Catherine Horng, ch756

Data Science and Optimization for Good:

Algorithm and Software Development for Green Fleet Transition to Zero Emission Transportation

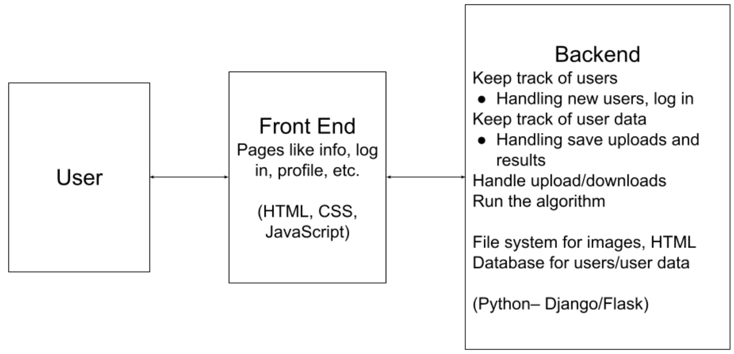
Name: Catherine Horng
Advisor: H. Oliver Gao

Air pollution is a significant and prevalent environmental hazard, having been a constant source of concern for the ecosystem, public health, and social welfare. As research into air pollution and emissions continue, it is seen to be of increasing danger to public health; in particular, it has shown to be linked to numerous health concerns such as respiratory disease, cardiovascular disease, and cancer. One of the primary sources of the emissions that can lead to these health concerns is vehicles. Vehicle transmissions have shown to be large contributors of nitrogen oxides (NOx), particulate matter (PM), hydrocarbons (HC), and other damaging pollutants found in our atmosphere. However, there is an increasing need and a continuation for the reliance of these vehicles. The massive amount of pollution caused by vehicles coupled with our dependence on the form of transportation naturally leads to the question: How can we continue maintaining and using these vehicles while reducing the amount of emissions we put in the air?

This question has been the focus of research done by Professor H. Oliver Gao and Timon H. Stasko. Fleet management requires sizable resources and fleet managers very naturally are interested in minimizing the cost of maintaining these fleets. With the multiple decisions, constraints, and uncertainties involved in making decisions about costs and maintenance regarding fleets, this minimization is not a simple task. In the previous work done, an optimization model has been developed to aid in the decisions of vehicle purchase, resale, and retrofitting while also factoring in emissions. This optimization model produces optimal policies for minimizing long-term and short-term costs as well as minimizing certain emissions while evaluating the potential environmental and emission impact the vehicle fleet will have. With the increasing concern of emissions to our health, emission regulations have also begun to be put in place; thus the optimization model can also form policies that comply with these regulations. Different regulations may affect the value of certain assets, so besides producing these policies, the model may also estimate the value of each asset in the fleet, thus allowing the fleet manager to evaluate the increasing or decreasing fleet value.

With the developed optimization algorithm, there is currently no method for it to be integrated into the daily management of vehicle fleets and assets. While software was developed and can be used to form and run these optimization models, there is a lack of tools to utilize them. Therefore this project aims to develop software tools in order to manage vehicle assets and fleets while simultaneously employing the described developed algorithm for green fleet management in order to facilitate the use of this optimization tool into the regular management of vehicle fleets and assets. The goal of this project is to have a beta version of green fleet management software done; we achieve this goal by developing an web-based asset management tool for vehicles with an integrated optimization model formulated to minimize cost or emissions based on a user built fleet. The web-based software offers advantages like accessibility to the public while still maintaining full functionality of the optimization model.

In order to build the web application, research was done into the development of software application architecture. Modern web application architecture consists of the frontend and the backend. The frontend is what the users see and interact with; this consists of collecting data from the user and displaying the results to the user. The backend contains all the application logic; the frontend will request information and data from the backend and the backend will respond. The backend is also responsible for maintaining the file systems in which the application stores the information as well as the databases.



