Make specifications specific, not ambiguous. Large - how large? Efficient - how efficient?

The final product is not just the software, but also the documents.

Reconnect with our client.

Attendance is important this semester. First class, not a big deal.

Ideally, both semesters will be the same grade.

Broad plan moving forward:

Scrape or sort data—------*latitude and longitude*

Create database—-----------*lat and long conversion to addresses (Reverse geocoding)*

Connect codespace to database through queries.

Once data can be fetched, calculate route, distance, time between the two points.

Once routes can be determined factor in typology requests

Comments from professor: (Need to work on requirements)

Overall, the structure was good - but still have several issues.

Ambiguous requirements - 1.1.2 (appropriate, available data, )

On a scale of 1-4, 1 being the least stressful and 4 being the most stressful:

1 - Neighborhood Street (incl. LN)

2 - Neighborhood Connector (incl. NB, CB, NN, EN)

3 - Sub-Urban Connector (incl. CC)

4 - Priority/Main (incl. UC, E/F, MS)

Flowchart:



