Choix technologiques

Projet de fin d’étude

Denuit Maxime, Degrève Olivier, Dubois Corenthin, Tavernier cedric,

Table des matières

[1. Choix Technologique : 1](#_Toc469946265)

[1.1. Back-End : 1](#_Toc469946266)

[1.2. Front-End : 1](#_Toc469946267)

[2. Diagramme d’architecture : 1](#_Toc469946268)

[3. Outils utilisés : 1](#_Toc469946269)

[4. Bibliographie : 1](#_Toc469946270)

# Choix Technologique :

## Back-End :

Nous utiliserons un back end en nodeJS. L’avantage de cette technologie est qu’elle est déjà très simple à apprendre. Dans notre groupe, 2 personnes connaissent déjà le NodeJS grâce au cours enseigné et les 2 autres possèdent des connaissances en JavaScript, ce qui rendra la tâche assez simple si l’un d’entre d’eux devait l’apprendre

Et comme NodeJS est assez récent, la documentation est très facile de compréhension et facilement trouvable sur le site de NodeJS.

Une autre chose intéressante à propos de Node est sa grande modulabilité. Il existe beaucoup de librairies externes permettant d’effectuer des tâches pour un backend. Que ce soit des librairies serveurs, de requêtes http/https ou autre. Ce même genre de librairie existe aussi pour le front pour avoir une interface simple à mettre en place (Exemple : AngularJS) ou pour avoir des fonctions temps réel grâce aux websockets de node (Exemple : High Chartz)

Si l’on compare NodeJS au Java pour le backend, il y a clairement plusieurs avantages. D’abord, l’utilisation de Node permet le développement d’une application de type « full stack » ce qui permet de ne pas avoir de middleware pour convertir d’éventuel données transmises par le front. (ce qui est le cas de java, il faut convertir le json avec Genson).

Ensuite, NodeJS a l’avantage d’être plus rapide que java pour faire de l’I/O et est aussi non-bloquant ! Il y a une bonne gestion de la concurrence en Node. Bien sûr, on peut implémenter de la concurrence en Java mais cela requiert plus de temps et engendre potentiellement plus de problèmes si mal implémenté.

NodeJS a aussi l’avantage d’avoir une très bonne scalabilité. En effet, on peut déployer rapidement une application Node pour une entreprise que ce soit une application ou mobile. De plus, nodejs à l’avantage d’avoir de très bonnes performances et d’être assez rapide lorsqu’on effectue des requêtes.

Cependant, NodeJS est là principalement pour effectuer des tâches peu complexes. En effet, si l’application requiert des calculs complexes, il serait alors préférable de donner la charge à un autre processus node alors que Java n’aurait pas de problème pour le faire en un seul processus.

## Front-End :

Pourquoi avons-nous décidé d’utiliser de l’HTML5/CSS3 ?

Nous avons décidé d’utiliser la dernière version d’HTML5/CSS3 car celle-ci nous permet d’utiliser toutes les dernières fonctionnalités.

Un inconvénient mineur sera qu’on devra tenir en compte que certains navigateurs ne prendront pas en compte la dernière version de l’HTML5. Pour régler ce problème nous devrons donc coder en fonction du navigateur le moins « up to date ».

Pourquoi nous n’avons pas utilisé bootstrap ?

Il nous semble intéressent de pouvoir créer de l’html / css « fait maison », ce qui nous permet d’utiliser l’html/css dans nos codes JAVASCRIPT facilement, avec les balises que nous voulons. Il nous semble intéressant de créer nous même un site réponsif et non d’utiliser un Template tout fait, comme proposé par bootstrap.

Comme language de programmation front-end nous avons décidé d’utiliser JAVASCRIPT 1.8 car :

Le javascript permet d’agir directement et ne dois pas attendre que les serveurs envoient une réponse. Ceci accélère l’ouverture des sites web sur les navigateurs des clients. Le javascript ne nécessite pas un programme d’interprétetation comme par exemple Flash player, en plus de ça, le javascript n’occupe pas une grande place sur disques des sites web.

# Diagramme d’architecture :

# Outils utilisés :

# Bibliographie :