



PARIS CITY MAPPER

Une application qui trouve vos itinéraires et simplifie vos trajets quotidiens
dans Paris et ses environs

Rapport de Projet Base de données
Licence 3 informatique
Institut Galilée

Abdou Lahat SYLLA & Mursal ZUHAIR

SOMMAIRE

INTRODUCTION	3
FONCTIONNALITÉS	4
HOPS	4
TYPE	5
MAPS.....	7
LES BOUTONS.....	9
ENTITY-RELATIONSHIP DIAGRAM	9
<i>ERD EXPLICATION</i>	12
LISTE DES TABLES.....	13
LISTE DES DEPENDENCES	14
SQL QUERIES	14
DIFFICULTÉS RENCONTRÉS.....	17
RÉPARTITION DES TACHES	21

INTRODUCTION

Dans le cadre de notre cours de base de données en Licence 3 Informatique, nous sommes amenés à créer une application city mapper pour une ville de notre choix. Pour notre part, nous avons choisi le grand Paris, le moyen de transport vers et depuis la ville de Paris et ses villes environnantes de la région Ile-de-France. L'application trouve plusieurs itinéraires pour un utilisateur en fonction de ses lieux de départ et d'arrivée. L'application utilise PyQt5, une bibliothèque qui permet d'utiliser le framework Qt GUI de Python et intègre une carte Folium (cf. figure 1). Il permet à l'usager de choisir le moyen de transport de son choix ou une combinaison de moyens de transport et lui propose des itinéraires au départ de la gare la plus proche de l'usager.

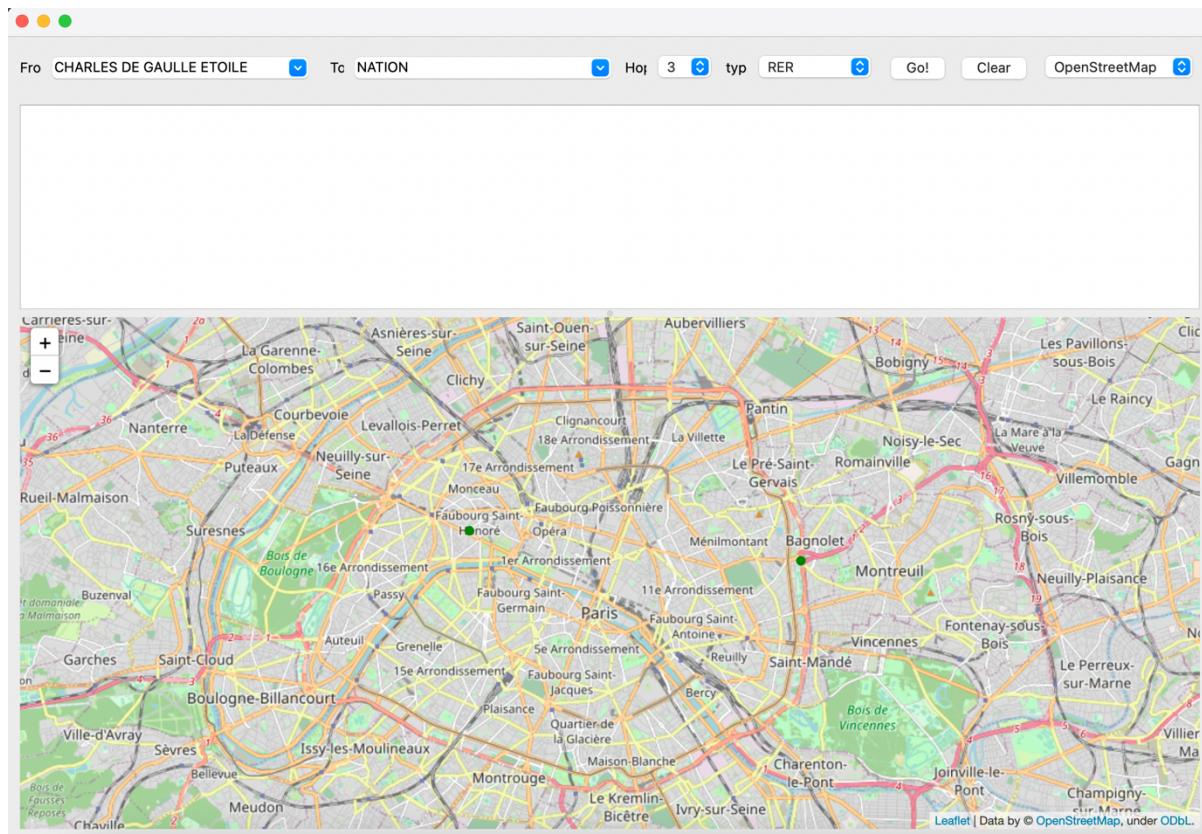


Figure 1

FONCTIONNALITÉS

Dans cette application, un utilisateur peut cliquer sur un emplacement sur la carte qui sera considéré comme le point de départ et un autre clic sur un emplacement différent qui serait sa destination, vers laquelle l'utilisateur se dirige. Les utilisateurs peuvent également saisir les gares de départ et d'arrivée en haut à droite de leur écran dans les rubriques « To » et « From. » En cliquant sur « Go », la trajectoire de l'utilisateur est tracée sur la carte et plusieurs choix de trajets lui sont proposés.

HOPS

L'usager peut choisir le nombre de fois maximum qu'il souhaite changer de moyen de transport. Par exemple, pour aller de Gare d'Austerlitz à Jean Vilar, 2 « hops » nous propose BUS 24 de Gare d'Austerlitz à Ecole Vétérinaire de Maison Alfort et BUS 103 de ce dernier à Jean Vilar (cf. Figure 2) et 3 « hops » nous propose le trajet précédent, BUS 24-> BUS 126 -> BUS 103 en plus de plusieurs autres trajets pour arriver à Jean Vilar (cf. Figure 3). Hops est un menu déroulant au-dessus de la carte où les utilisateurs ont la possibilité de choisir jusqu'à 5 changements de transport.

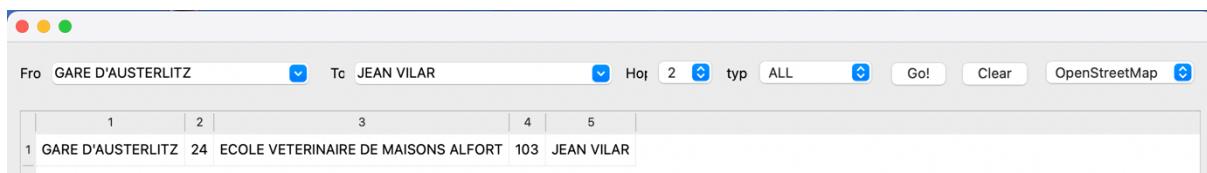


Figure 2

The screenshot shows a software interface with a toolbar at the top. The 'From' field contains 'GARE D'AUSTERLITZ' and the 'To' field contains 'JEAN VILAR'. Below these fields are dropdown menus for 'Hops' (set to 3), 'typ' (set to Bus), and 'Go!', 'Clear', and 'OpenStreetMap' buttons. A results table below the controls shows multiple rows for 3-hop routes, including: '1 GARE D'AUSTERLITZ 24 ECOLE VETERINAIRE DE MAISONS ALFORT 103 JEAN VILAR', '2 GARE D'AUSTERLITZ 24 CHARENTON ECOLES 325 GENERAL DE GAULLE 103 JEAN VILAR', '3 GARE D'AUSTERLITZ 24 CHARENTON ECOLES 325 MAISONS ALFORT ECOLE VETERINAIRE 103 JEAN VILAR', '4 GARE D'AUSTERLITZ 24 CONCORDE 107 ECOLE VETERINAIRE DE MAISONS ALFORT 103 JEAN VILAR', '5 GARE D'AUSTERLITZ 24 CONCORDE 299 GENERAL DE GAULLE 103 JEAN VILAR', and '6 GARE D'AUSTERLITZ 24 CUVIER 318 GENERAL DE GAULLE 103 JEAN VILAR'.

	1	2	3	4	5	6	7
1	GARE D'AUSTERLITZ	24	ECOLE VETERINAIRE DE MAISONS ALFORT	103	JEAN VILAR		
2	GARE D'AUSTERLITZ	24	CHARENTON ECOLES	325	GENERAL DE GAULLE	103	JEAN VILAR
3	GARE D'AUSTERLITZ	24	CHARENTON ECOLES	325	MAISONS ALFORT ECOLE VETERINAIRE	103	JEAN VILAR
4	GARE D'AUSTERLITZ	24	CONCORDE	107	ECOLE VETERINAIRE DE MAISONS ALFORT	103	JEAN VILAR
5	GARE D'AUSTERLITZ	24	CONCORDE	299	GENERAL DE GAULLE	103	JEAN VILAR
6	GARE D'AUSTERLITZ	24	CUVIER	318	GENERAL DE GAULLE	103	JEAN VILAR

Figure 3

TYPE

Les utilisateurs peuvent également choisir parmi plusieurs types de transport dont Bus, Métro, Tram, RER, RER et Tram, RER et Bus, Marche ou tout moyens de transport combinés.

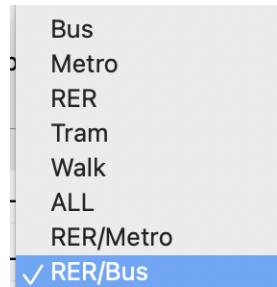


Figure 4

A titre d'exemple un usager choisit de se rendre des Maraîchers à la Basilique de Saint Denis en choisissant RER/BUS, les itinéraires suivants lui sont proposés :

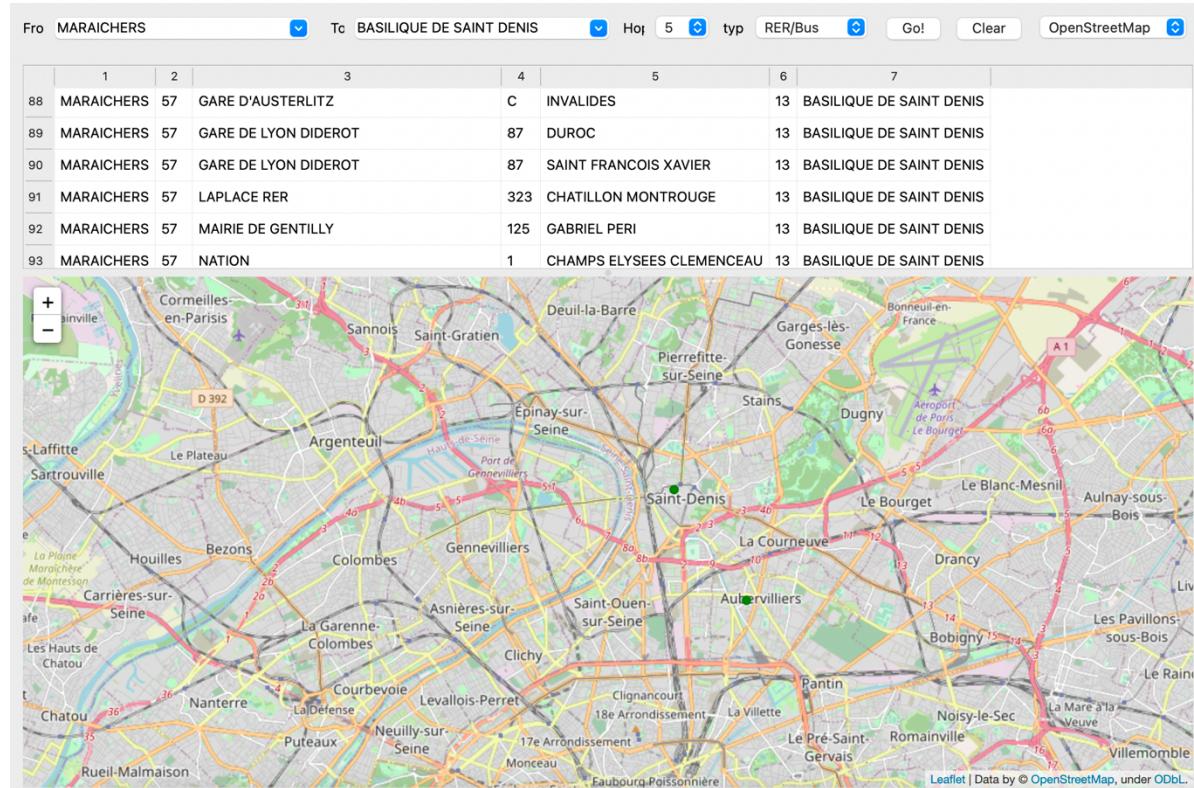
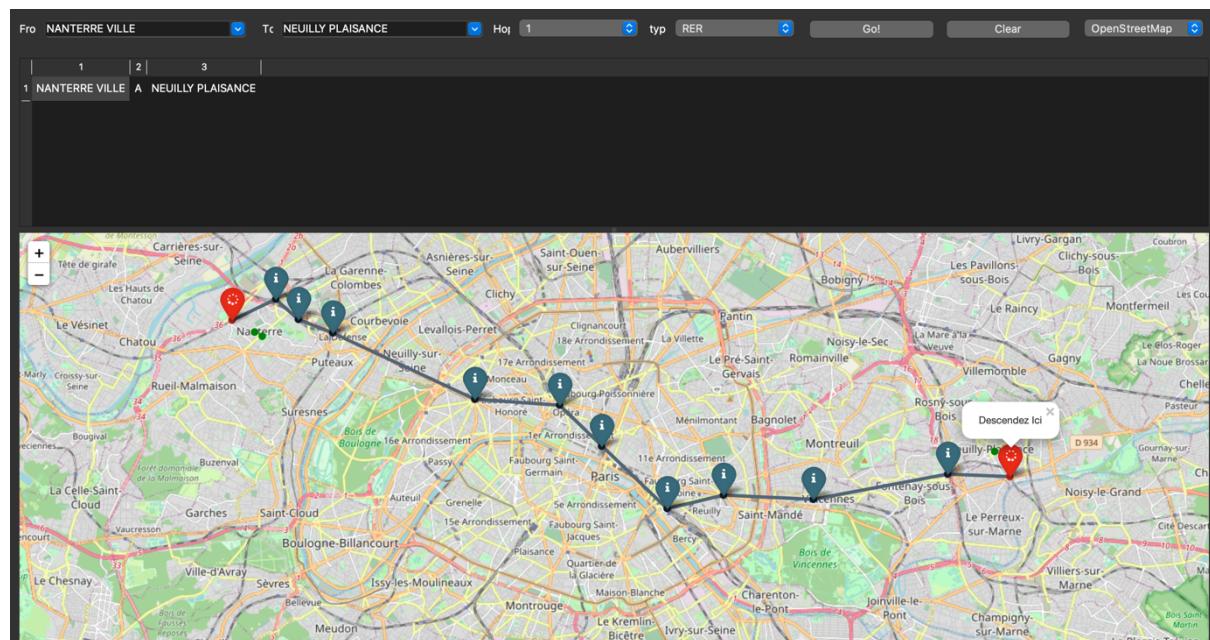
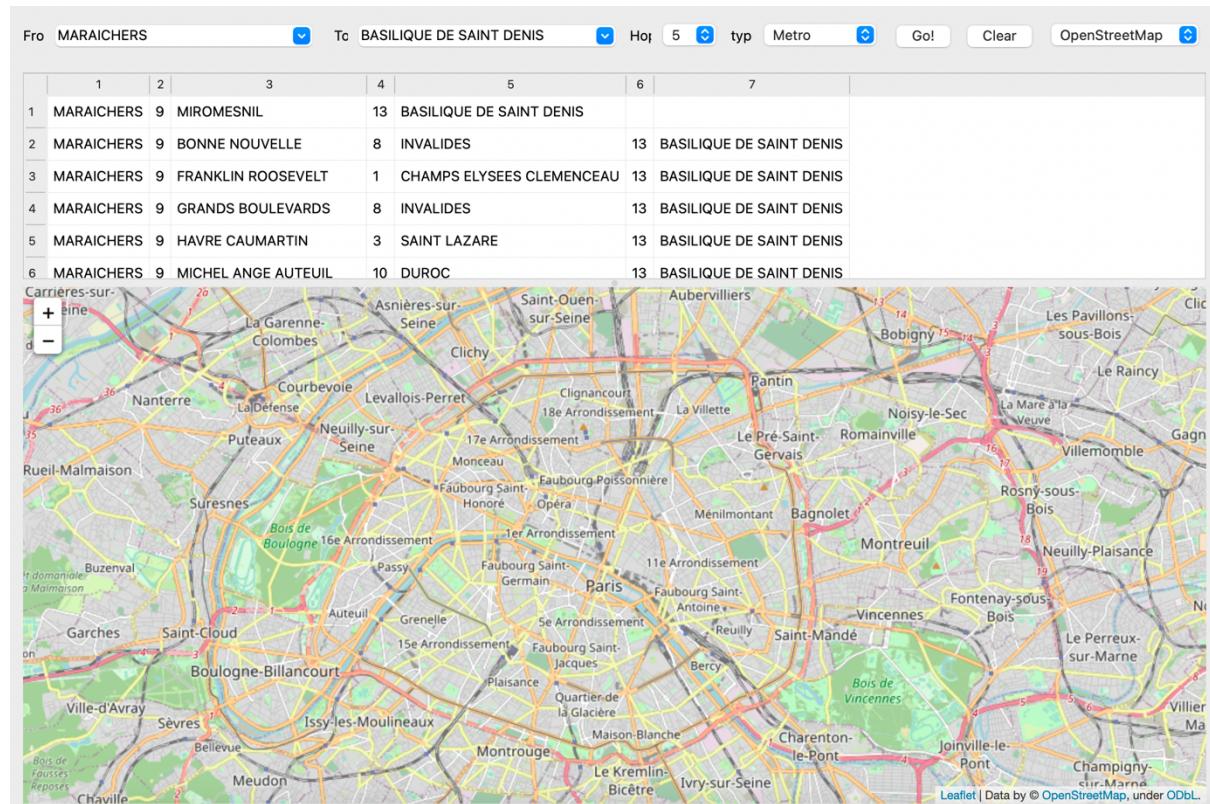


