Projet C++
Application RATP

Algorithme Dijkstra

Réalisation :

Koussaila KADI

Encadré par: Félix Rutard

04/06/2021

Sommaire

A. Introduction

B. Dijkstra
   A. Analyse du besoin
   B. Analyse fonctionnelle
   C. Architecture et diagramme de classes
   D. Implémentation

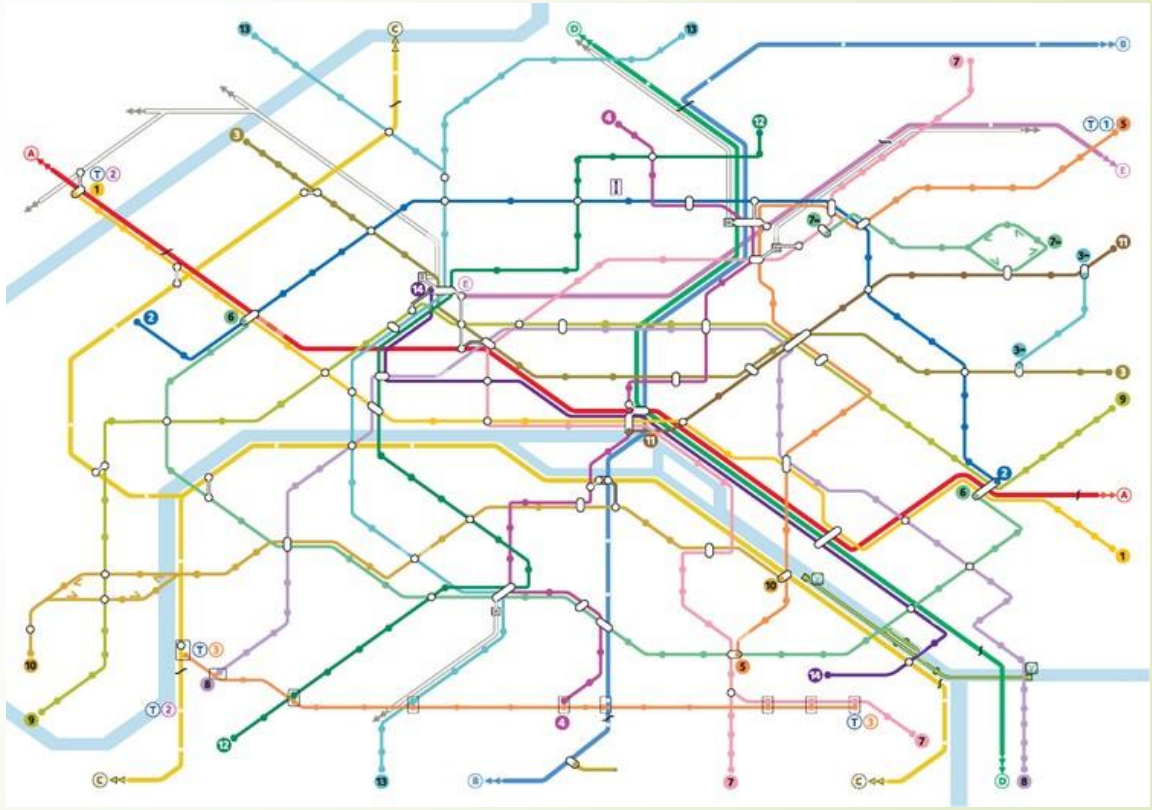
C. Conclusion

D. Indication et Read.me

A. Introduction

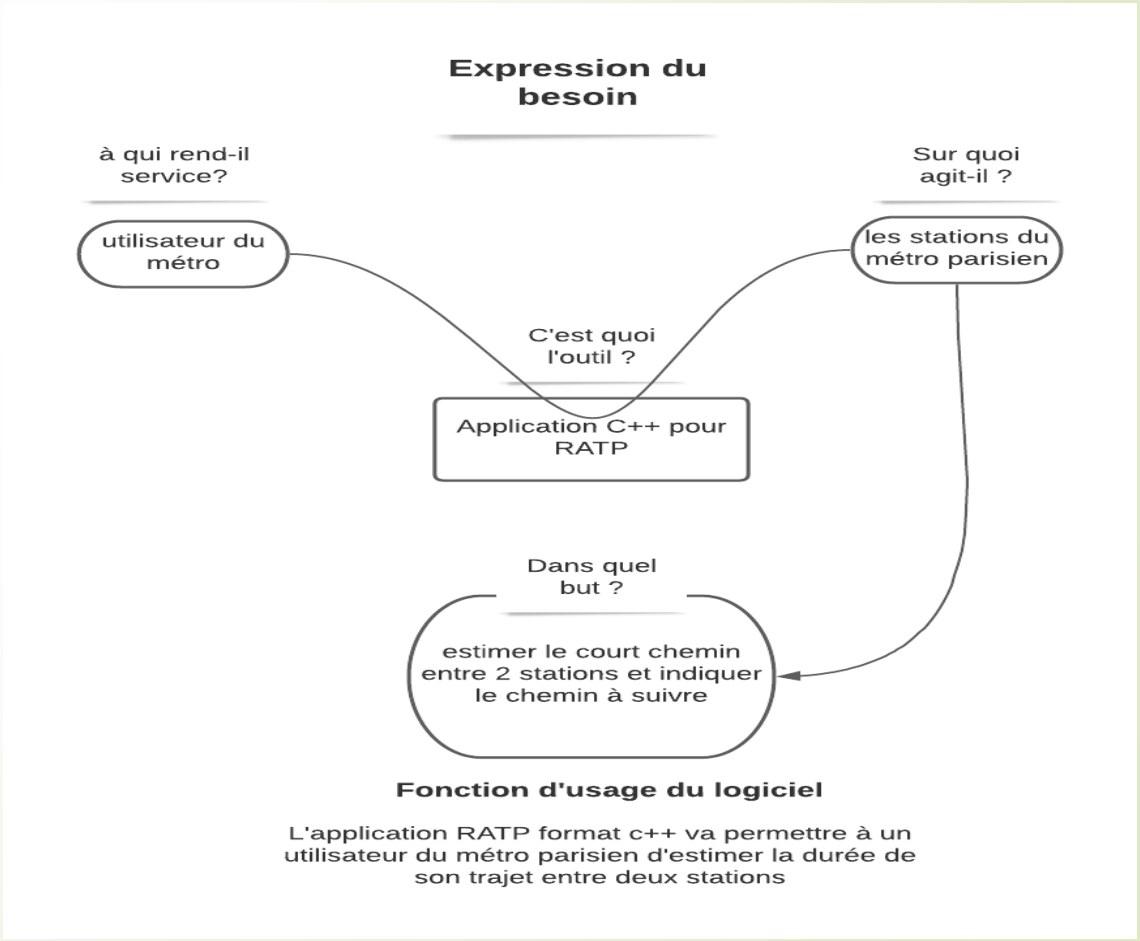
- L'algorithme de Dijkstra est un algorithme permettant de trouver les plus courts chemins entre les nœuds d'un graphe, qui peut représenter, par exemple, des réseaux routiers. Il a été conçu par l'informaticien Edsger W. Dijkstra en 1956 et publié trois ans plus tard.

- L'algorithme existe dans de nombreuses variantes. L'algorithme original de Dijkstra trouve le chemin le plus court entre deux nœuds donnés, mais une variante plus courante fixe un seul nœud comme nœud "source" et trouve les chemins les plus courts de la source à tous les autres nœuds du graphe, produisant un arbre des chemins les plus courts.



