

Motif / type de réunion : Réunion d'avancement	Lieu : Salle de TN
Présent(s) : Les quatre membres du groupe projet	Date / heure : 29/03/2023 13h15 - 14h00

I Ordre du jour

- 1) SWOT
- 2) WBS
- 3) État de l'art

II Informations échangées

II.1 SWOT

II.1.a Forces

- ▷ On a de l'expérience ensemble et on se connaît.
- ▷ Par exemple, préférer les deadlines courtes parce que Corentin travaille souvent à la dernière minute.

II.1.b Faiblesses

- ▷ L'associatif à TELECOM nous demande à tous beaucoup de temps.
- ▷ C'est pourquoi, il a été difficile de trouver un créneau pour la dernière réunion.

II.1.c Opportunités

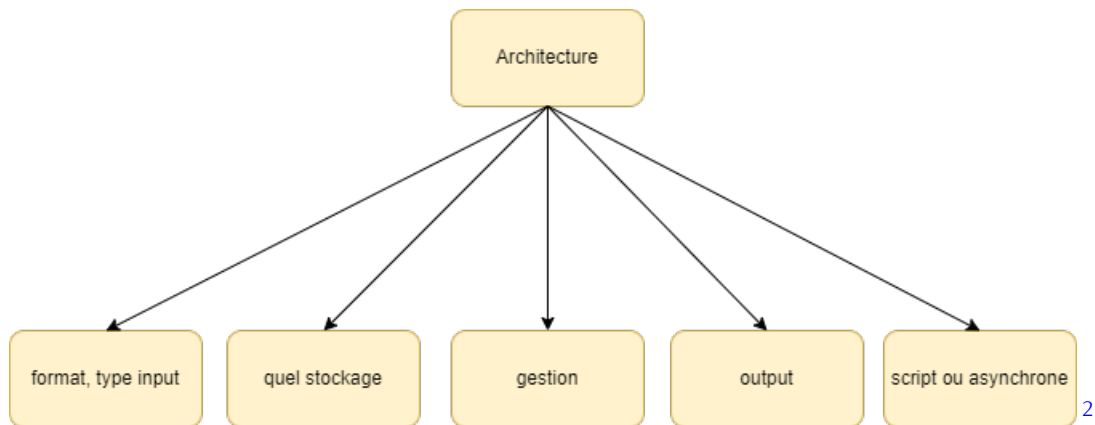
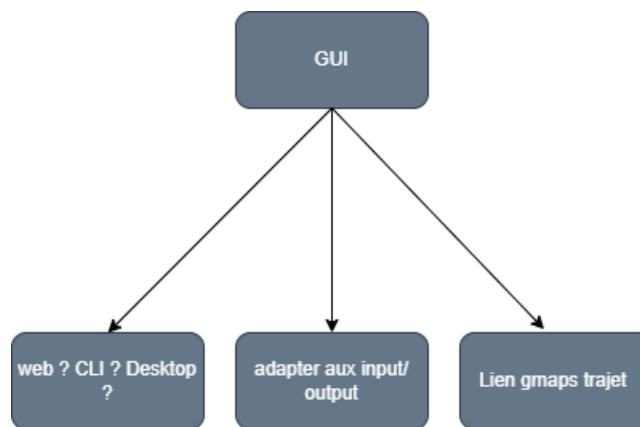
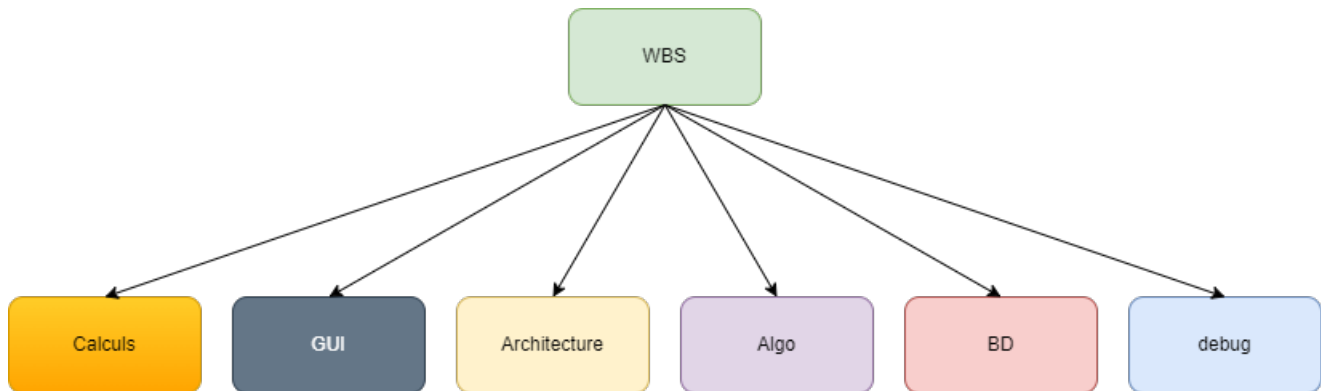
- ▷ Des locaux de Citroën ne se trouve pas très loin de TELECOM. Il pourrait être intéressant d'aller les voir pour se renseigner sur le sujet.
- ▷ Le sujet qu'on nous a donné est un sujet très important actuellement, ce qui nous permet d'avoir accès à beaucoup de ressource (sur Internet ou autre).