Figure 5

Pour parcourir le même trajet avec seulement Metro, les itinéraires suivants sont proposés :



L'utilisateur peut choisir de marcher d'un point à un autre. Dans ce cas, l'utilisateur peut cliquer sur deux points différents sur la carte, sélectionner le type Walk puis cliquer sur Go! qui tracera le meilleur chemin piéton sur la carte.

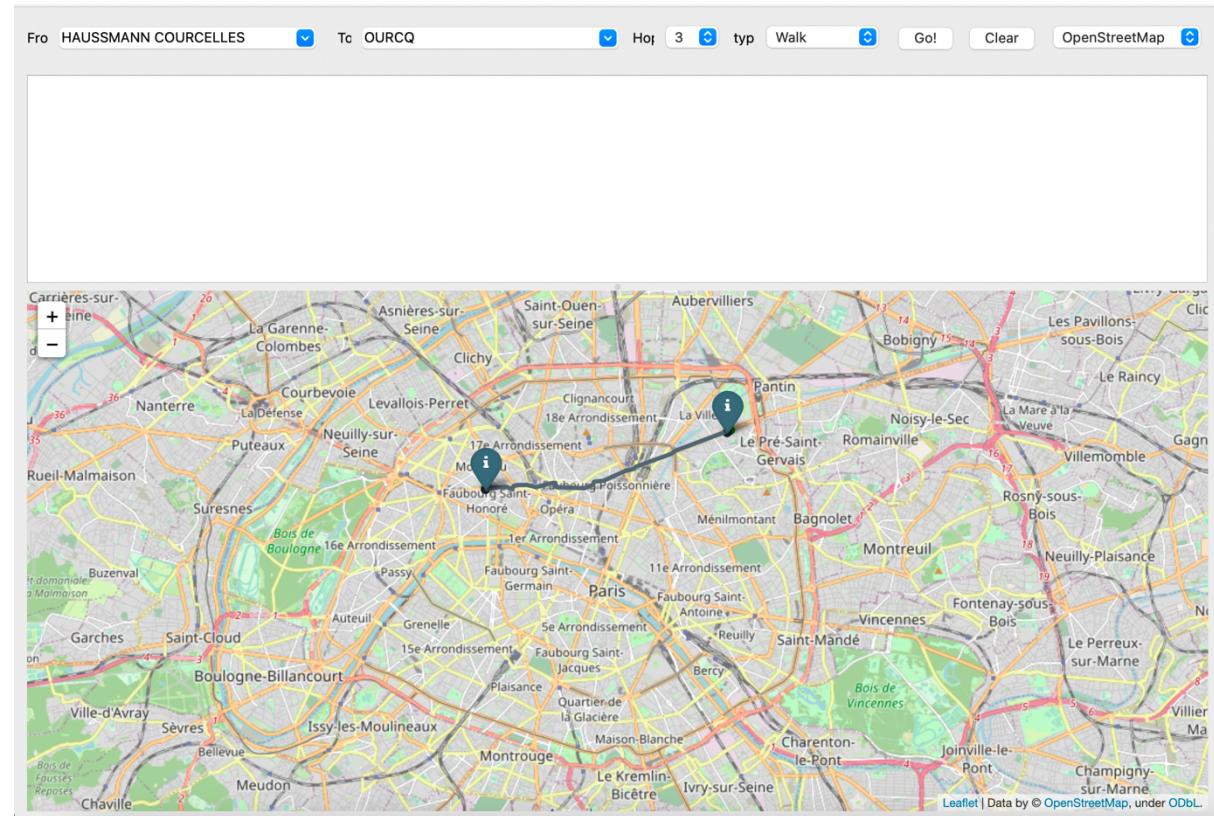


Figure 6

MAPS

Les utilisateurs ont la possibilité de choisir parmi 4 cartes : Cartodbposition, Stamentoner, Stamen Terrain et OpenStreetMap à l'aide du menu déroulant situé en haut à droite de l'écran.

