Technical Requirements

Team:

- Samir Lavingia
- Alexei Naumann
- Jeff Kang
- Ian Whelan
- Aidan Blant

Database Design:

- Data that will be put into the database:
    - All the data parsed from the JSON files in terms of car speeds, locations, etc.
- Use MySQL for data entry and retrieval
    - All messages received from server stored for fast retrieval
    - Must implement SQL, and must connect to it using SQL statements

---

Data Export:

- It should take the important data from the database and export it
- We want it to export to a comma delimited file that can be opened in a nice format in excel
- Will export:
    - Car Data:
        - How many cars were on the road over the course of execution
        - Most traveled highways
    - Travel Times:
        - This will be for all the nodes that were selected during the life of the program to all the other ones
        - Specifically:
            - Travel time at certain times of the day
            - Average travel times
            - Slowest highways on average

Map Design:

- (-- JavaScript --) (++ Java ++) Map will be embedded into standard Java application (-- (maybe?) --)
- Graphics will be overlayed on top of Map image
    - Cars should appear to be moving on top of map image
- Trace paths on map for navigation
- Map of LA
    - Integrate a mapping API to display map to user
        - Zoom-in and Zoom-out features
    - Overlay nodes for practical movement of data
    - Display flow of traffic in the form of individual cars
        - Display cars in varying colors depending on latest speed (relative to speed limit)
    - Start with map centered around downtown LA
    - Keep map up-to-date as server provides data

Car Design:

- List of cars and all pertinent data
    - Car speed / direction
    - Car location
- Cars will have different colors based on speed
- Cars can be clicked on/hovered over to see their actual speed in traffic rather than just a color code

---

Server Design:

- It will pull and parse the JSON into a datastructure (most likely ArrayList of Cars)
- This will be done on a certain time interval
- This data will be pulled by the database and the client to get the information on the cars that will be overlaid on the maps
- The server will run in the same program, just on a different thread

---

Client Interaction:

- Map Design:
    - Graphical Interface that displays map and car locations to user
        - Displays flow of traffic as individual cars moving at their listed speeds

        - Areas of heavy traffic represented with red cars
        - Areas of light traffic represented with green cars
    - Display fastest routes from specified starting and ending location
        - Method to select starting and ending locations
            - We will most likely overlay nodes (small ImageIcons) on the map that can be selected in order to do this
        - Provide estimated time of journey at speed limit and current speed of traffic
- Interface to display historical data from database
    - Allow for determinations such as best time to travel from source to destination
- Interface to display times and best paths
    - After choosing two points on a freeway a path will be drawn between them with the best route to take
        - We can choose between shortest amount of time or shortest physical distance
            - The time will be based on the historical data
        - You can also choose a third middle point to stop at for a certain amount of time
            - This will calculate the amount of time to get there, stop for the amount of time, then start again heading to the end location

---