

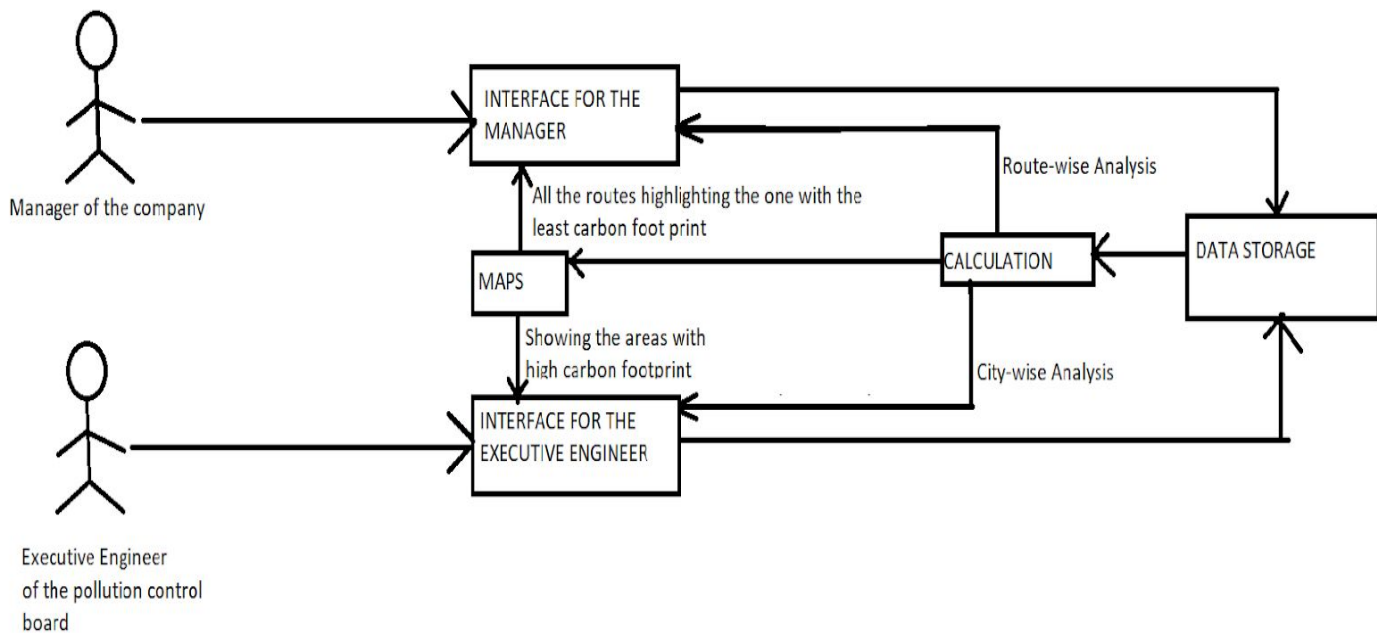
PRODUCT DESIGN

Team

- Number: 38
- Members:
 - Kushagra Agarwal
 - Shreeya Pahune
 - Sriharshitha Bondugula
 - Sravani Dama

Design Overview

Architectural design



System interfaces

User Interface

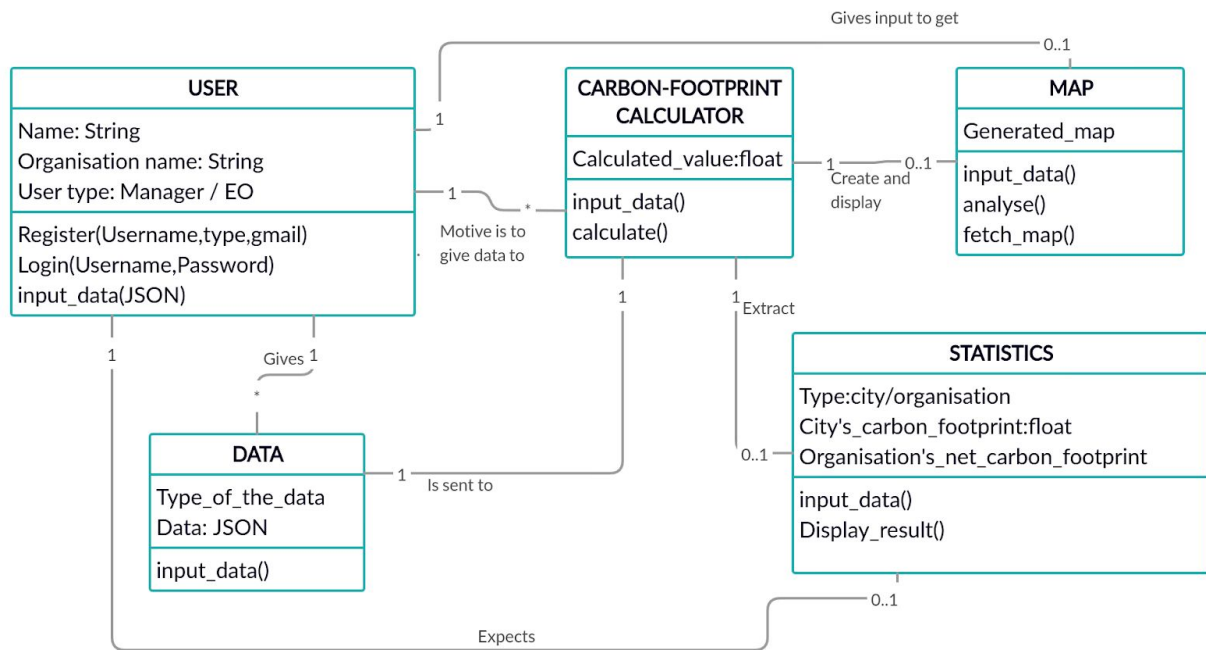
- 1) **Register** : User(Manager/Engineer) has to register by providing his details when he uses the system for the first time.
- 2) **Login** : User can login using the credentials given during the registration. Once he logs in, he can access all the features.
- 3) **Upload data** : Manager and the 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



