PRODUCT DESIGN

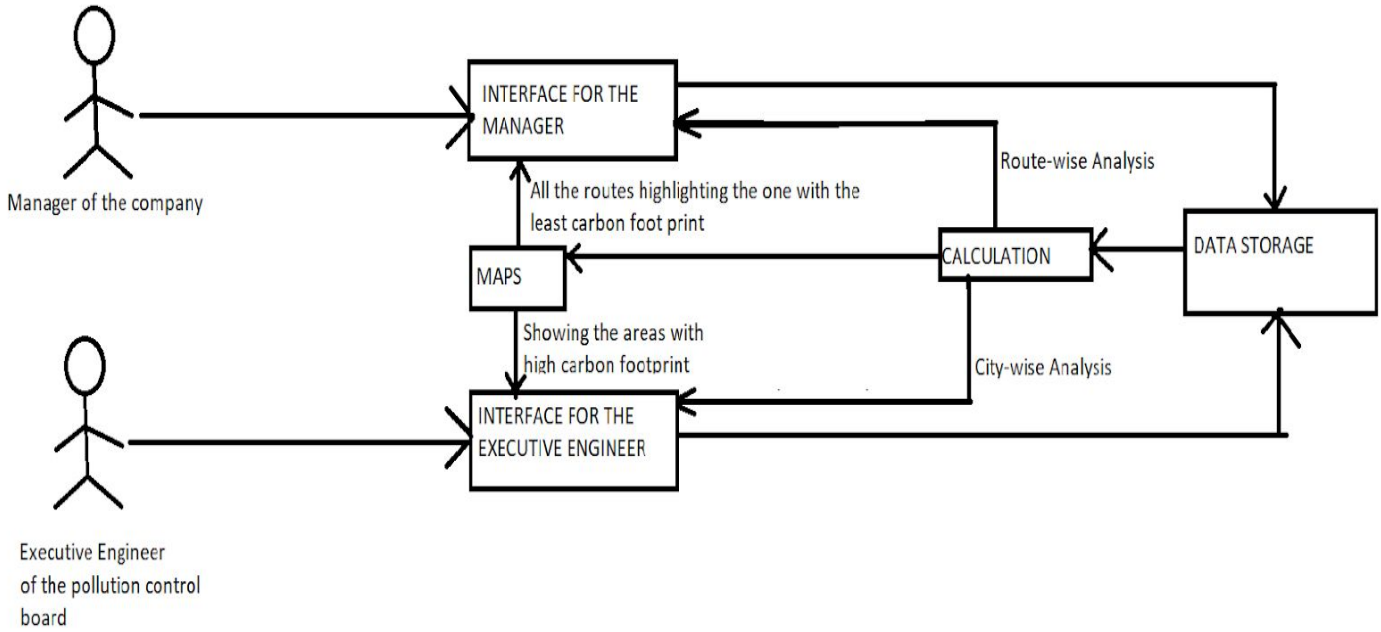
Team

TEAM NUMBER - 38

TEAM MEMBERS - Kushagra Agarwal
Shreeya Pahune
Sriharshitha Bondugula
Sravani Dama

Design Overview

Architectural design



System interfaces

User Interface

1) **Register :** User(Manager/Engineer) has to register by providing his details when he uses the system for the first time.
2) **Login :** User can login using the credentials given during the registration. Once he logs in, he can access all the features.
3) **Upload data :** Manager and the engineer upload the data in the format necessary to get the desired details using this feature.
4) **Route-wise and City-wise Carbon footprint :** The carbon footprint is calculated based on the given data and displayed correspondingly according to the user.
5) **Shortest path between nodes :** This is a use case of the manager. When the nodes are specified the shortest path between both the nodes is shown.
6) **Company's Carbon Footprint** : This is a use case of the manager. Based on input provided by the company's manager statistical analysis of data is shown.
7) **Pre-emptive Carbon Emission Calculation :** This is a use case of the manager. Given the number of trucks,source and destination as input, output the calculated carbon footprint emission.
8) **Map generation :** This is a use case of the manager. Given the input as source and destination generate the map which displays the route with least carbon footprint.
9) **Identifying hotspots with high carbon emission :** This is a use case of the engineer. Based on the input data displays the regions/routes with highest carbon footprint.
10) **Logout**

APIs

The following APIs are exposed to the users to enable them to interact with the system;

1) HTTP (HyperText Transfer Protocol)
2) OpenLayers (JavaScript - used to put a dynamic map in any web page. It can display map tiles, vector data and markers loaded from any source)

Model

Draw a simple class diagram and describe the classes in the table in this section. This diagram should represent the classes and their relationships. It is only necessary to show methods that are publically accessible by other classes. Only show an instance variable of a class if it is publically accessible. The diagram and the table should be consistent with each other.

Identify the classes (logical groupings of software methods that provide a related set of services). Make sure the design conforms to good design principles.

For each class, specify the information it maintains and the functionality it provides. Provide sufficient detail so that the purpose of each class in the design is clear.

<Class No. 1>	eClass state • What information is the class responsible for maintaining? • E.g., a printing subsystem might hold the current status of all the printers it controls as well as the queue of print jobs waiting to be printed. Class behavior • What methods does the class implement? • E.g., classes related to the printing subsystem might support the queuing up of new jobs, estimating the time until a given job completes, or emailing status information at the end of a job.

<Class No. 2>	Class state • What information is the class responsible for maintaining? Class behavior • What methods does the class implement?
<Class No. 3> add more rows as needed.	Class state • Etc. Class behavior • Etc.

.

Sequence Diagram(s)

Design Rationale

1) **We have chosen 2 different UIs for the 2 users we have; the manager of the company and the executive engineer of the pollution control board.**

   That is because of two of them have different goals and different use cases. The manager needs the route-wise analysis of the carbon footprint whereas the engineer needs the city-wise analysis of the carbon footprint. The input data format of the two users is different. So, we have decided to have two different UIs.

2) **Details about the shortest route and the route with minimum carbon footprint are shown independently to the manager of the company.**

   That is because the shortest route is necessary to know the expenditure based on the fuel consumption and the latter is to know the path to be taken to minimise the carbon footprint. Both necessarily need not be the same all the time and hence we have chosen to show them separately.