

Rapports individuels évolutifs sur le travail réalisé.

Période de travail : 14/09/2022 => 25/09/2022

Manager de la période : Samuel

Collaborateurs : Lucas, Quentin et Claude.

Samuel :

Les tâches :

- Faire la composition de l'équipe et la répartition du rôle de manager.
- Rendre l'organisation de l'équipe.
- Ecrire une documentation pour expliquer PHP, Bootstrap etc
- Rendre la documentation de développement.
- Rendre le rapport sur la répartition du travail, les tâches et l'évaluation des effectifs.

(Partie plus détaillée dans mon fichier personnel car je suis le manager.)

Lucas :

Les tâches :

- Créer un dépôt GitHub.
- Préparer et tester la plate-forme de développement.
 - création des programmes Lancer php mac.command et Lancer php windows.bat
 - création d'un programme de téléchargement des fichier .tsv download.py
- Tester le déploiement du projet sur un serveur web Apache.

Ressenti :

Pour commencer, j'ai créé le dépôt GitHub en donnant tous les droits à tout le monde. Ensuite, pour faciliter l'utilisation de tout le monde, j'ai créé un script qui lance Php, cependant je n'ai pas eu le temps de faire celui pour Windows, d'autant plus que je n'avais pas de Windows sous la main.

Ensuite, j'ai créé le fichier 'download.py' qui sert à télécharger tous les fichiers du lien qu'on lui donne, les désarchiver et les placer dans le fichier 'download'.

Le manager a réussi à bien tout nous répartir, en nous donnant des missions en fonction de nos points forts et faibles.

Quentin :

Les tâches :

- Tester le déploiement du projet sur un serveur web Apache.
 - analyse des données
 - création de l'UML concernant les tables de la base de donnée
 - création des tables de la base de donnée en SQL
 - mise en place des dépendances
- Préparer et tester la plate-forme de développement.
 - choix de l'IDE VScode
 - mise en accord avec l'équipe sur l'utilisation de PostgreSQL pour le SGBD
- Faire des recherches sur les CGU de la BDD.
 - premières recherche à approfondir lors du retour à l'IUT

Ressenti :

On a commencé par se mettre d'accord assez rapidement sur les technologies que l'on allait utiliser comme le SGBD PostgreSQL ainsi que les scripts fait par mes camarades. Après quoi j'ai dû choisir mon IDE. J'ai alors hésité et fait des recherches sur les différents IDE pour SQL et ai fini par porter mon choix sur VS code car au vu de ce que l'on va être amené à faire je le trouvais intéressant. Après quoi j'ai commencé à créer l'UML et l'SQL en simultanée. C'était l'enfer de trouver un agencement des tables avant notre dernier cours de base de données. Celui-ci m'a beaucoup aidé. A vrai dire, j'étais complètement bloqué et j'ai passé beaucoup trop de temps à me décider sur la création de l'UML. Après ce cours j'ai décidé de jeter tout ce que j'avais commencé à faire afin de tout remanier et partir sur de bonnes bases car partir sur des bases fragiles pourrait être dérangeant pour la suite du projet. Après avoir remanié toute la structure des tables et de les avoir construites en SQL je me suis mis à commencer les recherches par rapport aux CGU de la base de données. Je n'ai pas eu beaucoup le temps d'avancer alors je continuerai cette étape lors de mon retour après ma période en entreprise.

Claude :

Les tâches :

- Tester le déploiement du projet sur un serveur web Apache.
- Préparer et tester la plate-forme de développement:
 - tester plusieurs serveurs de base de données (MariaDB, SQLite, PostgreSQL)
 - trouver un serveur de base de données qui permet aussi l'interaction avec Python (les 3 cités ci-dessus le permettent)
 - tester les données Imdb sur plusieurs serveurs de base de données
- Faire des recherches sur les CJU de la BDD:

- recherche à approfondir lors du retour à l'université
- Écrire la documentation pour expliquer l'utilisation de JQuery, Python, etc.
- Création d'un script Python qui permet de :
 - convertir les fichiers tsv en csv
 - ajouter les accolades dans certain fichier qui feront correspondre le type array

Ressenti :

J'ai pu parfaitement mettre les données Imdb sur SQLite cependant, 2 problèmes surviennent : 2 types d'attributs ne sont pas respectés puisque SQLite ne possède en effet pas d'attribut tableau et booléen, l'attribut booléen ne pose pas vraiment problème puisqu'il peut être remplacé le plus compliqué reste le type tableau.

Sur SQLite, une alternative est possible qui sont les types JSON qui prennent en charge les tableaux, mais les stockent sous forme de chaîne, donc pas efficace en termes d'espace pour les nombres et les dates. SQLite n'est pas vraiment un serveur de base de données, c'est une librairie, il sera par conséquent, plus compliqué à mettre en place comparée aux différents serveurs de base de données.

Comme vous pouvez le voir, SQLite pose déjà énormément de problème puisque c'est à moi de m'adapter aux "faibles possibilités " que nous offre SQLite. Je décide donc de délaisser SQLite pour l'instant (dernier recours) pour les plusieurs raisons mentionnées ci-dessus d'autant plus que tous les membres du groupe ne savent pas tous utiliser SQLite.

Nous avons donc choisi d'utiliser PostgreSQL cependant si nous ne parvenons pas à l'utiliser, je pourrais à tout moment utiliser SQLite. Un autre problème rencontré concerne le type tableau (encore) mais cette fois-ci pour MariaDB et PostgreSQL. Les 2 serveurs de base de données ne permettent pas d'importer certains fichiers où le type tableau est présent, car les accolades annoncent un attribut de type tableau cependant sur les fichiers. Les accolades ne sont pas présentes, par conséquent, MariaDB et PostgreSQL ne les reconnaissent pas en tant qu'attribut tableau. Je vais donc devoir extraire les données avec la librairie Pandas de Python afin de pouvoir écrire dessus et ajouter les accolades aux endroits nécessaires. Pour l'instant, c'est mon but cependant, il se peut que je trouve une alternative d'ici là, cette tâche n'est donc pas officielle sauf si je ne trouve pas d'alternative.