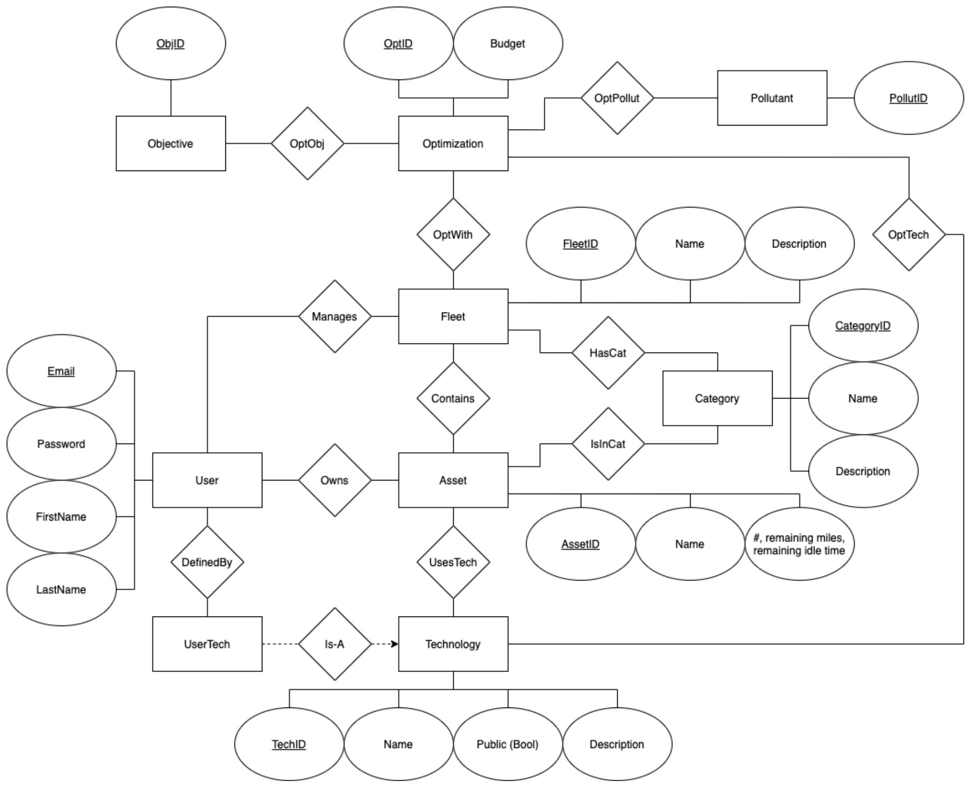
First, we began with the design of the underlying database structure of the website. The database is part of the backend structure of the web-application and contains all the user and website data. The application will request data such as user information, fleet information, and asset information from the database, so the design of the database itself is very important in order to best represent the objects needed in order to perform processes such as optimization. In designing the database structure, we want to model what the database is about conceptually by representing the data and the relationships between the data. One way to represent this data and the relationships is with an entity-relationship (ER) diagram. One analogy with an ER diagram is to think of entities as nouns and relationships as verbs. With fleet and asset management, there are many different entities involved including users, assets, and fleets. These entities can all hold different properties such as age of an asset or the type of retrofit or technology used. The different relationships in asset management include which user an asset is owned by or what assets a fleet contains.

The entities are:

- Users: users of the asset management system, stores information such as id, name, email, password, and other information needed to ensure user privacy
- Assets: assets in the management system, stores asset properties such as number of asset of this type, retrofit, and emissions
- Category: categories of assets, stores information such as common information of an asset of this category
- Fleets: fleets of the system, stores information such as assets contained in the fleet and categories contained
- Technology: technologies, or retrofits, available, contains library of technology predefined and user defined technology, cost of switching to this technology
    - User-Defined Technology
- Optimization: optimization done on fleets; stores information such as recommendations of switching retrofit, what to optimize, budget constraints, and regulations