II.1.d Menaces

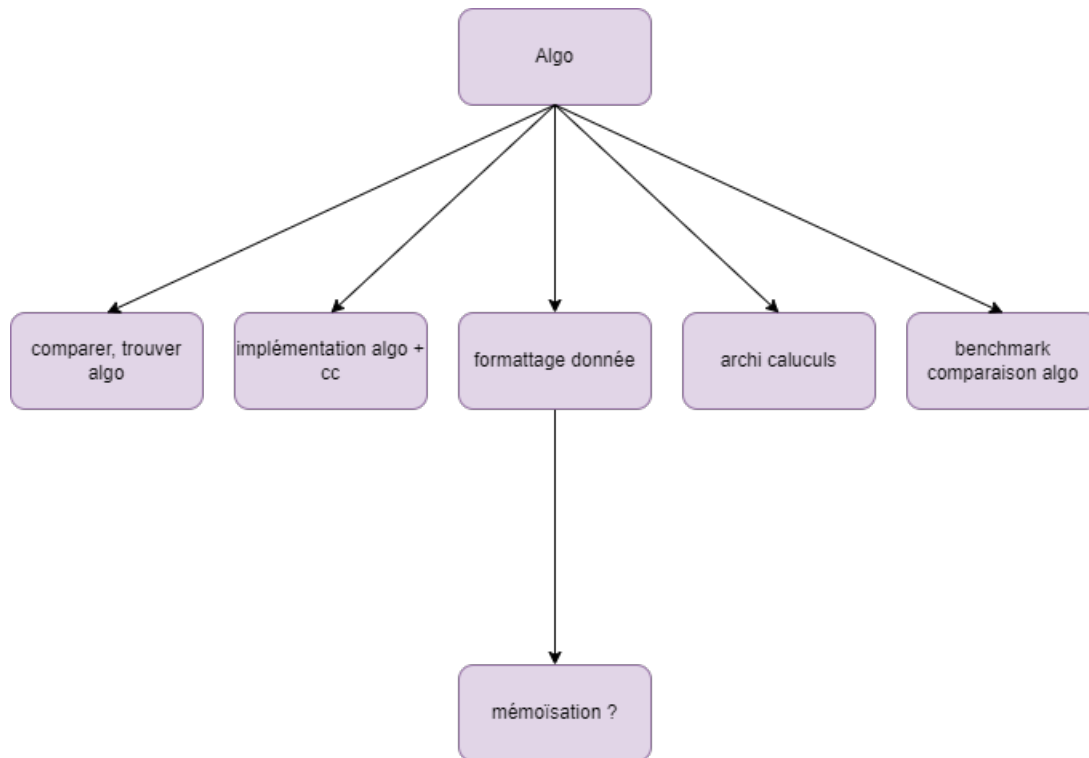
- ▷ Nous n'avons peut-être pas du matériel assez puissant pour faire tourner des algorithmes comme celui de Dijkstra sur une base de données aussi grosse que celle qui nous a été donnée. Nous devons alors chercher à optimiser ces algorithmes, voir chercher d'autres algorithmes plus puissants.¹
- ▷ Attention aux météorites

