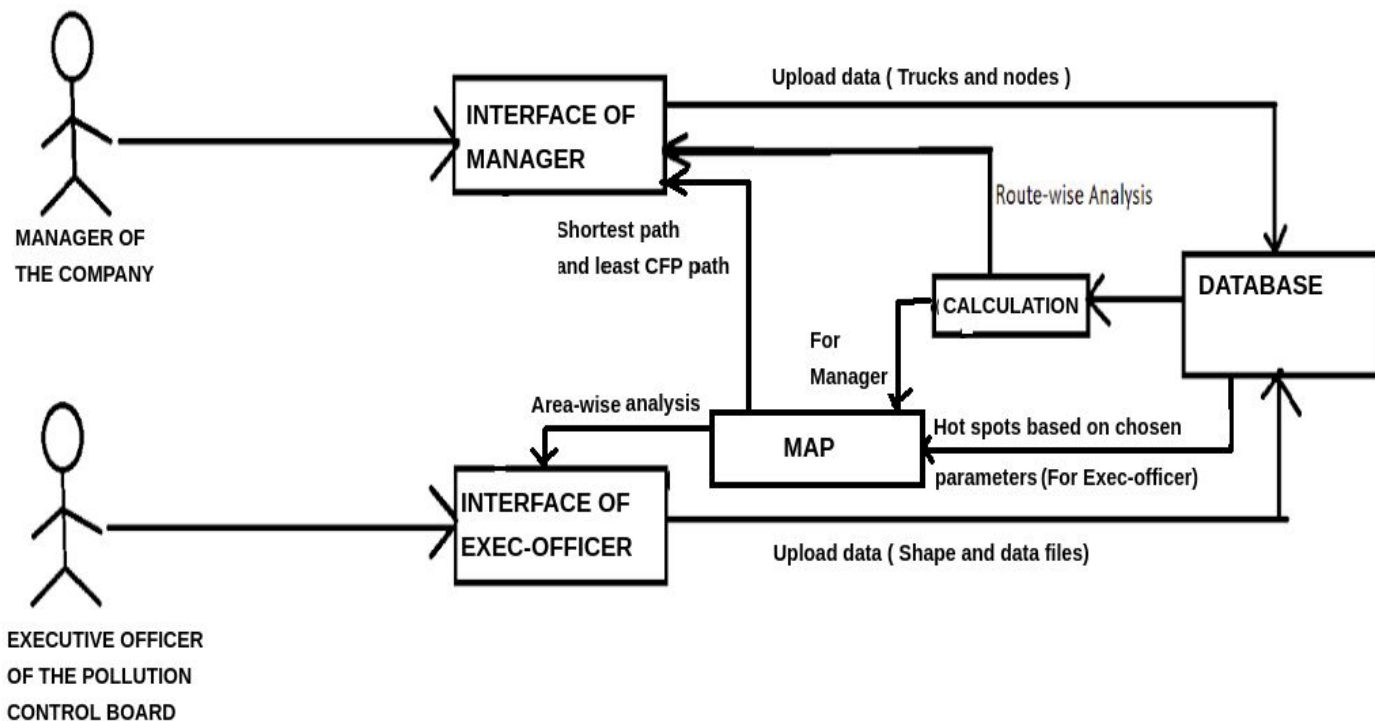


PRODUCT DESIGN

TEAM NUMBER	38
TEAM MEMBERS	KUSHAGRA AGARWAL SHREEYA PAHUNE SRAVANI DAMA SRIHARSHITHA BONDUGULA

Design Overview

Architectural design



System interfaces

User Interface

User is first asked to login if he wants to make use of the functionalities in the app. He has to Register first inorder to login or he can also login through facebook or gmail.

Once, the user is logged in he can make use of the following functionalities of the app..

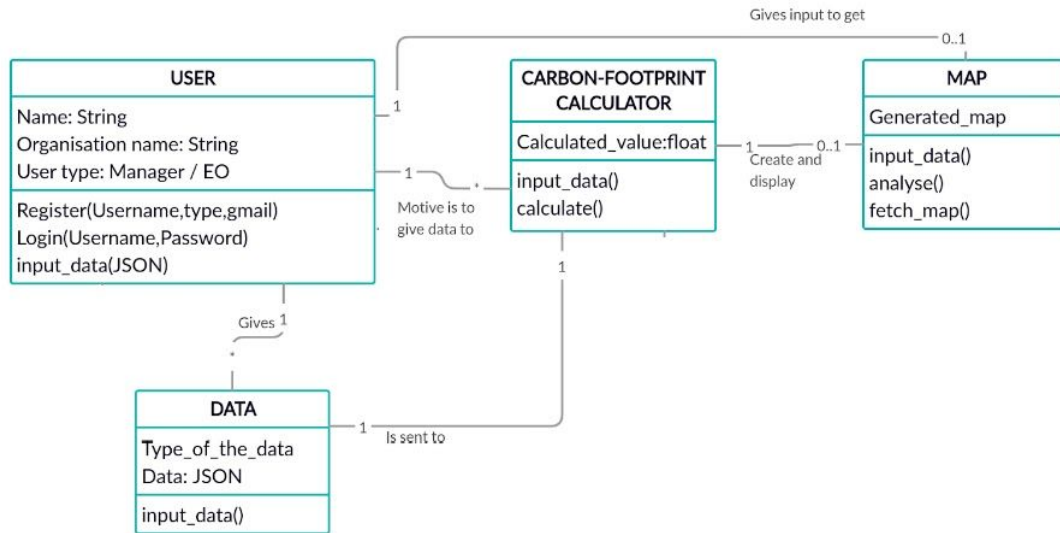
- 1) **User Registration** - Users should be able to register specifying their usertype.
- 2) **Login** - Registered users should be able to login to make use of the app.
- 3) **Upload Data** - Users should be able to Upload the data necessary to make the calculations.
- 4) **Shortest path between nodes** - Given source and destination as input,find the shortest path between these two and display in the map
- 5) **Path with least CFP** - Given the number of trucks , source and destination as input,find the path with least CFP and display in the map.
- 6) **Route-wise CFP calculation** - Based on input provided (Number of trucks) and path chosen, net CFP for that path is printed.
- 7) **Net CFP of the company** - To calculate the net carbon footprint of the company in order to take the necessary measures.
- 8) **Choose nodes from a Map** - Allow the user to choose nodes from the map.
- 9) **Log or history** - Storing all the paths chosen and printing the log.
- 10) **Preemptive CFP calculation** - Based on input provided (Number of trucks) and path chosen, net CFP for that path is printed.
- 11) **Identifying hotspots with high carbon emission** - Based on the input data, display the hotspots with respect to the selected parameters.