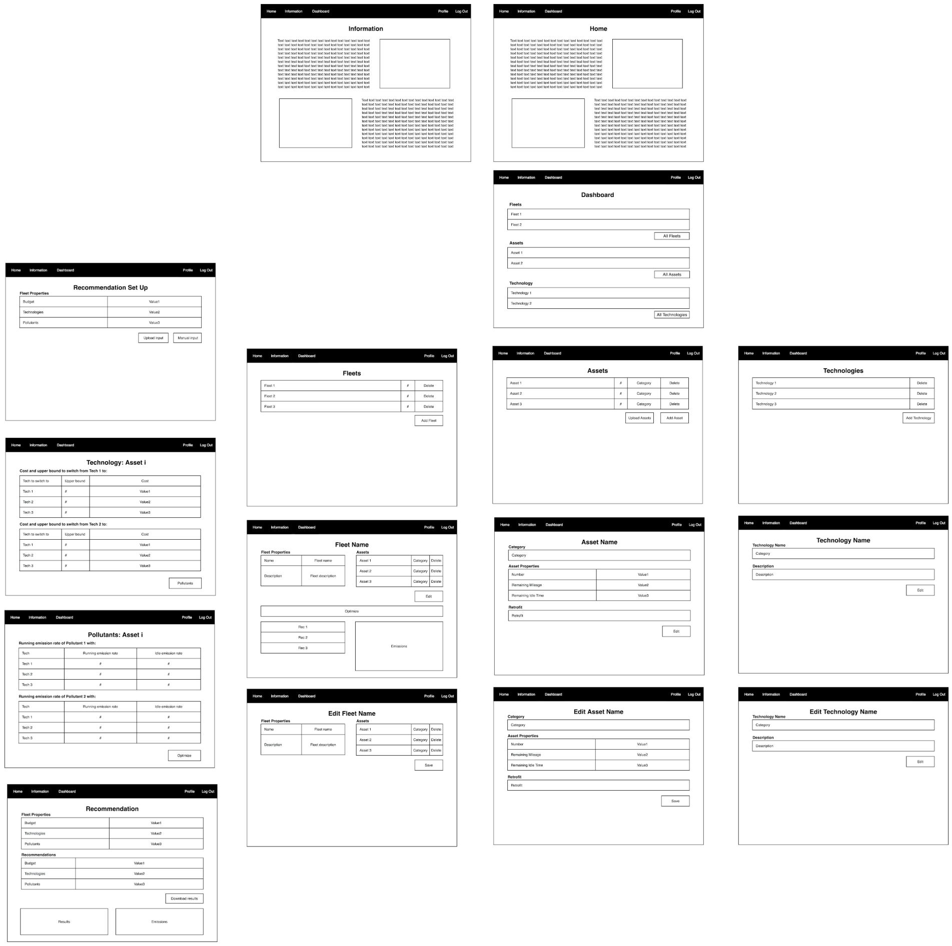
Between the entities, the relationships are:

- User owns Assets: maps which user owns which assets
- User manages Fleets: maps which user owns which fleets
- User defines Technology: maps which user defines which technologies
- Asset uses Technology: maps which assets uses which technologies
- Asset is-in Category: maps which assets uses which categories
- Fleet contains Assets: maps which fleets uses which assets
- Fleet optimized with Optimization: maps which optimizations optimized which fleets



Each entity and relationship requires its own table. Relationship tables contain the ID of the connecting entities. This design ensures scalability; in this way, a user can have a theoretically unlimited number of assets, fleets, technology, and so on. Additionally, a fleet may contain multiple assets and different fleets may contain the same asset so that optimization can be run on different configurations of fleets.

After designing the database structure, we can now begin designing the structure of the web-application itself. In order to build an efficient web-application and establish a structure to the application, a wireframe for the website was developed. Wireframing is a process in which an overview of the application and its interactive features are drawn out in order to establish the structure and flow of the application. This allows the design of the website to be laid out before building the website to ensure that all features and functionality can be met and aid in building the back-end of the application.



Looking at the above figure, the features included in the website include:

- User management: allowing users to sign up and log in to manage their assets, fleets, optimizations, etc.
- Asset management: adding, deleting, editing, uploading, and viewing assets
- Fleet management: adding, deleting, editing, uploading, and viewing fleets; adding and removing assets to fleets
- Optimization: running optimization strategies on fleets

Finally, we began building the website. For what language we would like to use to build our website, we chose Python. Python has a few web frameworks that have been built for the language and because of its ability to solve complex problems (which is needed for solving the optimization problem), I chose to build the web-application with Python. For this project, Flask, a micro web framework written in Python, was used. Flask is a simple to use framework with minimalistic features that allows for native database handling. Using this framework, we build the features described in our wireframe for asset management.

User Management: The website keeps track of users, their names, email addresses, and passwords. Users may sign up for an account, login, and logout. Each user's assets, fleets, technologies, optimizations, etc. are all saved and can be viewed anytime after logging in.