1. Voir commentaire dans la partie III. Remarques / questions

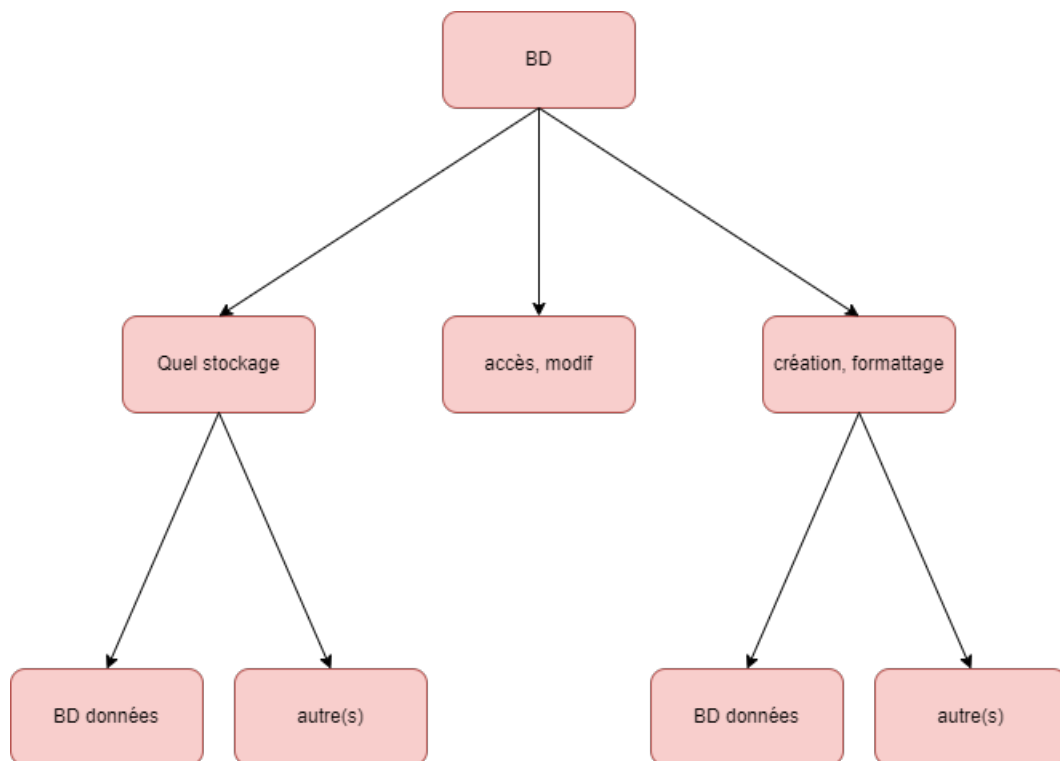
Le travail a été divisé en 6 parties :



