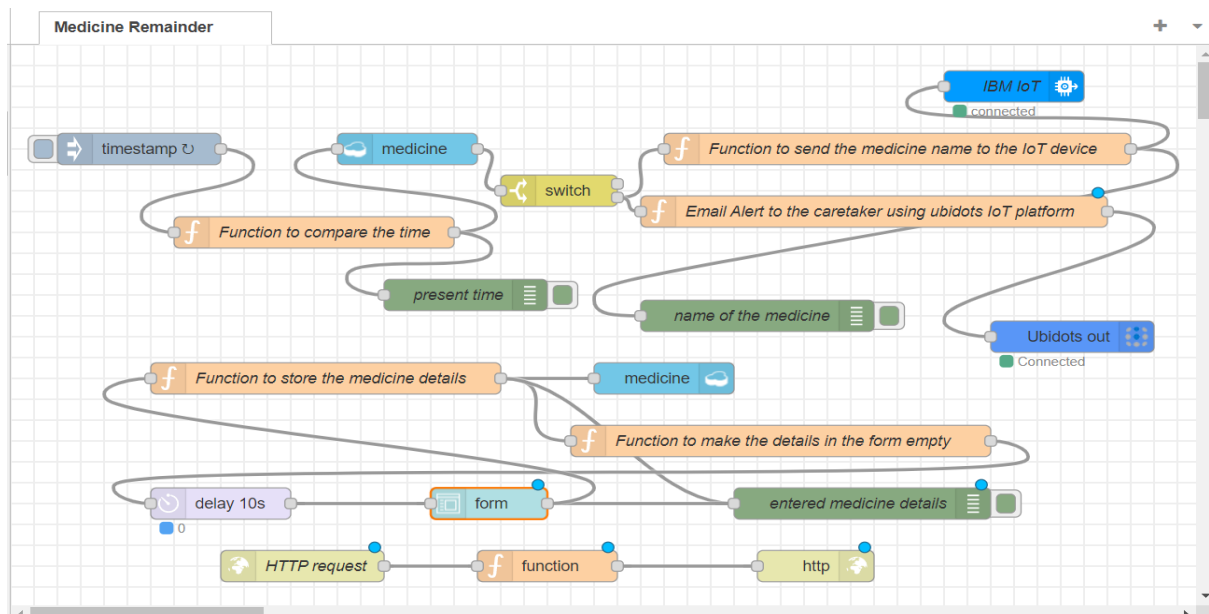




Email sent to the caretaker

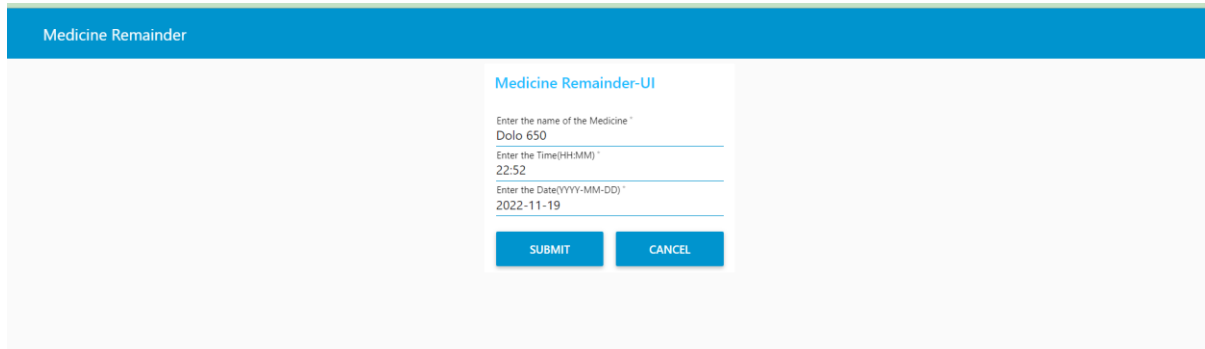


3) Updated nodered flow diagram



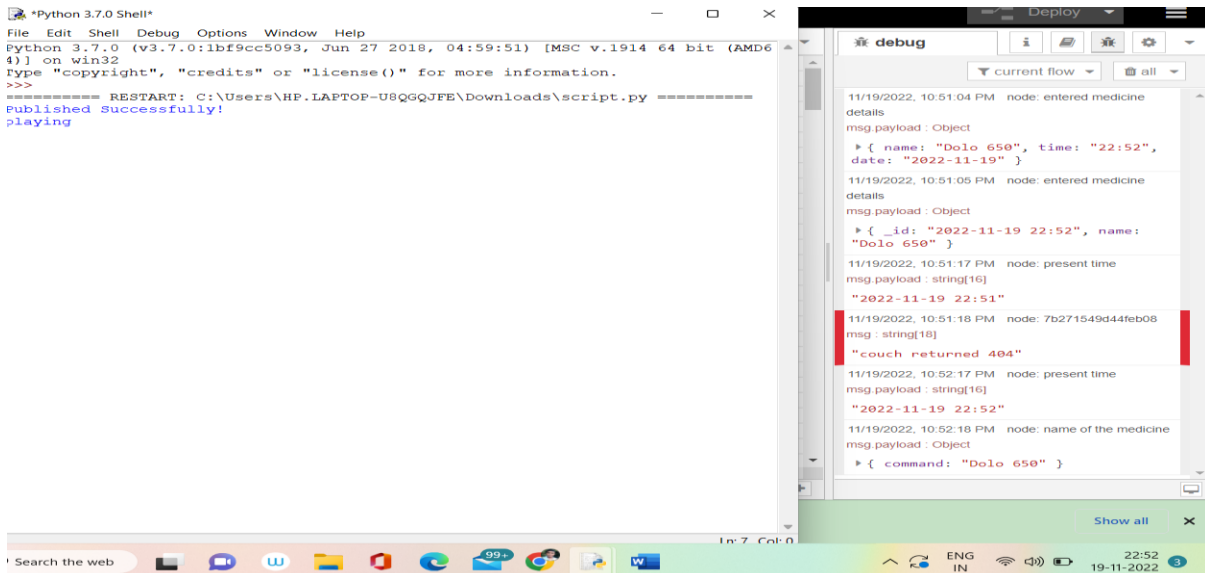
Result:

1) Medicine details are uploaded by the care-taker using the link <http://169.51.206.114:32641/ui>



The screenshot shows a web application titled "Medicine Remainder" with a sub-header "Medicine Remainder-UI". It contains three input fields: "Enter the name of the Medicine" with the value "Dolo 650", "Enter the Time(HHMM)" with the value "22:52", and "Enter the Date(YYYY-MM-DD)" with the value "2022-11-19". Below the fields are two buttons: "SUBMIT" and "CANCEL".

2) When the time arrives, the python code will send the name of the medicine to the IBM IoT Watson platform



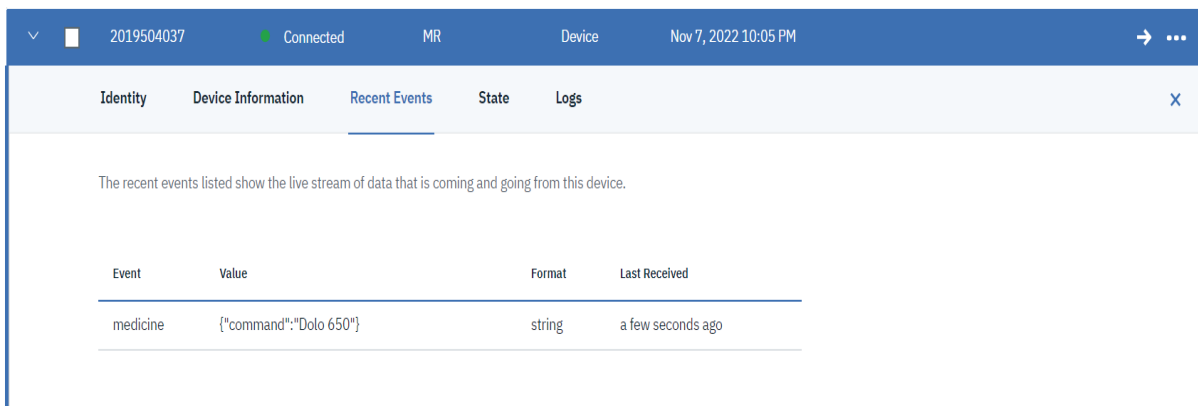
The screenshot shows a Python 3.7.0 Shell window on the left and an IBM IoT Watson debug console on the right. The Python shell shows the execution of a script that sends data to the Watson platform. The debug console shows the received messages, including the medicine name "Dolo 650" and the time "2022-11-19 22:52".

```
Python 3.7.0 Shell
File Edit Shell Debug Options Window Help
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\HP.LAPTOP-U8QGQJFE\Downloads\script.py =====
Published Successfully!
playing
```

debug console messages:

```
11/19/2022, 10:51:04 PM node: entered medicine details
msg.payload: Object
  { name: "Dolo 650", time: "22:52", date: "2022-11-19" }
11/19/2022, 10:51:05 PM node: entered medicine details
msg.payload: Object
  { _id: "2022-11-19 22:52", name: "Dolo 650" }
11/19/2022, 10:51:17 PM node: present time
msg.payload: string[16]
  "2022-11-19 22:51"
11/19/2022, 10:51:18 PM node: 7b271549d44feb08
msg: string[16]
  "couch returned 404"
11/19/2022, 10:52:17 PM node: present time
msg.payload: string[16]
  "2022-11-19 22:52"
11/19/2022, 10:52:18 PM node: name of the medicine
msg.payload: Object
  { command: "Dolo 650" }
```

