Rapport de projet web
Application pour la gestion de partage de fichiers en sein d’une formation.

Théo Ponthieu IG3
Polytech Montpellier
Année universitaire 2017/2018
www.dolcevitech.fr

SOMMAIRE

Introduction :

Au sein d’une formation, les membres sont souvent amenés à s’entraider et à partager des documents. Les outils de partage de fichiers sont de plus en plus performants. Ils permettent de partager tout type de fichiers avec n’importe qui.

Comment partager des fichiers de façon organisés entre de nombreuses personnes ?

L’application que j’ai réalisée permet d’améliorer l’organisation et les droits des utilisateurs qui se partagent les fichiers. L’objectif est d’améliorer la distribution et l’édition de fichiers PDF. Chaque utilisateur, s’il possède les droits, peut lire, partager, supprimer ou modifier des documents. Les documents sont organisés par thème pour chaque spécialité de chaque école. Les administrateurs des spécialités peuvent gérer les droits des participants et en inviter de nouveaux. Chacun est libre de consulter les documents directement sur le site ou de les télécharger pour les consulter hors connexion.

Après avoir présenté le cahier des charges et les objectifs, je détaillerai les outils techniques utilisés. Vous trouverez à la fin un retour sur l’expérience que m’a apportée la réalisation de ce projet.

I) Les objectifs.

1.1) Contexte.

Depuis plusieurs années les différentes filières de Polytech archivent des cours, des TDs, des annales…etc. Ces documents se transmettent entre les générations d'étudiants sur des drives. Drive ne permet pas d'avoir la main sur le détail des droits des utilisateurs et ceux qui le consultent. Les fichiers et documents sont mals mis à jour car peu de personnes détiennent les droits de modification.

L'application web vise à résoudre ces problèmes en permettant aux administrateurs d'une spécialité de gérer finement les droits des utilisateurs.

1.2) Les outils déjà existants.

On peut trouver en ligne une multitude d'applications qui permettent le partage de fichiers. Après des recherches je n'en ai pas trouvé qui implémente une gestion des droits des utilisateurs de façon simple et gratuite. Le système de partage est basique et ressemble à un drive avec la particularité de pousser les utilisateurs à organiser les documents proprement en obligeant un certain pattern.

1.3) Environnement d'exploitation.

Cette application a été pensée directement pour les étudiants de Polytech. Le pattern suit une certaine logique (Spécialité/Matière/Type de document) qui ne peut pas correspondre à toutes les utilisations. Elle peut donc être utiliseé par toutes les écoles Polytech ou autre qui suivent cette logique d'organisation.

II. Les fonctionnalités :

2.1) Possibilités (côté client)

