Report: Public Transport Project

I – Functionalities list

Given a dataset of public transport network data (.**csv** files containing network nodes and temporal data, .**geojson** files containing routes and stops names, all taken from <https://www.nature.com/articles/sdata201889>, as well as additional **.json** files taken from ) that is kept within a folder on the user’s computer, as well as a *PostgreSQL* server, our *PyQT5* application can:

* Prompt the user the path to the **data** folder (with a *FileDialog* that will be used to select the folder), as well as the login credentials necessary to connect to the postgres database (with a custom *QDialog* that has four entries: **user**, **password** (this one does not show plain text), **database** and **host** (the IP address which the postgres server runs on).
* Save the variables from previous step inside a “**params.json** file so they can be reused next time the program is launched.
* Fill the postgres database with data obtained from the files within the **data** folder.
* Display a *QWindow* where the user can select a departure and a destination from two different Combo Boxes. Once both are selected, the user can press the “Go” button to launch the query that will give a path from the departure point to the arrival point. The user can also click the “Clear” button to clear the text within the Combo Boxes.
* Said window also features a *QTableWidget* that shows the detailed route the user must take to go from the departure point to the destination point on its first row, and the estimated time it would take on the second row. Clicking on a transport name (*RER E, BUS 165*…) within that table will make appear a message box, asking the user if they want to see said transport’s plan/map. If the user accepts, we will open the transport’s plan page within their default web browser. If there are multiple plans related to a transport name (often happens given how generic bus names are for example).
* Below that table, is a map (displayed with the *folium* module from a web page), which, when a route has been decided, shows all the stops of said route, linking each stop by a line, otherwise it shows all the stops that we know of on the map. The user can click a location on the map to select their departure and destination stops as well.

II – Entity-Relationship Diagram

J

III – Tables & Dépendances

IV – Encountered Difficulties & Contributions per member