Development of programming scripts and SQL database to track and report ITS device status

INTRODUCTION

- Development of an Onboard Unit project focuses on creating a platform for the development of Connected Vehicles (CV) applications.
- The Virginia Connected Corridor (VCC) project provides a REST interface and includes a data archive system.
- I developed Python scripts that open a web socket to continuously access and retrieve the data as per our need.
- Permanent Count Station project involves checking the communication with all the stations on daily basis.
- I designed Python script that pings all the stations every hour for 24 hours to check the communication of all the stations and then generates an automatic email with the details of stations and number of times they were up the previous day and sends it to the appropriate user.
- Other part of the Permanent Count Station project involves retrieving the data from the SQL server database based on need and monitoring whether we are receiving non zero data in our database every hour or not.
- I developed SQL query that fetches the data we need from the repository and currently working on designing python scripts for accessing the SQL server database to monitor the data we receive daily.

METHODOLOGY

- Created a concept of operation that highlights the way current system is working, desired changes and the reasons for the changes.
- To retrieve data from the Virginia Connected Corridor, learned how to access the data stored in APIs. Designed and developed python scripts that generate and print output.
- For Permanent Count Station, checked the servers by pinging them using Python.
- Designed a Python script to send an email. Converted output generated to HTML table with background colors in it by developing a python script.
- Operated task scheduler to run python script at a particular time every day to monitor the stations and send the generated report at the end of the day to appropriate user.
- For retrieving the data from SQL server, trimmed the data using INNER JOIN on appropriate tables as well as columns to fetch the data needed.
- Working on designing a script to access SQL server using Python to monitor whether we receive data every hour or not for all the stations.

RESULTS

- Making connection with VCC API and receiving Basic Safety Messages (BSM)

```python
# OUTPUT - Working fine, getting messages

# Importing requests library to interact with API.
import requests


from websocket import create_connection

# In auth, we are providing credentials (key and password)
appToken = 'aq0akwkcgjv2zwpdmB5u3Yq/erd//tNJauCTmfuV/9so9rWjOp482LVJ'
r = requests.get('https://vcc-api.vtti.vt.edu/api/bsm/key',
                 auth=(appToken, 'password_is_ignored'))


# .status_code gives us the connection detail.
# If answer is 200, it means everything went okey and it is ready.
print(r.status_code)

# Saving r.text that is the key to the bsmKey variable.
#(Important : Output from API must be in string format even when it is digits(key) and that
bsmKey = r.text

ws = create_connection("wss://vcc-api.vtti.vt.edu/ws/bsm?rse_id=cell,11,12&key=" + bsmKey)
print("Receiving BSM from WebSocket...")

# Printing BSM messages.
for i in range(0,20):
  result = ws.recv()
  print("%d: Received! '%s'" % (i, result))
ws.close()
```

- When no messages are being sent.

```
200
Receiving BSM from WebSocket...
0: Received! '{ "bsm": [] }'
1: Received! '{ "bsm": [] }'
2: Received! '{ "bsm": [] }'
3: Received! '{ "bsm": [] }'
4: Received! '{ "bsm": [] }'
5: Received! '{ "bsm": [] }'
6: Received! '{ "bsm": [] }'
7: Received! '{ "bsm": [] }'
8: Received! '{ "bsm": [] }'
9: Received! '{ "bsm": [] }'
10: Received! '{ "bsm": [] }'
11: Received! '{ "bsm": [] }'
12: Received! '{ "bsm": [] }'
13: Received! '{ "bsm": [] }'
14: Received! '{ "bsm": [] }'
15: Received! '{ "bsm": [] }'
16: Received! '{ "bsm": [] }'
17: Received! '{ "bsm": [] }'
18: Received! '{ "bsm": [] }'
19: Received! '{ "bsm": [] }'
[Finished in 16.5s]
```

- When messages are being sent.

```
200
Receiving BSM from WebSocket...
0: Received '{ "bsm": [{
  "entityID" : "11",
  "latitude" : 37.1234567,
  "longitude" : -80.1234567,
  "heading" : 0.0,
  "speed" : 0.0,
  "airTempDegC": null,
  "roadTempDegC": null,
  "rseID" : 11,
  "brake" : 0,
  "timestamp" : 1438988681543
},{
  "entityID" : "10",
  "latitude" : 37.1234567,
  "longitude" : -80.1234567,
  "heading" : 0.0,
  "speed" : 0.0,
  "airTempDegC": null,
  "roadTempDegC": null,
  "rseID" : 11,
  "brake" : 0,
  "timestamp" : 1438988681627
}] '
```

- Receiving information like ID, longitude, latitude, location etc. for Dynamic Messages Signs (DMS)


```
{
  "id": 1113,
  "name": "StauntonDMS181664",
  "uniqueID": "RE1TAAAAARZ=",
  "latitude": 37.258141,
  "longitude": -79.045183,
  "direction": "north",
  "geoSection": "Staunton",
  "link": "http://www.vdotdatasharing.org/xmldb/DmsSignStatus/StauntonDMS-StauntonDMS181664",
  "location": "I65-220",
  "timeRegionData": null,
  "mileMarker": 1.0,
  "signsState": "off",
  "message": "[pt25o5]{jp3}[l13]ROAD WORK[n12]{jl13}I-581 NORTH[n12]{jl13}MM 2 - 3[np][pt25o5]{jp3}[l13]LEFT[n12]{jl13}LANE[n12]{jl13}CLOSED",
  "roadway": null,
  "timeID": 171668,
  "dateInitial": null,
  "dateExpired": null,
  "category": null
},
```