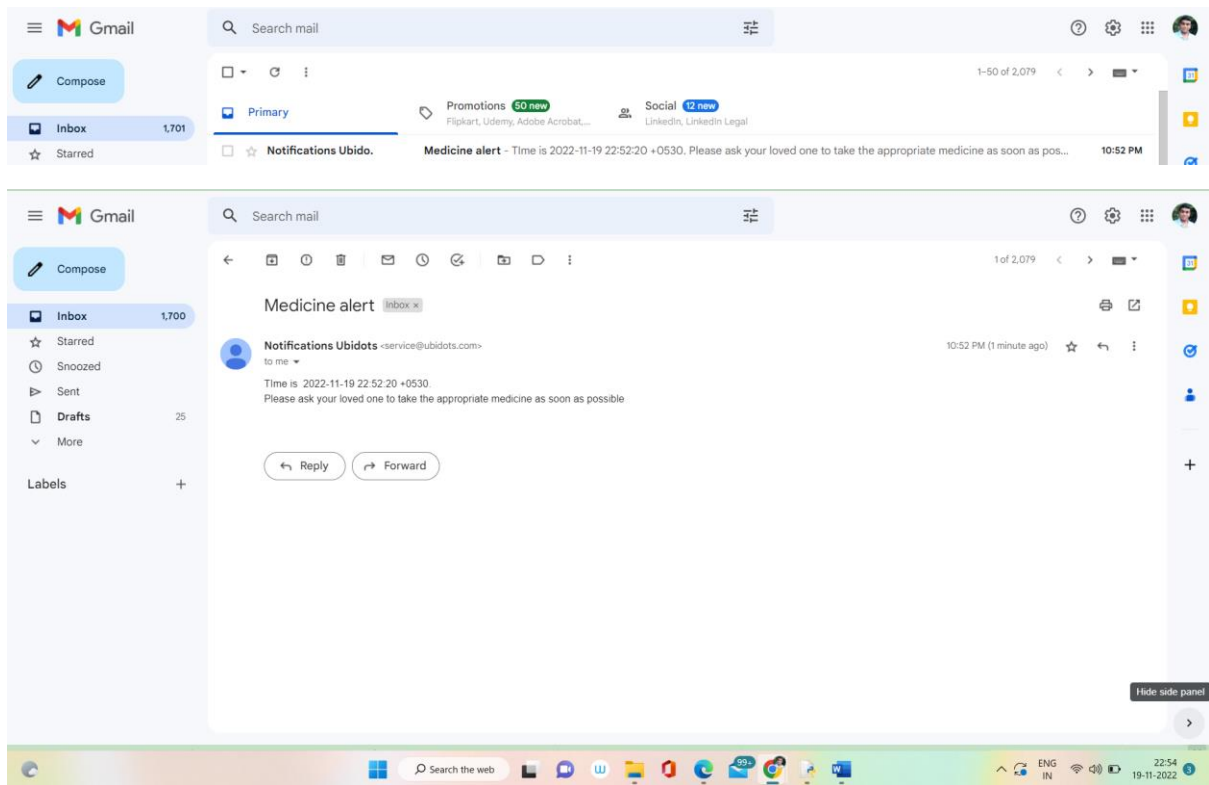
3) The following is the screen shot of the recent events in the IoT Watson platform



The screenshot shows the "Recent Events" page in the IBM IoT Watson platform. The page displays a table of recent events for a device with ID 2019504037. The table has columns for Event, Value, Format, and Last Received. The event "medicine" has a value of {"command": "Dolo 650"} and was received a few seconds ago.

Event	Value	Format	Last Received
medicine	{"command": "Dolo 650"}	string	a few seconds ago

4)An email is sent to the care taker asking the person to intimate the elderly patient



5) Text to Speech command signals to the user : The file is uploaded in the google drive.

https://drive.google.com/file/d/190BT9642_KT801JEqu5nEtIwwkkUql1Y/view?usp=sharing

Summary of Sprint 4:

- 1)Some features are added to the python code to enable TTS mechanism
- 2)An email is sent to the care taker to intimate the time of medicine intake of the patient
- 3)Time of medicine is intimated to the user via voice commands

*****END OF REPORT*****