Software Requirements Specification (SRS) Document

PROJECT NUMBER	**4**
PROJECT NAME	**Quantifying carbon footprint for logistic hubs**
TEAM NUMBER	38
TEAM MEMBERS	Kushagra Agarwal Shreeya Pahune Dama Sravani Sriharshitha Bondugula

Brief problem statement

Quantifying carbon footprint using geospatial data

Global climate change is adversely affected by the increasing amounts of greenhouse gases being released. Carbon footprint is a measure of the amount of greenhouse gases produced which can be in different sectors such as Agriculture, Industrial, Transportation etc. In this project our goal is to make a person handling a logistics hub to quantitatively understand the carbon footprint due to the truck movement in the hub. Using this information then suggest alternative routes using a pgRouting server to minimize this emission.

System requirements

Users require a browser capable of running JavaScript, handling HTML5 and CSS and an internet connection.

Developers require:

1. **Frontend** - Javascript (React JS - version 7.8.3) and Openlayers version 6(For Maps)
2. **Backend** - Flask (version 1.1.1)
3. **To maintain database** - PostgreSQL version 12.2 (along with an extension PostGIS) and pgAdminIV
4. **For routing and other network analysis functionality** - Pgrouting (An extension of PostgreSQL)
5. OAuth for Google Authentication.

Users profile

Many organizations and governments are looking for strategies to reduce emissions from greenhouse gases which are responsible for global warming. The quantitative assessment of the emissions in the form of carbon

footprint will be helpful to all such organisations and governments for example the State Pollution Control board.

The two main users of our system are:

1) **Managers of different companies** who are trying to reduce their company's net carbon footprint in accordance to their carbon credits.For example, they can use the app to find the routes with less CFP between two nodes (includes viewing it in a map).
2) **Executive Engineer in the Pollution Board** and related authorities in order to get an overview of the net carbon footprint of their city and major hotspots in the city.

It will be a simple app. Users mostly will be equipped with basic knowledge on the subject and related software, hence can comfortably use our service.

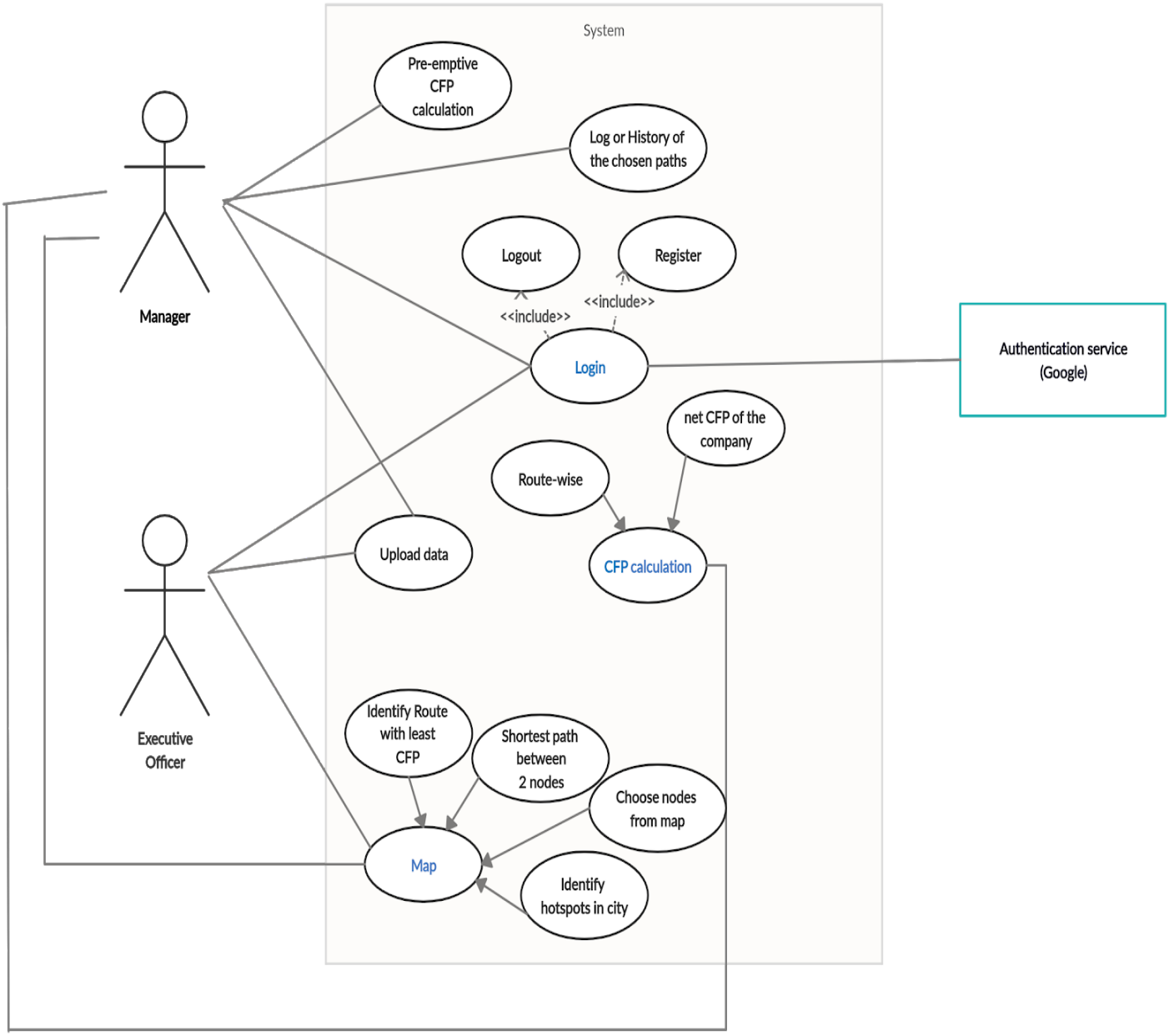
Instructions to help them make good use of services will be provided in the app. For example, instructions about input data format will be provided.