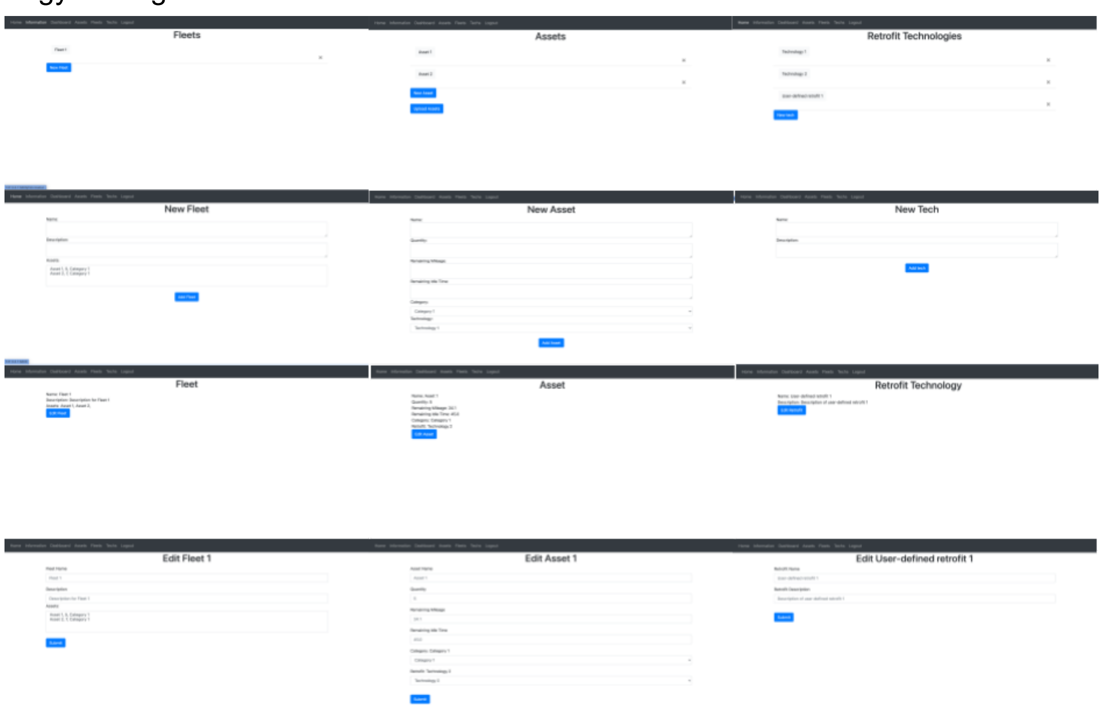
Asset Management: Users can create new assets, view assets, edit assets, and delete assets. Assets have properties such as quantity of asset, remaining mileage, remaining idle time, category, and retrofit used. The above properties are all required for the optimization problem besides category.

Fleet Management: Users can create new fleets, view fleets, edit fleets, and delete fleets. Fleets have properties such as name, description, and assets contained within the fleet. Fleets are used as a grouping in order to perform optimization on.

Retrofit Technologies: Users can create retrofit technologies, which have properties such as name and description. These technologies are what the optimization features recommend to the users, which leads us to the next feature. Retrofit technologies may have user defined technologies as well as technologies available to anyone.

Optimization: Users can optimize an owned fleet. By defining the optimization objective (short-term budget, long-term budget, emission reduction), providing costs of switching to different retrofits, and defining pollutants to reduce and their emissions as a result of switching to different retrofits, users can use the software to aid in the decision making of fleet management. The model will output recommendations such as switching some number of a certain asset to use a certain retrofit while remaining within budget.

The following are a few screenshots of the beta version of the asset, fleet, and technology management sections of the website:



The project is still in its preliminary workings of the website and is still in its beta stage of development. The asset management capabilities function as needed while the optimization capabilities are still in development as discussed above. However, the design of the website will aid in facilitating the use of the previously developed algorithms and optimization models. Following implementation of optimization, next steps include data visualization, user experience design, and extension to other asset management problems. For example, dynamic optimization with time periods and general multi-asset management problems such as green building management can be formulated.