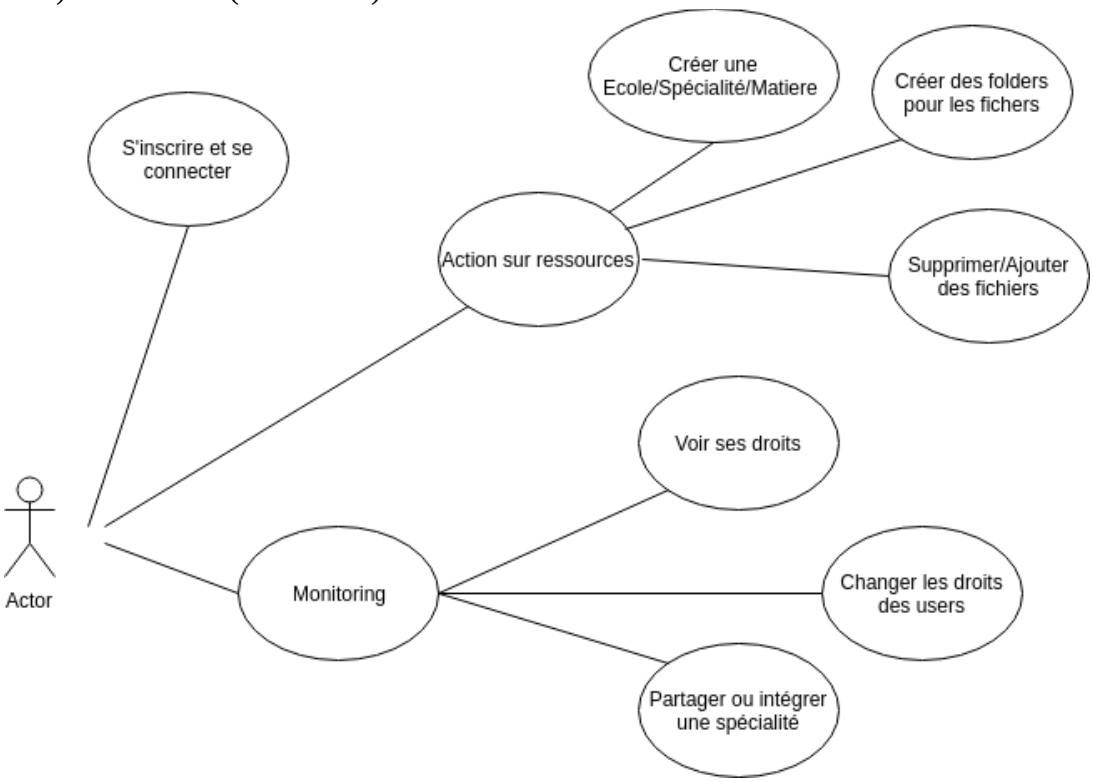

Figure 1 :Diagramme des fonctionnalitées

Un utilisateur doit s'inscrire avant de pouvoir utiliser le système de partage. Lorsque l'utilisateur est connecté, il peut créer un dossier pour sa spécialité si elle n'existe pas. Le créateur de la spécialité obtient tout les droits sur celle-ci. Il peut ajouter, supprimer les dossiers et fichiers de cette spécialité. Il peut gérer les droits des autres utilisateurs. Il peut créer des liens de partage qui lui permettent d'inscrire d'autres utilisateurs dans sa spécialité avec les droits de base qu'il souhaite.

2.2) Possibilités (côté serveur)