- For getting the information of potholes

```python
import requests

# Importing JSON library for encoding and decoding the data
import json

# response is variable name. It could be any name.
# GET is a method used to fetch the data from the mentioned API
# In auth, we are providing credentials (key and password)
response = requests.get('https://vcc-api.vtti.vt.edu/api/pothole?status=[open|closed|all] & start_date=mm/dd/yyyy& end_date=mm/dd/yyyy',
auth=('aq0akwkcgjv2zwpdmB5u3Yq/erd//tNJauCTmfuV/9so9rWjOp482LVJ', 'goingToBeIgnored'))


# .status_code gives us the connection detail.
# If answer is 200, it means everything went okey and it is ready.
print(response.status_code)

# Encoding and decoding the json data and printing it at last.
# Printing the content in a way that is readable and meaningful using indent.
#Indent creates the space and each item is printed separately.
string = json.loads(response.text)
print(json.dumps(string , indent = 4))
```

- Receiving information like id, severity, latitude etc. of potholes


```
200
[
    {
        "id": 527,
        "phoneID": 429135,
        "latitude": 38.9122145,
        "longitude": -76.9826828,
        "heading": 221.725,
        "timestamp": 1509732607962,
        "speed": 3.04,
        "severity": 417,
        "confidence": 0.902168,
        "clusterID": 278,
        "isResolved": false
    },
```


- Receiving autogenerated email of the report of servers

Daily report of servers Inbox x

hgmehta1@gmail.com 12:08 PM (0 minutes ago)
to me, hmehta12

Dear team,
Please find the attached daily report of yesterday.
Thank you!

Team ITS

Station ID	Location	Sensor type	IP address	Count
ADR1	I-395 Bridges	Loops	10.40.63.250	23
ADR2	I-395 Bridges	Loops	10.40.63.250	20
ADR3	I-395 Bridges	Loops	10.40.63.249	18
ADR4	Theodore Roosevelt Bridge (I-66)	Loops	10.40.63.245	20

Reply Reply all Forward

- SQL query that fetches the required data only from the large database

```
SQLQuery10IM.sql - ...dministrator (180))*
SELECT TSH.IntId, TSH.RawVehicleCount, TSH.SampleTime, TS.Name
FROM TransSuiteHistorical.dbo.FMS_DetectorStationData_20190727 AS TSH JOIN TransSuite.dbo.FMS_DetectorStation AS TS
ON TSH.IntId = TS.IntId
```

- Displaying the data, we required.

Results | Messages

	IncId	RawVehicleCount	SampleTime	Name
1	8346	37	2019-07-27 04:36:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
2	8346	16	2019-07-27 04:37:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
3	8346	40	2019-07-27 04:38:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
4	8346	22	2019-07-27 04:39:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
5	8346	21	2019-07-27 04:40:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
6	8346	49	2019-07-27 04:41:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
7	8346	47	2019-07-27 04:42:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
8	8346	15	2019-07-27 04:43:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
9	8346	49	2019-07-27 04:44:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
10	8346	28	2019-07-27 04:45:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
11	8346	33	2019-07-27 04:46:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
12	8346	22	2019-07-27 04:47:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
13	8346	22	2019-07-27 04:48:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
14	8346	41	2019-07-27 04:49:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
15	8346	20	2019-07-27 04:50:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
16	8346	26	2019-07-27 04:51:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
17	8346	17	2019-07-27 04:52:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...
18	8346	30	2019-07-27 04:53:00.000	I-295/DC-295 0.13 Mile South of Benning Rd (Thro...

Query executed successfully.

SIGNIFICANCE

- Developing programming scripts help us in
  - improving real time situational awareness
  - vehicle tracking
  - pothole detection
- Developing programming scripts strategically enhance operations via collected data.
  - Basic Safety Messages are collected and utilized appropriately.
  - Message being displayed in Dynamic Message Sign can be tracked.
- Communication with all the stations are monitored on daily basis so if a station is not working, then it can be detected, and necessary action can be taken.
- Monitoring SQL server database daily helps us know whether we are continuously receiving data or not from all the stations.

ACKNOWLEDGEMENT

- Extreme gratitude to Ms. Kelli Raboy, Mr. Jason Tao and ITS team for giving me an opportunity to work on live projects along with their guidance and constant feedback that helped me improve my programming skills
- I would also like to thank the District Department Of Transportation as well as Howard University Transportation Research & Data Center for providing me a platform to exhibit my professional skills.

REFERENCES

- Multifunctional OBU Deployment – ITS Maryland annual meeting, November 8, 2017
- VCC External communication system – Virginia Tech Transportation Institute, October 2017

Harsh Mehta

Database development & management Intern

Email : harsh.mehta1@dc.gov

Software Engineering Grad student at University of Maryland-College Park

Intelligent Transportation System (ITS), District Department of Transportation

Washington, District of Columbia

District Department of Transportation