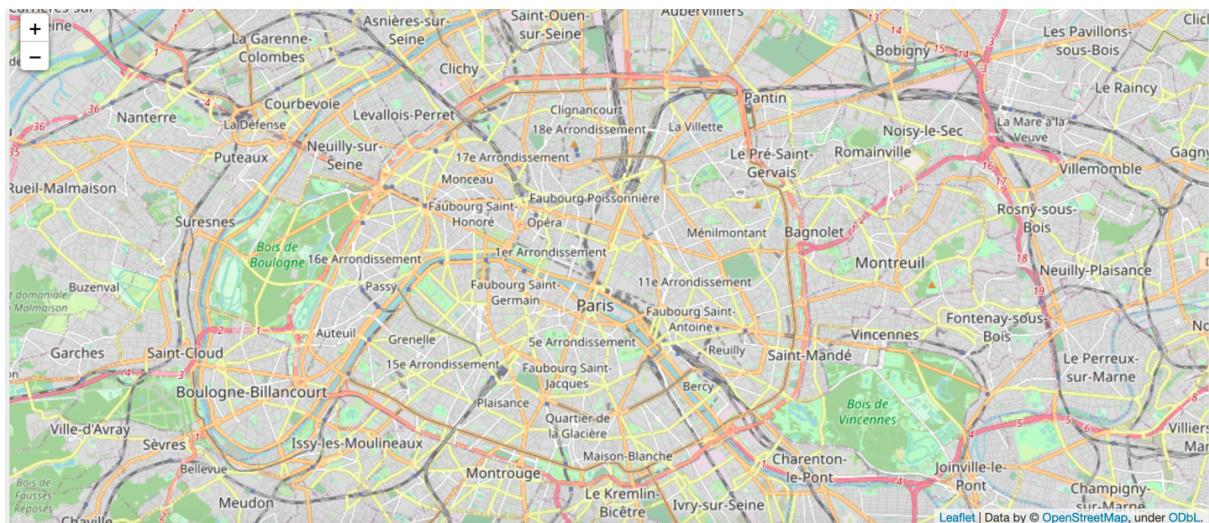


Figure 7 OpenStreetMap

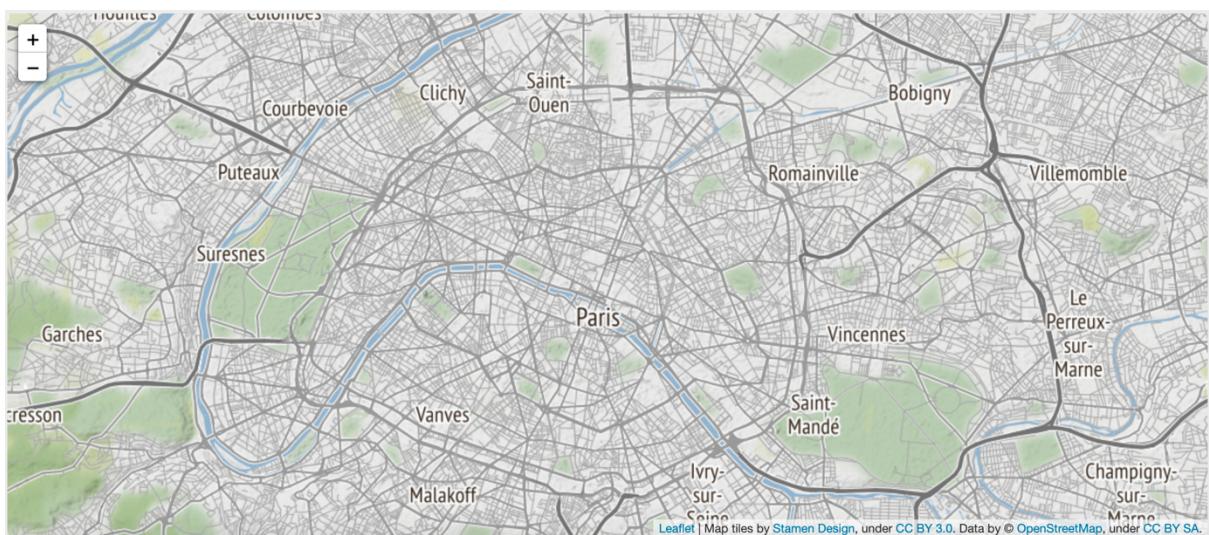


Figure 8 Stamen Terrain



Figure 9 Stamen Toner

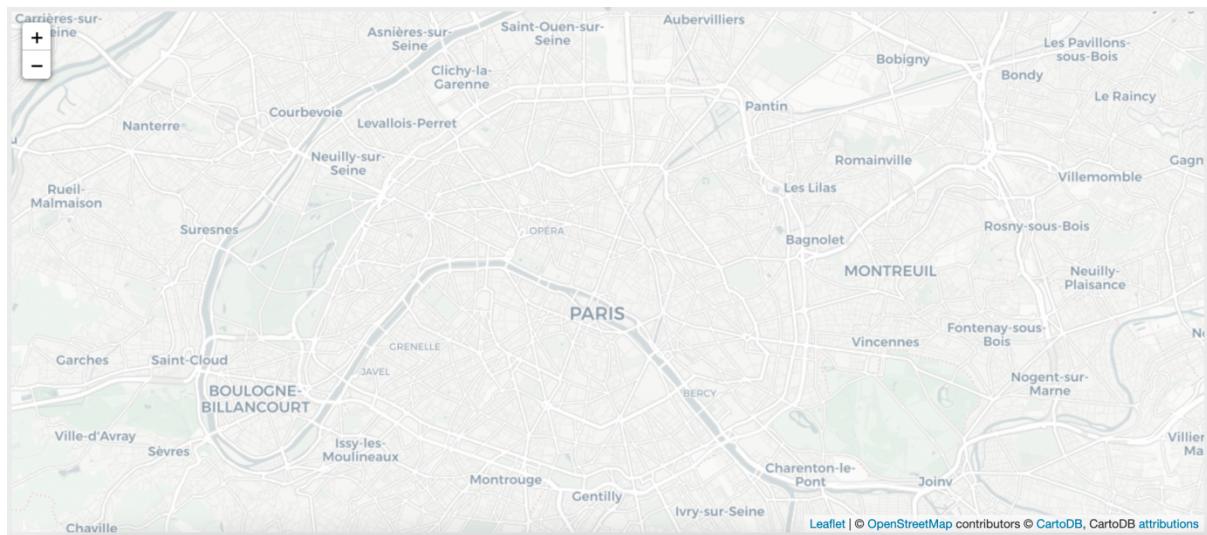


Figure 10 CartodbPositron

LES BOUTONS

Le window principale de l'application se compose également de deux boutons Go et Clear qui respectivement affiche le résultat et réinitialise la window.

