

ELAIN Kevin
ELIE Benoît

JUSTIFICATION DES CHOIX TECHNIQUES

SOMMAIRE

1) Introduction.....	2
2) GIT	2
2.1) En lui même :	2
2.2) Face aux autres :	2
3) Dropbox	2
3.1) En lui même :	2
3.2) Face aux autres	2
4) Html5/CSS3/media-query et responsive design	3
4.1) En eux mêmes :	3
4.2) Face aux autres :	3
5) AJAX	3
5.1) En lui même :	3
5.2) Face aux autres :	3
6) MySQL	3
6.1) En lui même :	3
6.2) Face aux autres :	3
7) BootStrap.....	4
7.1) En lui même :	4
7.2) Face aux autres :	4
8) NoSql – version personnel.....	4
8.1) En lui même :	4
8.2) Face aux autres :	4

1) INTRODUCTION

Les choix techniques pris lors de ce projet ont été les suivants:

- L'utilisation de GIT pour la gestion de version
- La mise en place d'une synchronisation via Dropbox
- L'utilisation de Html5, CSS3, des media-query et du responsive design
- La mise en place d'une couche AJAX
- La mise en place d'une BDD sous MySQL

2) GIT

2.1) EN LUI MÊME :

Une gestion des versions simple, par auteur et par ligne, le tout daté et avec les fichiers concernés.

La possibilité d'un travail en branche et d'une navigation entre les versions.

2.2) FACE AUX AUTRES :

GIT a l'avantage d'être multiplate-forme, de disposer d'un mode en ligne de commande simple à prendre en main et qui ne change pas entre les plate-formes.

Il se caractérise aussi vis-à-vis de SVN par la possibilité d'un travail en local, sans se référer à un serveur principal.

3) DROPBOX

3.1) EN LUI MÊME :

Synchronisation rapide, possibilité de restaurer des versions antérieures, bon espace de stockage disponible.

3.2) FACE AUX AUTRES

Le principe de dossier synchronisé est simple à prendre en main et automatique.

L'espace disponible (2Go) est aussi adapté à un travail sur des sources logiciel.

Il est lui aussi multiplate-forme et dispose même d'une interface web.

4) HTML5/CSS3/MEDIA-QUERY ET RESPONSIVE DESIGN

4.1) EN EUX MÊMES :

Ils permettent une gestion simple des différentes tailles d'écrans que peut rencontrer une application web.

Le fait que l'application s'adresse à des employés permet de s'assurer que les machines pourront être mise à jour pour le support des technologies requises.

4.2) FACE AUX AUTRES :

Le responsive design présente l'avantage de pouvoir s'adapter à la fois à un poste fixe doté d'un affichage supérieur à 17", comme à celui d'un smartphone ou d'une tablette.

L'application n'est donc qu'à développer une seule fois, avec une compatibilité maximale.

Ces technologies permettent de tirer profits de toute la surface d'affichage disponible, ainsi que de mettre en place des améliorations d'ergonomie.

5) AJAX

5.1) EN LUI MÊME :

Il permet d'éviter des rechargements intempestifs de page entière et permet donc à l'utilisateur une navigation plus fluide par la modification en temps réel de la page qu'il consulte.

5.2) FACE AUX AUTRES :

Il reste assez bas niveau et ne demande pas le chargement de bibliothèque externe (jQuery).

La manipulation du DOM étant bien développée en JavaScript, l'AJAX permet une dynamisation rapide des contenus sans complexifier réellement le code.

6) MYSQL

6.1) EN LUI MÊME :

SGBDR dans le trio de tête, il permet une gestion de base via les moteurs les plus courants.

Distribué sous licence GPL il est à la fois utilisé par le grand public et les professionnels.

6.2) FACE AUX AUTRES :

Il est derrière Oracle, mais présente l'avantage d'être plus simple à administrer et mettre en place (au détriment d'une petite perte de fonctionnalité).

Il est aussi moins lourd en termes de ressources consommées pour le serveur.

7) BOOTSTRAP

7.1) EN LUI MÊME :

C'est un kit de CSS, de JQuery et de modèle HTML. Il permet une mise en place rapide d'un site aux normes responsive design.

Il se compose sous la forme de plug-in rapide à appeler dans le code.

7.2) FACE AUX AUTRES :

Très rapide à prendre en main, personnalisable rapidement par des surcouches.

Son aspect « responsive-friendly » l'a rendu rapidement incontournable sur le web mobile. Il permet notamment de gérer des menus et des block.

Peu de concurrence, si ce n'est la version « from scratch » ou des framework bien plus lourd et moins souple.

8) NoSQL – VERSION PERSONNEL

8.1) EN LUI MÊME :

C'est une petite class de gestion de données en fichier/dossier. Le but est de rendre la manipulation rapide, simple et avec le moins de contrainte possible

Des méthodes de base sont mises à disposition et permettent dans notre cas d'ajouter des mots de passe aux comptes des Techniciens, sans avoir à retoucher la base de données.

8.2) FACE AUX AUTRES :

Une solution SQL standard est tout à fait envisageable dans ce cas-ci, mais le travail se basant sur une BDD commune, le fait d'y établir nos propres modifications sans en aviser les autres développeurs peut poser problème. Ce choix permet donc une fusion des projets sans aucun risque de conflit.

Les grandes solutions NoSql sont quand à elles un peu complexes à mettre en place (car plus optimisées), ce qui justifie le choix d'une solution personnelle.