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

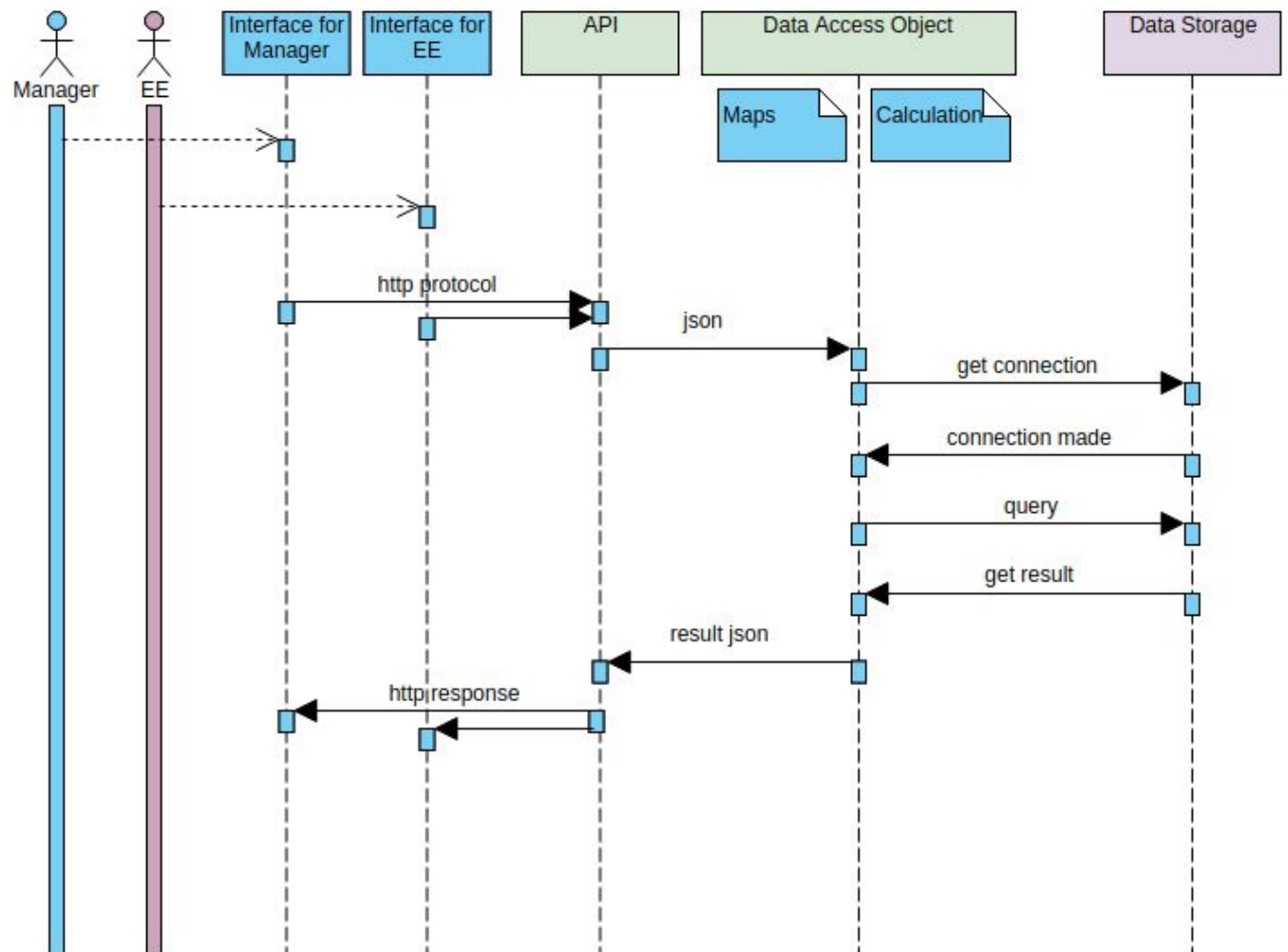


USER	<p>Class state:</p> <ul style="list-style-type: none"> • Maintains the information of the user (Name, Organisation) • Also holds the information about the type of the user. (Manager or Executive Officer) <p>Class behaviour:</p> <ul style="list-style-type: none"> • This class supports logging in of the user. • It supports registration of the user and taking necessary details 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 files 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.

.

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.

- 3) **We started implementing the use cases with a static data which is available (London osm data).**

This is because we need some data to work on and the only data available and open is London data and hence we chose and started working on it.

- 4) **We have decided to implement the route-based functionalities on a few areas first and then we have decided to extend it to other areas.**