

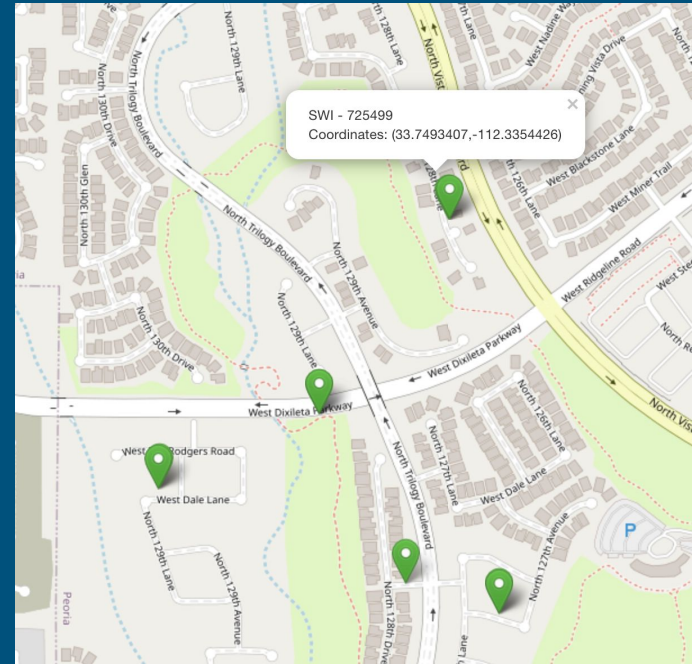
AI-VVO sdmay22-36
Spring 2022
Update #4

2/17/2022 - 2/24/2022



Front-end (This Week)

- Worked with the back-end team to try and call an API to return the coordinates of bus nodes onto the map-based grid
- Researched axios get request
- Tested markers with exported csv files and with the api url



Front-end (Next Week)

- Continue to work with the back-end team to get end-to-end communication fully working with markers
- Start researching how to send data to the back-end through inputted values from the front-end
- Implement svg files as custom icons

Back-end (This Week)

- Created python scripts to upload and pull data from the neo4j db.
- Created api /api/bus_nodes that will allow the frontend to grab all of the needed data from neo4j
- Only has node coordinates at the moment unsure where to get the connection information from
- Updated logging for django so that major issues will be logged for debugging
- Added a folder to keep all of the database scripts in one spot

Back-end (Next Week)

- Find out where to get the connection information from the nodes
- Upload the connection data to make the node data more meaningful
- Update current api to return both coordinates and connections at the same time.

Documentation (This Week)

- Do final tweaks on documentation
- Merge branch

Machine Learning (This Week)

- Wrote BusCoordsExport.py script to give data for the front-end for map display
- OpenDSS dss.text command used to export bus coordinates to CSV for parsing by the front-end team for map-based display

```
GridTesting.py x BusCoordsExportScript.py x
1 import os.path
2
3 import py_dss_interface
4
5
6 #Generate the dss instance with the given data
7 current_path = os.path.abspath(os.getcwd())
8
9
10 open_dss_path = "C:\Program Files\OpenDSS"
11
12
13 dss = py_dss_interface.DSSDLL(open_dss_path)
14
15
16 dss_file = current_path + "\\CLW13_March15_v6\\Master_v6_March15.dss"
17
18
19 dss.text("compile {}".format(dss_file))
20
21
22 dss.solution_solve()
23
24 #Export the buscoords for the front-end map
25
26 dss.text("export buscoords")
```



Machine Learning (Next Week)

- Write script to pull data from bus node locations in the OpenDSS model into the Neo4J database
- Create Python code to push connection data from the grid model to the Neo4J database (relational data is vital for the Neo4J database)
- Script to pull data (reactive powers at all buses) from the backend into the machine learning application as control signal