Feature requirements (described using use cases)

No.	Use Case Name	Description	Release
1	Registration	Users can register by providing details.	R1
2	Login	Users can login through their own accounts.	R1
3	Upload Data	The two types of users; manager of an organisation / company and executive officer upload the data.	R1

4	Shortest path between nodes	Given source and destination as input,find the shortest path between these two and display in the map	R1
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.	R2
6	Route-wise CFP	Based on input provided (Number of trucks) and path chosen, net CFP for that path is printed.	R2
7	Choose nodes from a Map	Allow the user to choose nodes from the map.	R1
8	Log or history	Storing all the paths chosen and printing the log.	R2
9	Net CFP of the company	Print the net CFP based on the paths in the log.	R2
10	Pre-emptive CFP Calculation	Given the number of trucks, source and destination, calculate the CFP.	R2
11	Identifying hotspots with high carbon emission	Based on the input data, display the regions/routes with the highest carbon footprint. (Heatmap).The hotspots or heatmap should be displayed parameter-wise. For eg, year-wise.	R2

Use case diagram



Login (includes Register, Logout) and Upload data are common use cases for both the users.
A **Manager** has use cases like Log - History, CFP calculation, finding the shortest path (with map) and the path with least CFP between 2 nodes (with map) and giving input from map (choosing nodes) and pre-emptive calculation.
An **Executive Officer** has use cases like finding the hotspots in the city (using map) and upload data.

Use case description

Use Case Number:	UC-1
Use Case Name:	Registration
Overview:	Users can register by giving their details.
Actors:	Both
Precondition:	None
Flow:	Main Flow: 1. Front-end support to let the user enter the details. 2. Maintain a database to store the details.
	Alternate Flow: 1. Error messages when fields are left empty. 2. Error when Username is repeated. 3. Error when the Email is not in the given format. 4. Error when the username is not unique.

Post Condition:	Using the User ID and password the user can login

Use Case Number:	UC-2
Use Case Name:	Login
Overview:	Users can login by providing the details that are provided during registration or through their google account.
Actors:	Users with valid login IDs.
Precondition:	The user should already be registered or should have a google account.
Flow:	Main Flow: 1) Front end support for accepting details and beginning the session based on the usertype.

	2) Validation of user's id and password with the help of data base. 3) Google authentication.
	Alternate Flows: 1. Proper error messages where ever entered details are wrong. 2. Error when fields are left incomplete. 3. Error when user is not registered.
Post conditions:	User's session is now active and the user can make use of the app's functionalities.

Use Case Number:	UC-3
Use Case Name:	Upload data
Overview:	Two types of users can upload the details. Managers can upload source node,destination node and number of trucks. Executive officer can upload the shape file and other required files.
Actors:	Managers and Executive officers
Precondition:	The uploaded files should be in specified format.

