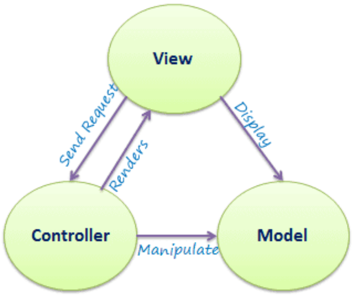
EleNA system

Group Members: Venkata (Dennis) Billagiri, Shreya Sawant, Dhruv Keyal, Anushree Jana, Aditya Vikram Agarwal

Software Architecture



The EleNA system is implemented using the MVC architecture pattern. Using this, we base the software architecture on Model, View and Controller. We create four separate folders to satisfy this requirement - with them being Model, View, Controller (the test cases being inside the Controller folder).

1. **Model** uses config.py and graph.py which holds the graph generation process using OSMnx.

2. **View** uses a directory structure which includes “src” and “public.” This uses a React Application with Mapbox GL JS (API) and Material UI for user experience, and provides the front-end framework of the EleNA system. It displays the map component of Amherst, MA. Running the application allows the user to enter in the start and end location, and choose minimization/maximization. These functionalities are a shell right now as they need to be implemented with the back-end using flask web framework. A screenshot of the map component can be found under View in map.png. There also exists a package.json which has all the dependencies for the front-end part of the application.

3. **Controller** uses 6 files for implementation and 1 test case file. dijkstra.py and astar.py are developed for their respective algorithms. search.py is used to find the path search between two nodes. This search.py is used as a super class which the two search algorithms extend. shortest.py is implemented to find the shortest path in the graph. utils.py is used to find cost and elevation for graph nodes with caller.py being implemented to call the rest of the files using shortest.py. The last file here is the test file, test.py which does extensive testing of the back-end part of the application.