USER

Class state:

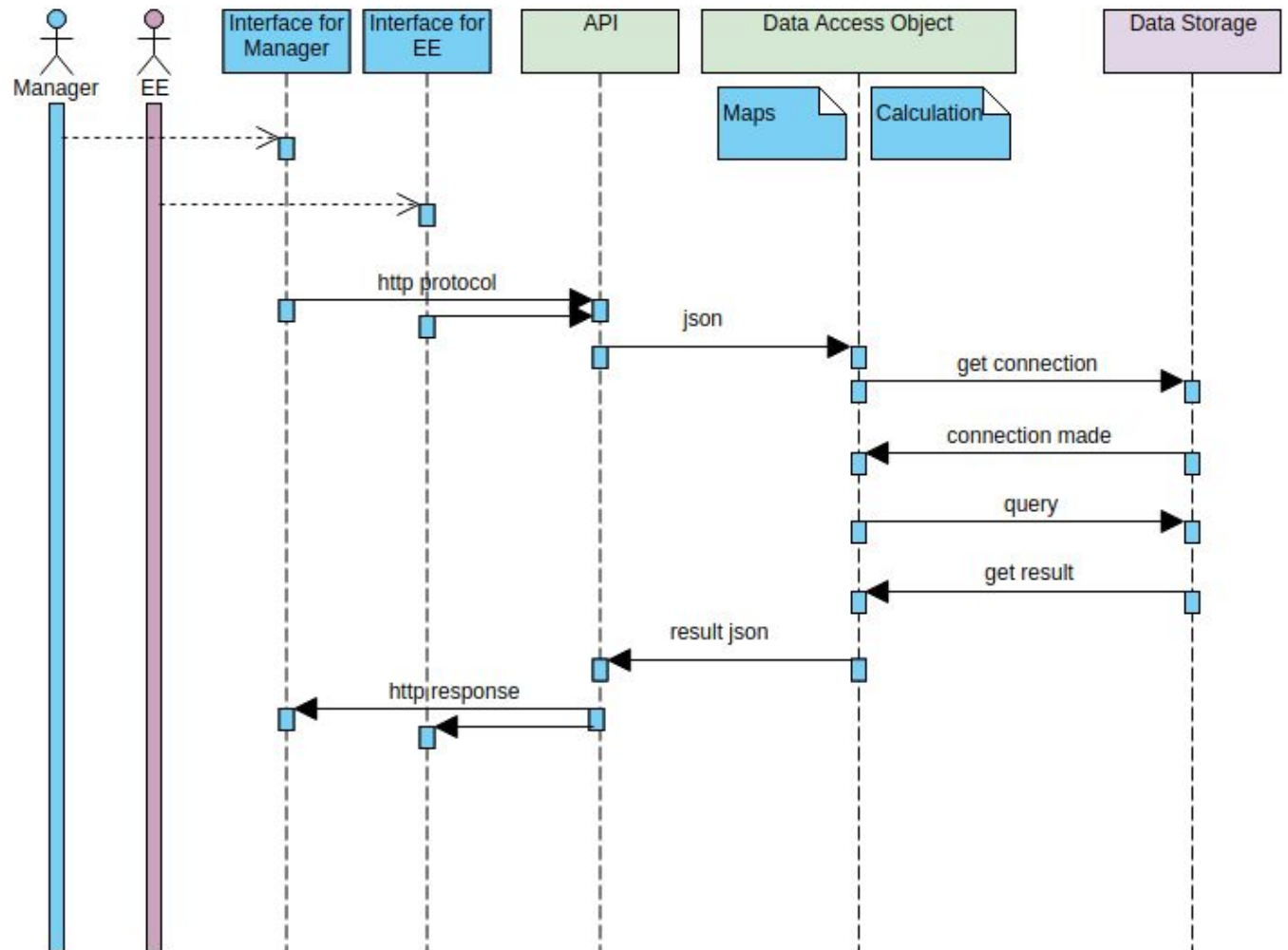
- Maintains the information of the user (Name, Organisation)
- Also holds the information about the type of the user. (Manager or Executive Officer)

Class behaviour:

- This class supports logging in of the user.
- It supports registration of the user and taking input data from the user.

DATA	<p>Class state:</p> <ul style="list-style-type: none"> • Maintains the data • Also holds the type of the data (i.e given by which user) <p>Class behaviour:</p> <ul style="list-style-type: none"> • A method that adds the inputted data to the database is called from here.
CARBON-FOOTPRINT CALCULATOR	<p>Class State:</p> <ul style="list-style-type: none"> • It holds the calculated carbon footprint of the given data. <p>Class behaviour:</p> <ul style="list-style-type: none"> • A method that converts JSON file to calculable format is also supported. • A method that calculates the carbon footprint is called here.
MAP-GENERATOR	<p>Class State:</p> <ul style="list-style-type: none"> • It holds the visualised map for the calculated data. <p>Class behaviour:</p> <ul style="list-style-type: none"> • A method that creates a map for a given data is called here that also supports getting data from the previous class. • It finally supports fetching/displaying the resulting map.
STATISTICS	<p>Class State:</p> <ul style="list-style-type: none"> • It holds the type of statistics that is being expected (City's or Organisation's statistics). • It also holds the calculated values for a given data. <p>Class behaviour:</p> <ul style="list-style-type: none"> • It calls the methods that display the expected statistics that also supports inputting data from the carbon footprint calculator class.

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 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.