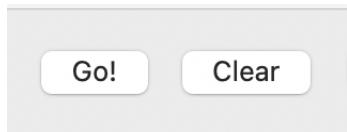
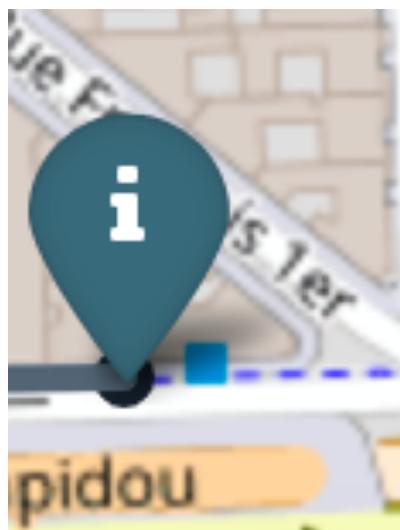


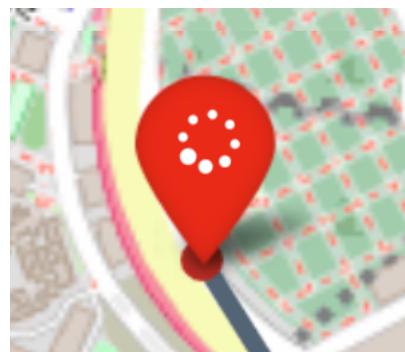
Figure 11

Marquer

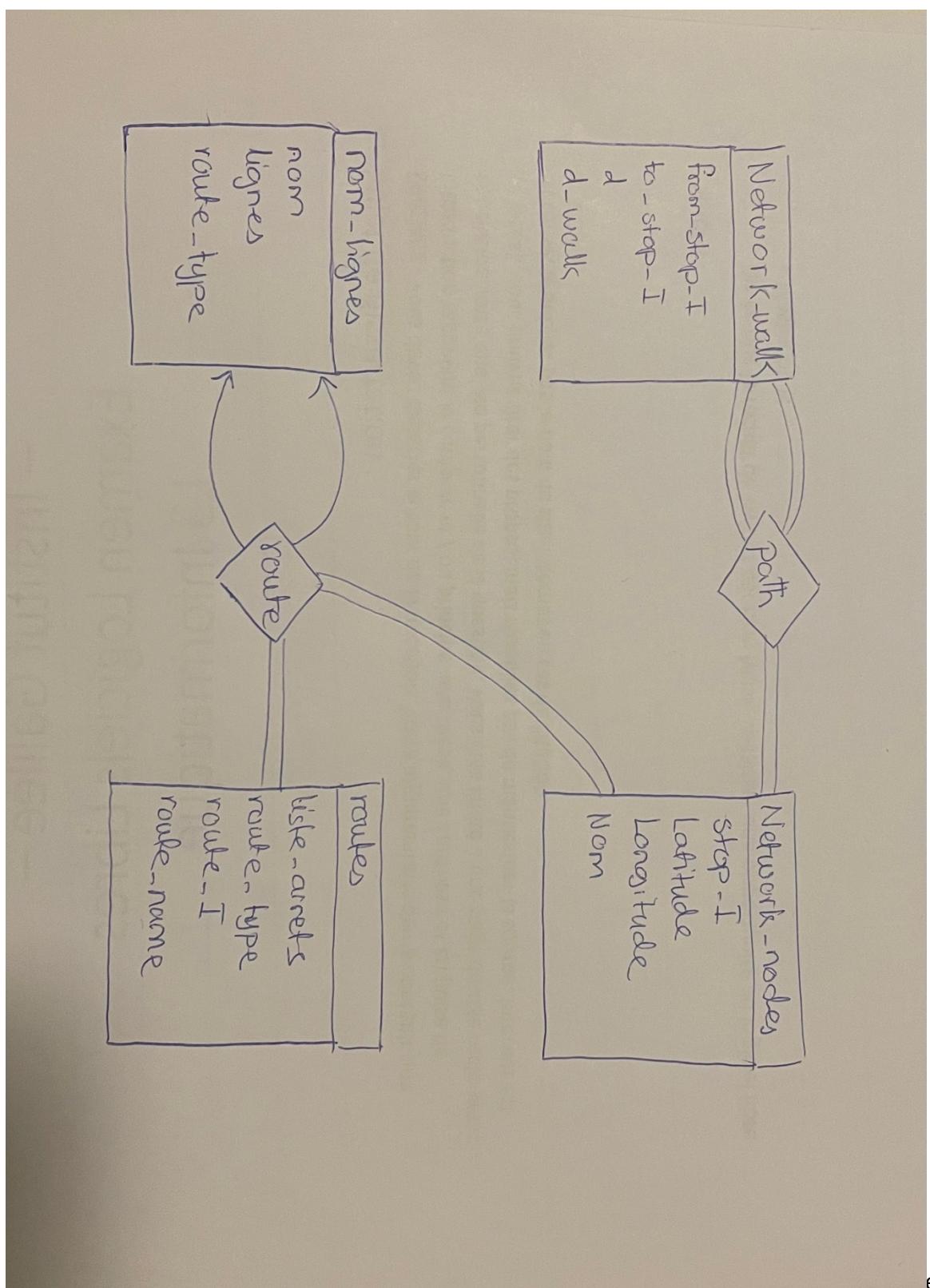
Pour montrer que c'est une station ou on ne s'arrête pas :



Pour montrer une station ou on s'arrête :



ENTITY-RELATIONSHIP DIAGRAM



es.

ERD EXPLICATION

Nous avons utilisé les fichiers suivants pour créer nos tables :

- network_node.csv
- network_walk.csv
- stops.geojson
- sections.geojson

Pour obtenir la table routes on a utilisé le fichier routes et network_node.csv fourni sur le site des données. Le fichier sections.geojson a été utilisé pour entrer les stations manquantes dans routes dans la table.

On a remplacé les coordonnées par les stop_i correspondant à chaque coordonnées stockées dans routes dans l'attribut listes_arrets avec le nom, le route type (0 pour le tramway, 1 pour le métro, 2 pour le train et 3 pour le bus).

Pour obtenir la table stops on a parser le fichier network_node.csv avec le même nom des attributs.

Pour la table stations, on a combiné les fichiers section.geojson, network_node.csv.

Les noms de la table stops et stations sont en majuscule sans point en fin de chaîne et sans accents pour éviter certaines répétitions dans l'affichage.

Pour la table walk on a parse le fichier network_walk.csv.

On a omis les relations dans le diagramme qui sont jugés inutiles dans la partie application.

LISTE DES TABLES

Nous avons les tables suivantes dans notre base de données : network_node, network_walk, routes, lignes, noms_lignes.