B. Dijkstra

- **A. Analyse du besoin:**



B. Dijkstra

B. Analyse fonctionnelle

1. Fonction principale:

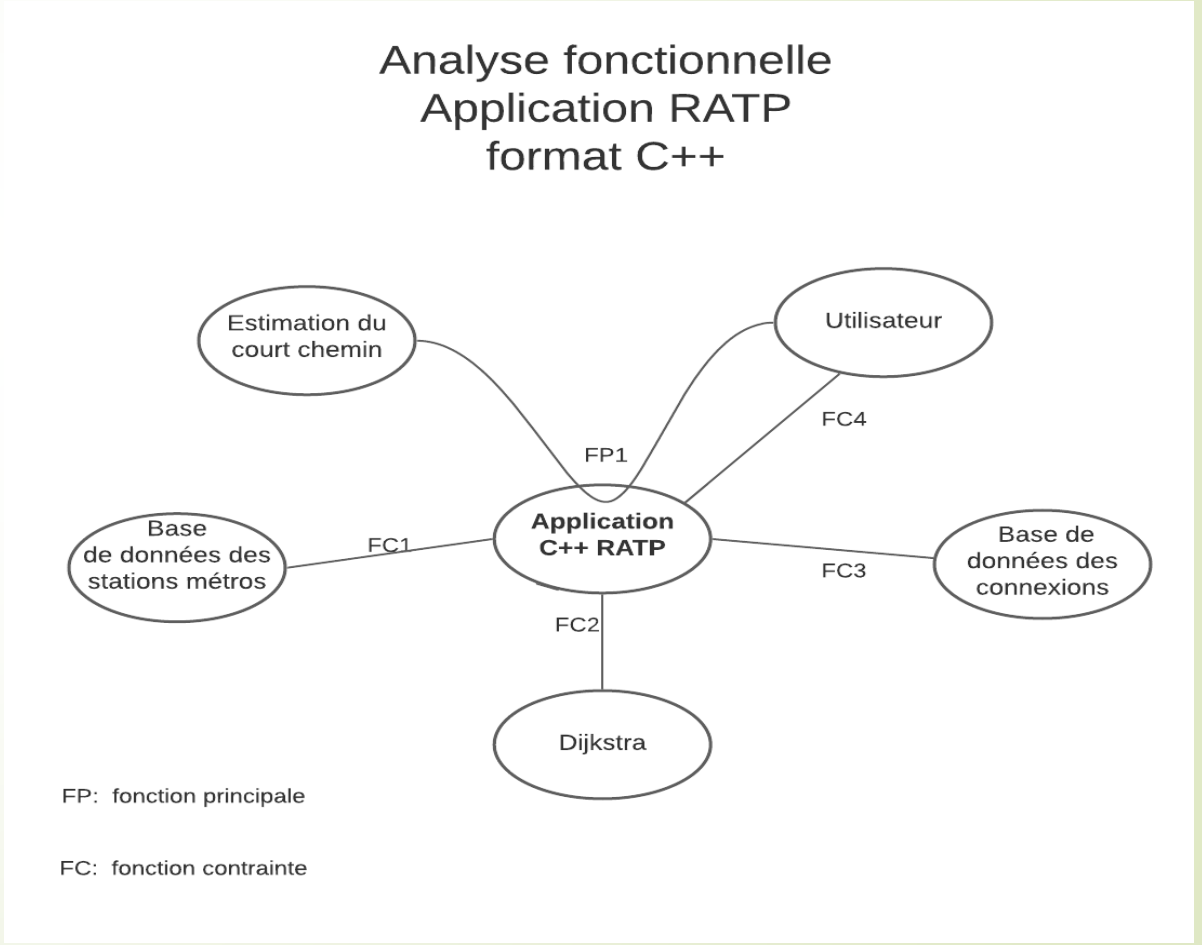
FP1: permettre au utilisateur de trouver le court chemin entre 2 stations.

FC1: doit respecter la structure de la base de données sur les stations.