2. Voir commentaire dans la partie III. Remarques / questions



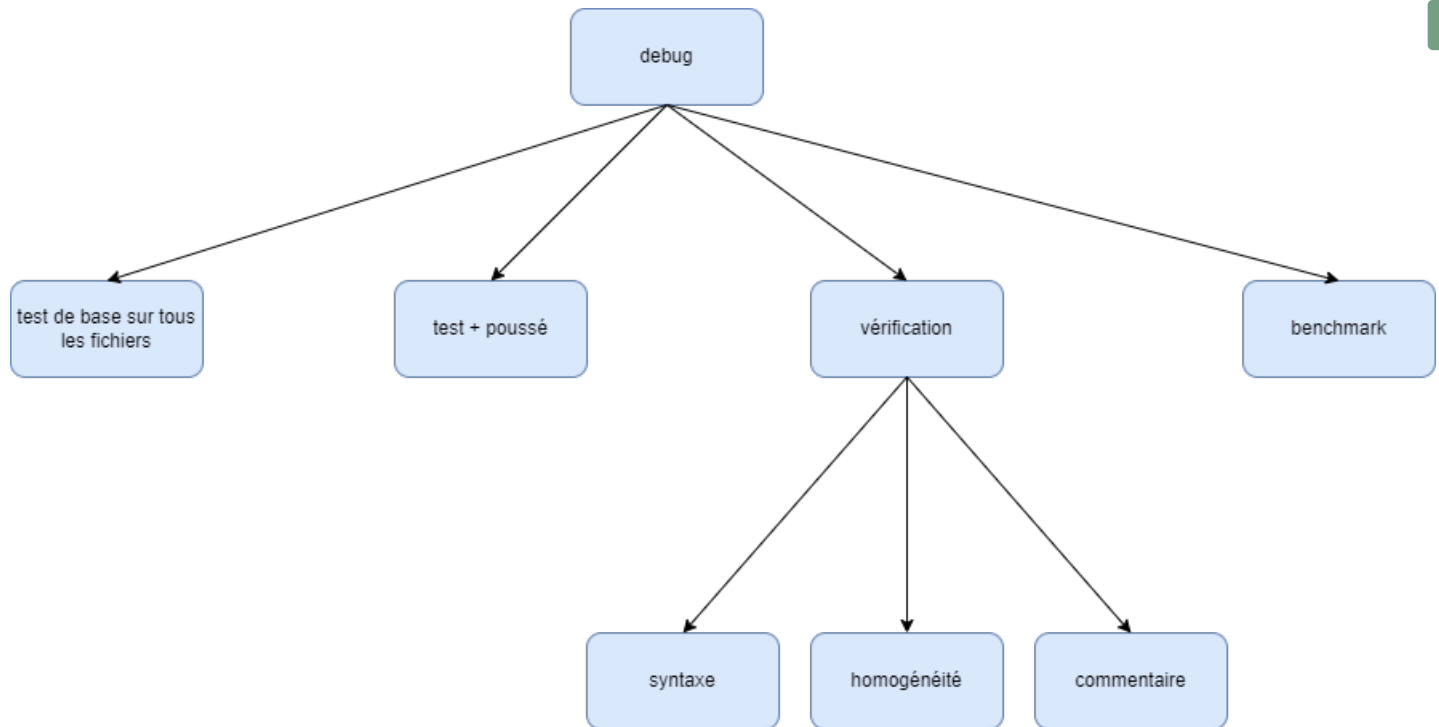
3



4

3. Mémoïsation ou autre technique

4. Autre(s) base(s) de donnée ajoutée(s) plus tard



II.3 État de l'art

II.3.a Le problème

- ▷ Le problème qui nous a été donné est un problème de recherche de plus court chemin.
- ▷ Problème du plus court chemin : soit un graphe orienté. Le problème du plus court chemin entre un sommet A et un sommet B revient à chercher le chemin de A à B où la somme des poids des arcs qu'il traverse est minimale.⁵
- ▷ Il faut donc choisir un algorithme qui trouve des solutions à ce problème.

II.3.b Algorithme

Les potentiels algorithmes pour ce problème sont :

- ▷ Dijkstra
- ▷ A*⁶
- ▷ Bellman-Ford
- ▷ Viterbi
- ▷ Floyd-Warshall⁶
- ▷ Johnson⁶

5. Si on ajoute des contraintes à ce problème comme des fenêtres de temps, le problème peut devenir NP-difficile.

6. Cet algorithme semble le plus efficace d'après les recherches

6. Si l'on veut tous les chemins à partir d'un point

II.3.c Application existante

Comme expliqué plus haut (cf. II.1.c page 1), ce problème a déjà été étudié. Il existe alors des applications permettant de répondre aux besoins de notre sujet, notamment :

- ▷ **Chargermap** : Son fonctionnement est très simple. Il faut d'abord lancer l'application sur son smartphone Android ou Apple, puis renseigner ses points de départ et d'arrivée. Vous devez ensuite préciser le modèle de votre véhicule, les niveaux de batterie souhaités au départ comme à l'arrivée et enfin, la vitesse maximale que vous prévoyez au cours du trajet. À partir de ces informations, l'algorithme calcule en une poignée de secondes le chemin le plus adapté. L'itinéraire à suivre et les bornes où s'arrêter sont ainsi affichés sur une carte. Une feuille de route vous indique également la distance à parcourir entre chaque escale et la durée de chaque recharge. Vous verrez qu'il n'est pas systématiquement nécessaire de faire un plein complet, un appoint de quelques dizaines de minutes suffisant la plupart du temps.
- ▷ **A Better Routeplanner** : Le fonctionnement de cet outil est globalement similaire celui de Charge-map. Lieu de départ et d'arrivée, niveaux de batterie, modèle de véhicule : on y renseigne les mêmes informations. A Better Routeplanner (ABRP) se démarque cependant en proposant un mode expert, qui ouvre accès à un panneau extrêmement complet de données à préciser : poids des bagages et passagers, météo, consommation moyenne personnalisée, type de chargeurs, etc. Contrairement à Chargemap, ABRP présente aussi l'avantage d'estimer les prix des recharges sur sa feuille de route. Les différences de temps de charge et de parcours entre Chargemap et A Better Routeplanner sont souvent le résultat d'estimations différentes de ses concepteurs. Le premier outil peut, par exemple, être plus ou moins optimiste sur la capacité d'un véhicule à recharger rapidement que le second.

III Remarques / questions

- ▷ Pour ce qui est de la menace de la performance de nos machines (cf. II.1.d page 1), il ne faut pas oublier que nous utilisons du C qui est un langage bas niveau. Ainsi, les calculs sont plus rapide avec ce langage qu'avec un langage haut niveau comme Python.
- ▷ **script ou asynchrone** (cf. II.2 page 2) pose la question : veut-on un code qui s'exécute comme un script, à savoir une fois pour un trajet, ou de manière asynchrone ; une fois lancer, il suffit de lui demander de faire un calcul pour qu'il le fasse même si un résultat précédent n'est pas encore calculé.
- ▷ Il faut qu'on définisse des conventions que tout le monde suivront.
- ▷ Rappel de l'importance de commenter le code.

Prochaine réunion : 07/04/23