Liste des attributs et leurs types pour chacune des tables dans notre base de données ci-dessous :

Network_node				
Attributes	Stop_I	Latitude	Longitude	NOM
Attribute type	NUMERIC(10,2)	TEXT	TEXT	VARCHAR

Network_walk				
Attributes	From_stop_I	To_stop_I	d	D_walk
Attribute type	NUMERIC(10,0)	NUMERIC(10,0)	NUMERIC(10,0)	NUMERIC(10,0)

Routes				
Attributes	Listes_arrets	Route_type	Route_I	Route_name
Attribute type	TEXT	NUMERIC(1)	NUMERIC(4)	VARCHAR(32)

Nom_lignes			
Attributes	nom	Lignes	route_type
Attribute type	TEXT	TEXT	NUMERIC(1)

LISTE DES DEPENDENCES

Network_node : route_I -> latitude, longitude, nom
Network_walk : from_stop_I, to_stop_I -> d, d_walk
Routes : route_I -> listes_arrets, route_name, route_type
Lignes : nom -> stations, route_type
Nom_lignes : nom -> lignes, route_type

SQL QUERIES

- Requête utilisée pour remplir le graphe pour rechercher les chemins pour la marche :

```
SELECT from_stop_I,to_stop_I, d_walk from walk
```

- Requête utilisée le chemin direct entre 2 stations de métro :

```
SELECT DISTINCT A.nom,A.ligne,B.nom FROM stations as A, stations as B WHERE B.nom =  
$$\{_tostation}$$ and A.nom = $$\{fromstation}$$ AND A.ligne = B.ligne AND A.route_type =  
B.route_type AND A.route_type = 1 AND B.route_type = 1
```

- Requête utilisée le chemin direct entre 2 stations de tramway

```
SELECT DISTINCT A.nom,A.ligne,B.nom FROM stations as A, stations as B WHERE B.nom =  
$$\{_tostation}$$ and A.nom = $$\{fromstation}$$ AND A.ligne = B.ligne AND A.route_type =  
B.route_type AND A.route_type = 0 AND B.route_type = 0
```

- le chemin direct entre 2 stations de train/RER :

```
SELECT DISTINCT A.nom,A.ligne,B.nom FROM stations as A, stations as B WHERE B.nom =  
$$\{_tostation}$$ and A.nom = $$\{fromstation}$$ AND A.ligne = B.ligne AND A.route_type =  
B.route_type AND A.route_type = 2 AND B.route_type = 2
```

- Requête utilisée pour le chemin direct entre 2 stations de bus :

```
SELECT DISTINCT A.nom,A.ligne,B.nom FROM stations as A, stations as B WHERE B.nom =  
$$\{_tostation}$$ AND A.nom = $$\{_fromstation}$$ AND A.ligne = B.ligne AND A.route_type =  
B.route_type AND A.route_type = 3 AND B.route_type = 3
```

- Requête utilisée pour le chemin par 2 métros entre 2 stations :

```
SELECT distinct A.nom, A.ligne, B.nom, C.ligne, D.nom FROM stations as A, stations as B,  
stations as C, stations as D WHERE A.nom = $$\{_fromstation}$$ AND D.nom =  
$$\{_tostation}$$ AND A.ligne = B.ligne AND B.nom = C.nom AND C.ligne = D.ligne AND  
A.ligne <> C.ligne AND A.nom <> B.nom AND B.nom <> D.nom AND A.route_type = 1 AND  
B.route_type = 1 AND C.route_type = 1 AND D.route_type = 1
```

- Requête utilisée pour le chemin par 2 train/RER entre 2 stations :

```
SELECT distinct A.nom, A.ligne, B.nom, C.ligne, D.nom FROM stations as A, stations as B,  
stations as C, stations as D WHERE A.nom = $$\{_fromstation}$$ AND D.nom =  
$$\{_tostation}$$ AND A.ligne = B.ligne AND B.nom = C.nom AND C.ligne = D.ligne AND  
A.ligne <> C.ligne AND A.nom <> B.nom AND B.nom <> D.nom AND A.route_type = 2 AND  
B.route_type = 2 AND C.route_type = 2 AND D.route_type = 2
```

- Requête utilisée pour le chemin par 2 bus entre 2 stations :

```
SELECT distinct A.nom, A.ligne, B.nom, C.ligne, D.nom FROM stations as A, stations as B,  
stations as C, stations as D WHERE A.nom = $$\{_fromstation}$$ AND D.nom =  
$$\{_tostation}$$ AND A.ligne = B.ligne AND B.nom = C.nom AND C.ligne = D.ligne AND  
A.ligne <> C.ligne AND A.nom <> B.nom AND B.nom <> D.nom AND A.route_type = 3 AND  
B.route_type = 3 AND C.route_type = 3 AND D.route_type = 3
```

- Requête pour le chemin par 3 train/RER entre 2 stations :

```
SELECT distinct A.nom, A.ligne, B2.nom, B2.ligne, C2.nom, C2.ligne, D.nom FROM stations as  
A, stations as B1, stations as B2, stations as C1, stations as C2, stations as D WHERE A.nom =  
$$\{_fromstation}$$ AND A.ligne = B1.ligne AND B1.nom = B2.nom AND B2.ligne = C1.ligne  
AND C1.nom = C2.nom AND C2.ligne = D.ligne AND D.nom = $$\{_tostation}$$ AND A.ligne  
<> B2.ligne AND B2.ligne <> C2.ligne AND A.ligne <> C2.ligne AND A.nom <> B1.nom AND  
B2.nom <> C1.nom AND C2.nom <> D.nom AND A.route_type = 2 AND B1.route_type = 2
```

AND B2.route_type = 2 AND C1.route_type = 2 AND C2.route_type = 2 AND D.route_type = 2

- Requête pour le chemin par 3 métros entre 2 stations :

```
SELECT distinct A.nom, A.ligne, B2.nom, B2.ligne, C2.nom, C2.ligne, D.nom FROM stations as A, stations as B1, stations as B2, stations as C1, stations as C2, stations as D WHERE A.nom = $$\_{fromstation}$$ AND A.ligne = B1.ligne AND B1.nom = B2.nom AND B2.ligne = C1.ligne AND C1.nom = C2.nom AND C2.ligne = D.ligne AND D.nom = $$\_{tostation}$$ AND A.ligne <> B2.ligne AND B2.ligne <> C2.ligne AND A.ligne <> C2.ligne AND A.nom <> B1.nom AND B2.nom <> C1.nom AND C2.nom <> D.nom AND A.route_type = 1 AND B1.route_type = 1 AND B2.route_type = 1 AND C1.route_type = 1 AND C2.route_type = 1 AND D.route_type = 1
```

- Requête pour le chemin par 3 bus entre 2 stations :

```
SELECT distinct A.nom, A.ligne, B2.nom, B2.ligne, C2.nom, C2.ligne, D.nom FROM stations as A, stations as B1, stations as B2, stations as C1, stations as C2, stations as D WHERE A.nom = $$\_{fromstation}$$ AND A.ligne = B1.ligne AND B1.nom = B2.nom AND B2.ligne = C1.ligne AND C1.nom = C2.nom AND C2.ligne = D.ligne AND D.nom = $$\_{tostation}$$ AND A.ligne <> B2.ligne AND B2.ligne <> C2.ligne AND A.ligne <> C2.ligne AND A.nom <> B1.nom AND B2.nom <> C1.nom AND C2.nom <> D.nom AND A.route_type = 3 AND B1.route_type = 3 AND B2.route_type = 3 AND C1.route_type = 3 AND C2.route_type = 3 AND D.route_type = 3
```