Flow:	Main Flow: 1) Front end support for uploading the files. 2) Commit the data into databases.
	Alternate Flows: 1. Proper error message when the format doesn't match. 2. Error when file name is repeated.
Post conditions:	The data gets pushed into database and can be used for further computations

Use Case Number:	UC-4
Use Case Name:	Shortest path between nodes
Overview:	Based on the start and end nodes taken as the input, the shortest path between them is found.
Actors:	Managers

Precondition:	Check if all the required details are given and check if all of them are in the required format.
Flow:	Main Flow: 1) Input the nodes. 2) Find the shortest path between them. 3) Print it in the map.
	Alternate Flows: Proper error message when fields are left empty or are not in the given format.
Post conditions:	This can be further used by the manager to compare with the path with least CFP and choose one.

Use Case Number:	UC-5
Use Case Name:	Path with least CFP
Overview:	Given input as source and destination find the path with the least CFP between these two nodes.
Actors:	Managers

Precondition:	Check if all the required details are given and check if all of them are in the required format.
Flow:	Main Flow: 1) Take input of 2 nodes. 2) Use a query to find out the path with least CFP between them. 3) Display the path using a map.
	Alternate Flows: Proper error message when fields are left empty or are not in the given format.
Post conditions:	This can be used to reduce the expenditure and also to analyse if the shortest path and the path with least carbon footprint are the same.

Use Case Number:	UC-6
Use Case Name:	Route-wise CFP
	Based on input provided by the company's manager CFP is printed.

Overview:	
Actors:	Managers
Precondition:	Input the number of trucks, start and end nodes.
Flow:	Main Flow: 1) Input the data. 2) Compute the carbon footprint using the input provided. Alternate flows: Error when fields are not filled or not in the correct format.
Post conditions:	Can be further used by the companies for managing their carbon credits.

Use Case Number:	UC-7
Use Case Name:	Choose nodes from a Map.

Overview:	Selection of nodes from map.
Actors:	Managers
Precondition:	None
Flow:	Main Flow: 1)Display the map. 2)Allow the selection of nodes from the map. 3)Store them in the DB.
	Alternate Flow: Clear button if he wants to choose again.
Post conditions:	This can is further used to find the shortest path and the path with least CFP between the nodes

Use Case Number:	UC-8

Use Case Name:	Log or history
Overview:	Storing all the paths chosen and printing the log along with the net CFP of the company.
Actors:	Managers
Precondition:	Users should choose a path for it to be uploaded in the log.
Flow:	Main Flow: 1) Take the input as source and destination, number of trucks. 2) Show the paths. 3) Input the chosen path. 4) Push into the DB. 5) Retrieve and display.\ 6) Net CFP is also displayed.
	Alternate Flow: If no paths are chosen, the log is empty.
Post conditions:	Users can analyse this data and make future changes to save credits.

Use Case Number:	UC-9
Use Case Name:	Net CFP of the company
Overview:	To calculate the net carbon footprint of the company in order to take the necessary measures.
Actors:	Manager
Precondition:	Logged in as manager.
Flow:	Main Flow: 1) Retrieve the data from the DB and calculate. 2) Display the sum.
	Alternate Flow: If there is no data print 0

Post conditions:	Users can analyse this data and make future changes to save credits.

Use Case Number:	UC-10
Use Case Name:	Pre-emptive CFP Calculation
Overview:	Based on input provided by the company's manager CFP is printed.
Actors:	Managers
Precondition:	Input the number of trucks, start and end nodes.
Flow:	Main Flow: 1) Input the data. 2) Compute the carbon footprint using the input provided.

	Alternate flows: Error when fields are not filled or not in the correct format.
Post conditions:	Can be further used by the companies for managing their carbon credits.

Use Case Number:	UC-11
Use Case Name:	Identifying hotspots with high carbon emission
Overview:	Based on the input data display the regions/routes with the highest carbon footprint parameter wise. (For eg, year wise)
Actors:	Executive officer
Precondition:	Check if the necessary details are present in the Database (shape file).

Flow:	Main Flow: 2) Retrieve the files inputted from the DB. 2) Generate a map by highlighting the routes and areas with high carbon footprint
	Alternate Flow: If the necessary files don't exist in database pop an error message to the user
Post conditions:	Necessary measures can be taken to decrease the carbon footprint in the hotspot areas.