FC2: Doit utiliser l'algorithme de Dijkstra

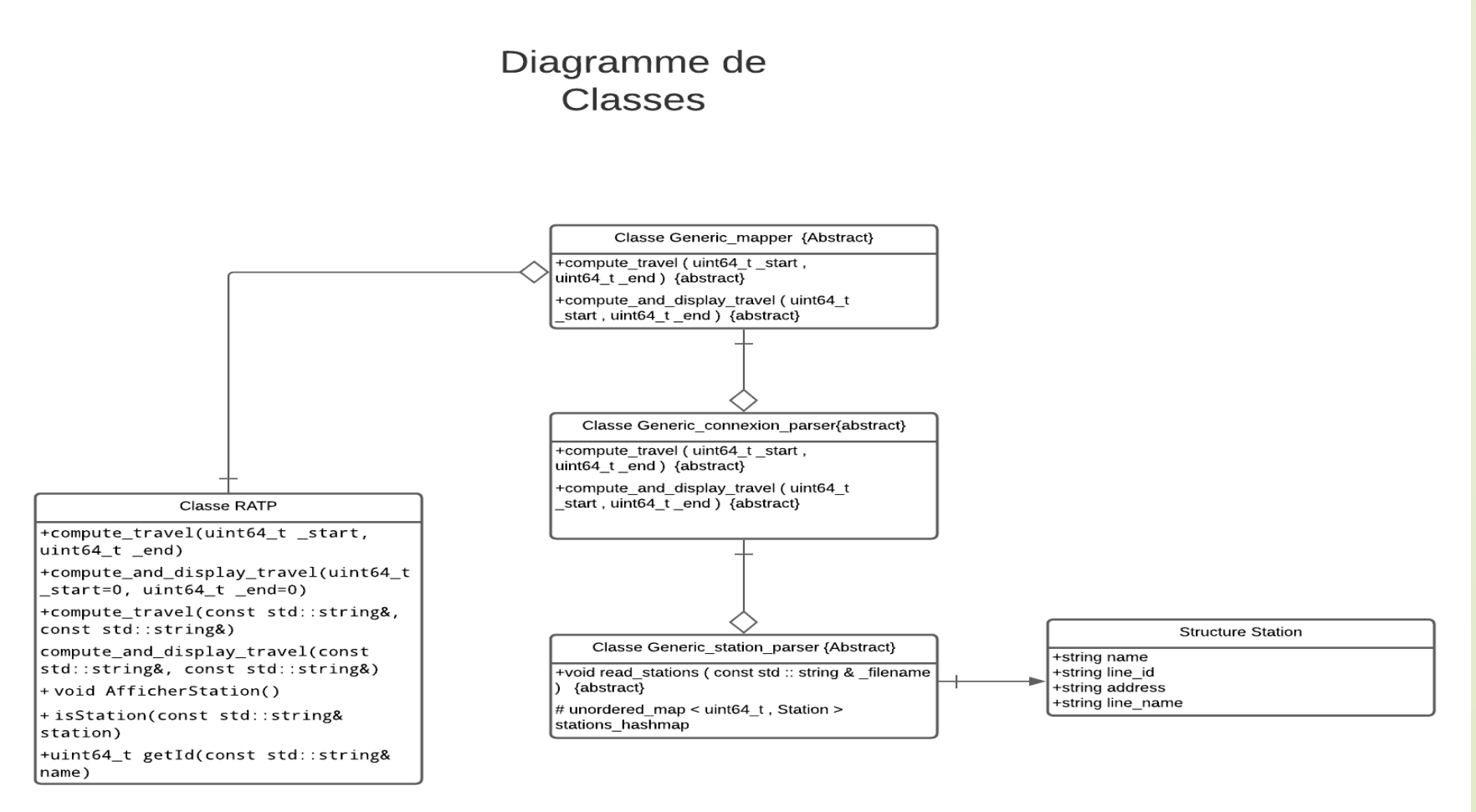
FC3: Doit respecter la structure de la base de données sur les connexions

FC4: Doit donner à l'utilisateur la possibilité de taper les noms des stations, d'afficher toutes les stations, et de quitter l'application.



B. Dijkstra

C. Architecture et diagramme de classes



B. Dijkstra

- **C. Implémentation:**
- **1. lecture des données :**
  - **1.1 Lire les stations:** lire les données des stations sous format csv, et enregistrer ces données dans un container unordred_map. Nous avons choisi ce type de container car il nous permets de travailler sur des clés qui ne sont pas forcément des entiers.
  - **1.2 Lire les connexions:** lire et stocker les données dans une table de hashage.
- **2. Implémentation de l'algorithme :**
  - **1. diffécultés:**
  - - diffécultés de choix des bons container dans l'algorithme Dijkstra.
  - - diffécultés sur l'utimlisation des containers.

    2. Solutions:

    - Consultation de la documentation c++.

C. Conclusion

- Ce projet m'a permet de découvrir un algorithme très puissant, qui a beaucoup d'utilisations dans la vraie vie et que n'utilise souvent quand on prend les transport.

- **Bibliographie:**

[1] https://fr.wikipedia.org/wiki/Algorithme_de_Dijkstra

[2] http://www.csl.mtu.edu/cs2321/www/newLectures/30_More_Dijkstra.htm

[3] https://www.cplusplus.com/reference/stl/

[4] https://openclassrooms.com/forum/sujet/implementation-algorithme-de-dijkstra

[5] https://www.techno-science.net/definition/6470.html

Indication et Read.me

- **Test de l'application et indications:**
- **1. éxécution:**
- dans ce projet, j'ai utiliser la compilation séparée, dans un fichier Makefile.
- Remarque: Si vous être sous Linux, il suffit de taper la commande make pour générer un fichier exécutable de l'application. Sinon, je joint un fichier exécutable nomé **projet.o**
- Et cette éxécutable s'éxécute tout simplement en tapant la commande **./projet** sur le terminal.
- **2. L'interface graphique:** l'utilisateur a la possibilité de taper une station de départ, et une station d'arrivée. et l'application estime le court chemain.
  - **Exemple de test:**
  - **Départ:** Jussieu
  - **Arrivée:** Madeleine
  - La commande **exit** pour quitter l'application
  - La commande **show** pour afficher les stations