- Requête pour le chemin par 3 tramways entre 2 stations :

```
SELECT distinct A.nom, A.ligne, B2.nom, B2.ligne, C2.nom, C2.ligne, D.nom FROM stations as A, stations as B1, stations as B2, stations as C1, stations as C2, stations as D WHERE A.nom = $$\_{fromstation}$$ AND A.ligne = B1.ligne AND B1.nom = B2.nom AND B2.ligne = C1.ligne AND C1.nom = C2.nom AND C2.ligne = D.ligne AND D.nom = $$\_{tostation}$$ AND A.ligne <> B2.ligne AND B2.ligne <> C2.ligne AND A.ligne <> C2.ligne AND A.nom <> B1.nom AND B2.nom <> C1.nom AND C2.nom <> D.nom AND A.route_type = 1 AND B1.route_type = 0 AND B2.route_type = 0 AND C1.route_type = 0 AND C2.route_type = 0 AND D.route_type = 0
```

DIFFICULTÉS RENCONTRÉS

La première difficulté que nous avons rencontrée a été de comprendre les données. Il était compliqué de savoir comment les utiliser au début et il y avait beaucoup de données et d'informations, il fallait faire une sélection pour savoir ce qui devait être conservé pour notre application et ce qui devait être supprimé.

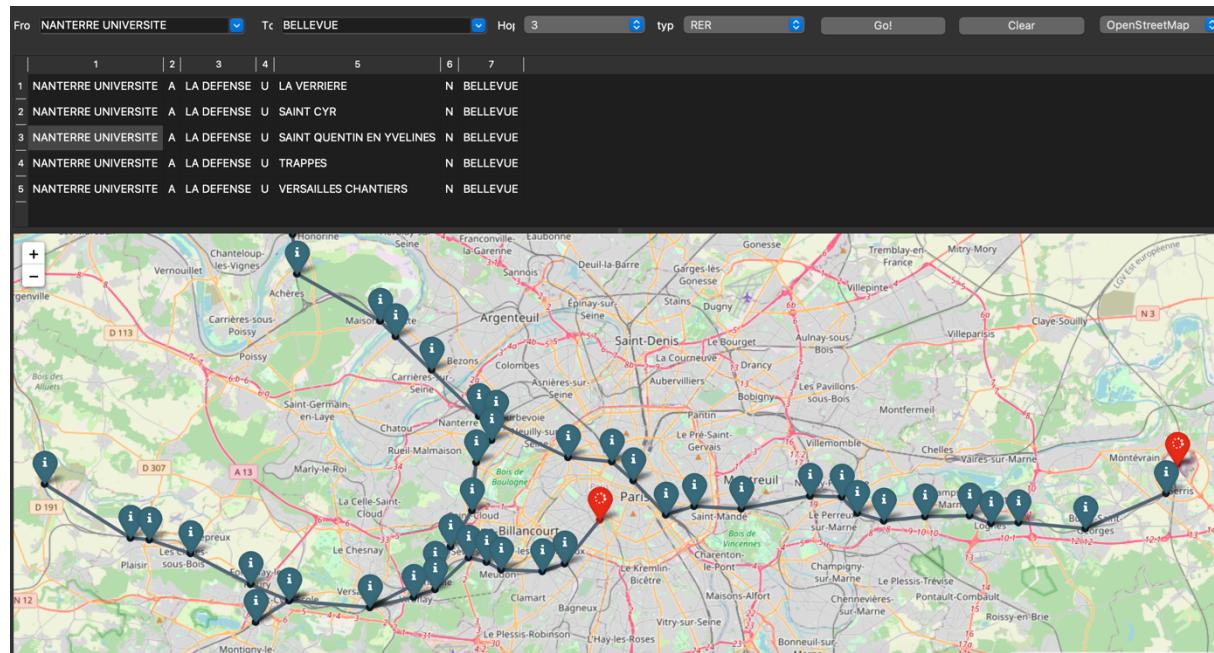
Ensuite, nous avons eu un problème avec l'application et l'affichage de comment ajouter une nouvelle boîte et les données à l'intérieur, car nous pensions qu'il fallait apporter de grosses modifications, alors que c'était très simple à faire.

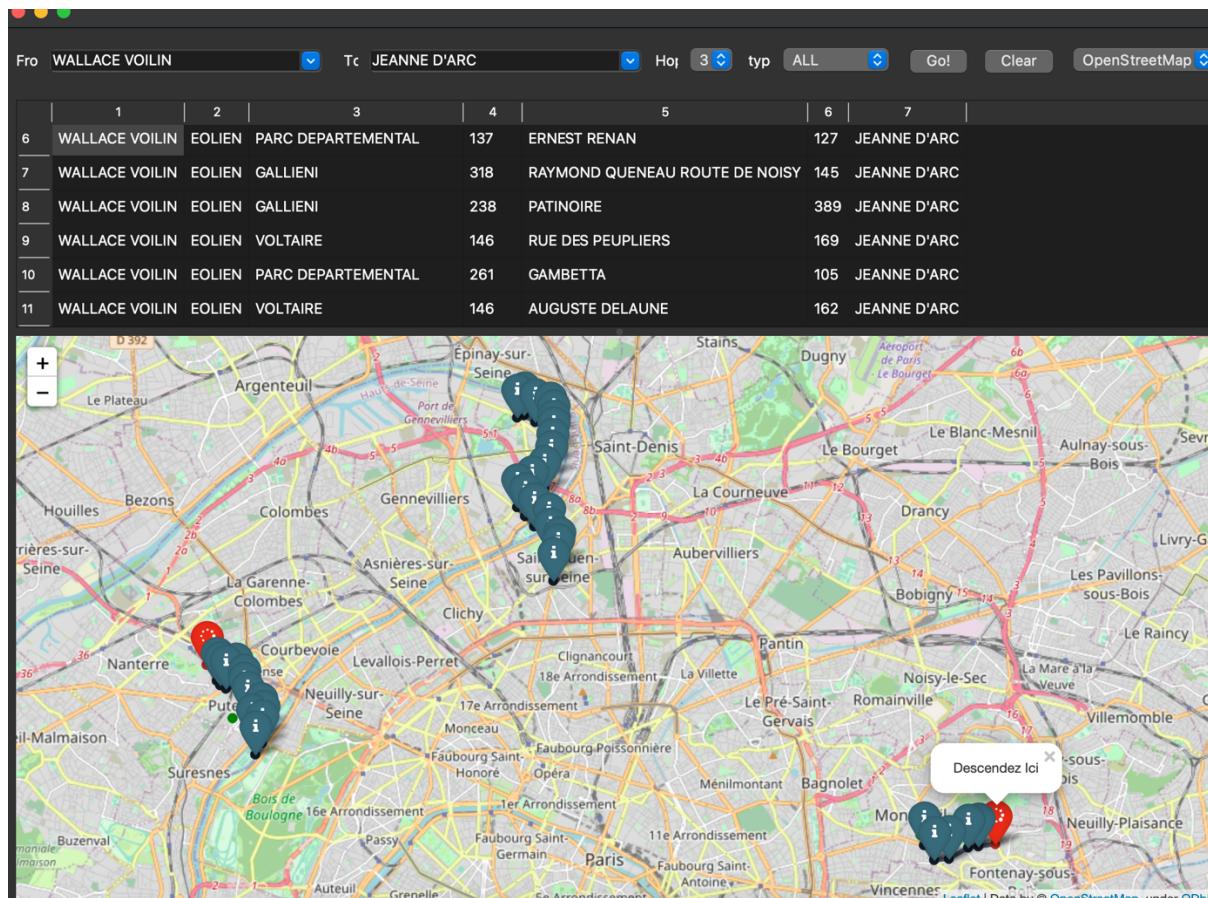
Ensuite, nous avons eu un problème avec le bouton "go", car nous devions prendre en compte les différents modes de transport (marche à pied, bus, RER, métro) car nous devions faire des choses différentes pour chaque cas. C'était vraiment difficile de le faire.

Ensuite, nous avons eu la partie "mouseclick", qui était selon nous la partie la plus compliquée et la plus longue, car nous devions également faire des requêtes et des choses différentes en fonction du mode de transport. Par exemple, si le type est "métro", il faut trouver la station de métro la plus proche, et de même pour les autres types de transport, il faut trouver la station correspondante spécifiée dans le type_box. Cela a été très compliqué à faire et cela nous a pris beaucoup de temps.

Ensuite, nous avons eu un problème avec "tableClick", car nous voulions faire quelque chose de plus propre, étant donné que nous avions les routes de chaque mode de transport, donc nous voulions tracer selon cette route et non simplement ajouter des segments entre les points. C'était très difficile d'extraire la route, car il y avait plusieurs routes possibles selon le "route_i", donc il fallait faire plusieurs boucles et requêtes pour trouver la bonne route contenant les deux stations et la prendre. Et pour chaque mode de transport, il était différent de savoir comment extraire les données de la base de données.

Le problème le plus difficile que nous avons rencontré était celui de la marche entre deux stations. Nous avons dû utiliser les graphes grâce à "networkX" et trouver les chemins les plus courts selon les distances, puis récupérer le "path" (chemin). Ensuite, nous avons pris ce chemin qui était composé de "stop_i" et avons effectué une autre requête pour récupérer les latitudes et longitudes.





Pour hops = 3, on n'a pas réussi à marquer les stations de départ et d'arrivée avec une couleur différente des stations intermédiaires en locurence rouge pour les stations départ et arrivée.

On n'a pas réussi à le faire pour à chaque fois qu'on prend une nouvelle ligne et qu'on descend, donc c'est un problème qu'on n'a pas pu résoudre.

Mais on va continuer de travailler dessus même si ça va pas changer notre note dans ce projet.

Fro WALLACE VOILIN Tc JEANNE D'ARC Hop 3 typ ALL Go! Clear OpenStreetMap

1	2	3	4	5	6	7
6 WALLACE VOILIN	EOLIEN	PARC DEPARTEMENTAL	137	ERNEST RENAN	127	JEANNE D'ARC
7 WALLACE VOILIN	EOLIEN	GALLIENI	318	RAYMOND QUENEAU ROUTE DE NOISY	145	JEANNE D'ARC
8 WALLACE VOILIN	EOLIEN	GALLIENI	238	PATINOIRE	389	JEANNE D'ARC
9 WALLACE VOILIN	EOLIEN	VOLTAIRE	146	RUE DES PEUPLIERS	169	JEANNE D'ARC
10 WALLACE VOILIN	EOLIEN	PARC DEPARTEMENTAL	261	GAMBETTA	105	JEANNE D'ARC
11 WALLACE VOILIN	EOLIEN	VOLTAIRE	146	AUGUSTE DELAUNEAU	162	JEANNE D'ARC

Map showing the locations of various points of interest in the Paris region, including the 16th arrondissement and surrounding suburbs like Argenteuil, Garges-lès-Gonesse, and Bobigny.

RÉPARTITION DES TACHES

Nous nous sommes répartis les tâches comme suit :

Abdou Lahat Sylla s'est occupé des requêtes, itinéraires et routes en implémentant les fonctions suivantes :

- mouseClicked
- add_customjs
- handleClick
- addSegment
- addRoute
- addCol
- addMarker
- setMap
- clearMap
- traceitineraire

Mursal s'est occupé de l'application et de l'intégration des données en implémentant les fonctions suivantes :

- connect_DB
- table_Click
- button_Go
- addPoint
- button_Clear
- javaScriptConsoleMessage

Retour d'expérience Abdou Lahat :

Dans ce projet, lorsque j'y repense maintenant, il y a certaines choses que j'aurais pu faire différemment. Tout d'abord, j'aurais pu me concentrer dessus dès le début et m'impliquer davantage afin de comprendre rapidement le sujet. Cela nous aurait permis de terminer plus rapidement. Nous aurions également pu mieux nous organiser et nous donner des

tâches plus indépendantes, en discutant de nos forces et faiblesses dès le début. Cela nous aurait permis de répartir le travail de manière plus équitable et d'être sûrs que le projet avancerait bien, même si nos horaires étaient difficiles à concilier. Ce projet m'a permis de mettre en pratique les connaissances que j'ai acquises au cours de ce semestre en base de données, mais aussi en algorithmique des graphes, grâce aux graphes de réseaux de transport. Cela m'a également donné l'occasion de travailler sur un projet de grande envergure, ce qui est très bénéfique car, dans le futur, je suis amené à travailler sur de gros projets et à avoir des tâches assignées et un délai non extensible

Retour d'expérience Mursal :

Si je pouvais recommencer le projet, je diviserais les fonctions et fixerais un délai pour que tous les membres du groupe viennent avec, si ce n'est un code fonctionnel, un algorithme ou juste une idée pour résoudre un problème. Ensuite, après chaque date limite, chacun serait obligé d'envoyer ses codes sur Git, que leur code fonctionne ou non. Cela empêcherait un autre membre du groupe de redéfinir une fonction dont il pourrait avoir besoin pour ses fonctions. Je poserais également plus de questions au professeur.

Ce projet était plus qu'un travail universitaire pour moi. Je n'étais toujours pas sûr de mon choix d'études avant ce dernier semestre. Je pensais que les informaticiens n'avaient pas de vie sociale. Je craignais de passer le reste de ma vie devant un écran. Grâce à ce projet, j'ai réalisé que l'informatique est bien plus que ce que je pensais. Nous ne sommes pas loin du monde lorsque nous sommes sur nos ordinateurs, nous sommes plus connectés que jamais. Une connexion qui ne connaît pas la distance. Ce projet de groupe m'a aussi ouvert les yeux sur l'aspect communicationnel de l'informatique, travailler en équipe, partager nos idées, et surtout, donner et recevoir des critiques sans les prendre personnellement.