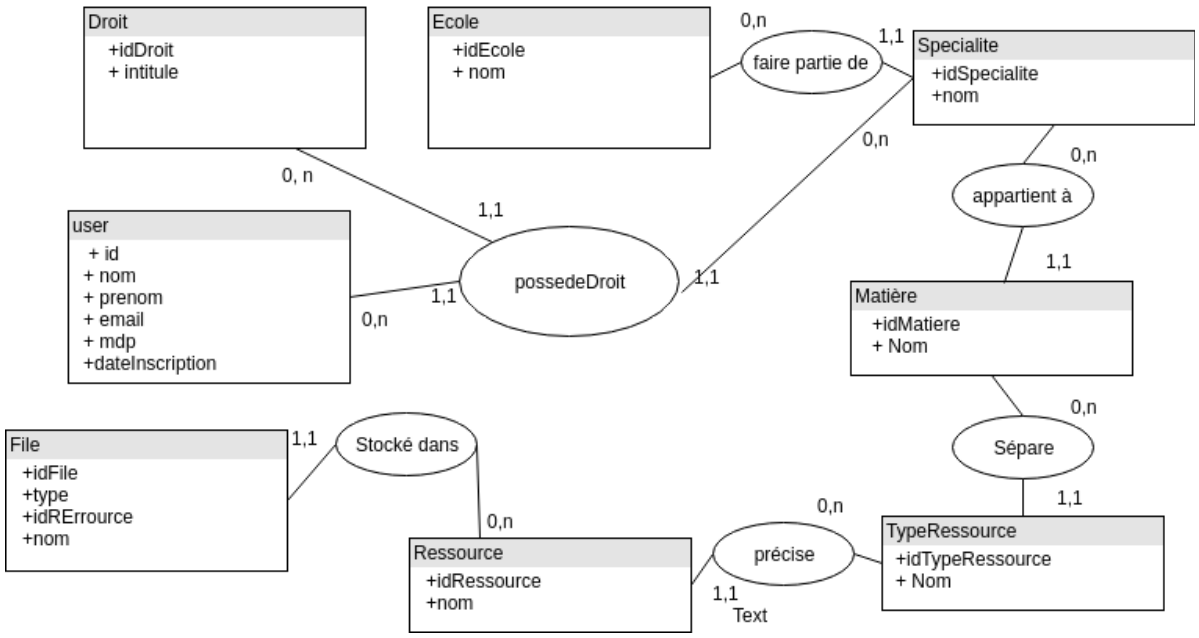

Figure 2 : MCD de la bd mysql

Un utilisateur peut avoir des droits pour n'importe quelle spécialité. L'arborescence est assurée par les tables Matières, TypeRessources, Ressource. Elles peuvent être considérées comme des dossiers. Le dossier ressource peut contenir un nombre indéterminé de « File » que les utilisateurs peuvent s'échanger.

III. Outils et structures utilisées :

3. 1) Les langages utilisés :

J'ai utilisé Angular 6 pour le « frontEnd » et nodeJs avec le framework Express pour le « backEnd ». J'ai déjà réalisé un projet personnel avec ces technologies. Je souhaite consolider mes connaissances et mon expérience sur ces outils. NodeJs et Express ont une grande communauté sur les forums d'aide et plusieurs tutoriels gratuits sont mis à disposition. Angular 6 est très pratique et puissant pour réaliser des applications « single page » très réactive.

Le design du site a été réalisé à l'aide de « Twitter Bootstrap » et de « Materialize ». Je me suis servi de Materialize pour les formulaires d'inscription et de connexion. Bootstrap m'a servie énormément pour l'organisation et l'affichage des modules et fonctionnalités sur une page.

La base de donnée utilisée est MySql, créée à l'aide de l'interface graphique phpMyAdmin. Je connais très bien MySql que j'ai utilisé dans tout mes précédents projets. Une base de donnée MySql est gratuite et les fonctionnalités qu'elle implémente suffisent largement pour ce type d'application.

3. 2) Structure Redux :

C'est la première fois que j'utilise une architecture type Redux. J'ai choisi cette architecture que je ne connaissais pas car je voulais découvrir et apprendre une nouvelle technologie pendant le projet Web. « L'architecture redux met en place un Store global qui permet à chaque composant d'avoir accès aux informations et réagir de façons différentes en fonction de l'action souhaité ». Les différentes descrition de l'architecture Redux m'ont donné envie de tester cette organisation. Angular met a disposition des librairies qui permettent de très bien gérer cette architecture.

L'architecture est de type SOFEA, la page est créée côté client et le backend permet d'avoir accès à des services.

3. 3 Sécurité :

Je n'ai aucune base en sécurité Web. Je me suis renseigné et j'ai essayé de faire le maximum pour garantir l'intégrité des données des utilisateurs ainsi que leur cloisonage. J'ai vu que Angular permet de renforcer la sécurité, notamment contre les injections SQL. Je pense que cette partie est celle qui nécessite le plus de travail et d'apprentissage, à titre personnel.

Les mots de passe sont dans un fichier de configuration annexe pour éviter qu'ils apparaissent dans le fichier du serveur.

3.4 Déploiement :

J'ai développé entièrement l'application en local avec quelques tests sur un serveur distant pour vérifier que tout fonctionne. Le serveur distant est une micro instance EC2 louée sur Amazon Web Service. J'ai choisi une micro instance car elle permet de faire tourner l'application de manière fluide et AWS permet de faire migrer son site vers une plus grosse instance si besoin. De plus la

micro instance est flexible. Si le trafic augmente, l'instance devient payante mais permet de suivre le besoin.
L'instance tourne sous Ubuntu. Le serveur est aussi nodeJS v8.11.1/express v4.15.5.

3. 5 Scalabilité

Les ressources sont stockés sur le serveur, seul les liens uniques sont stockés dans la bd pour retrouver le document. Si le volume de données devient trop important, je déplace la base de donnée sur un serveur dédié avec de plus gros volume de stockage. Si le trafic devient trop important et que je dois multiplier les serveurs, l'application reste fonctionnelle.

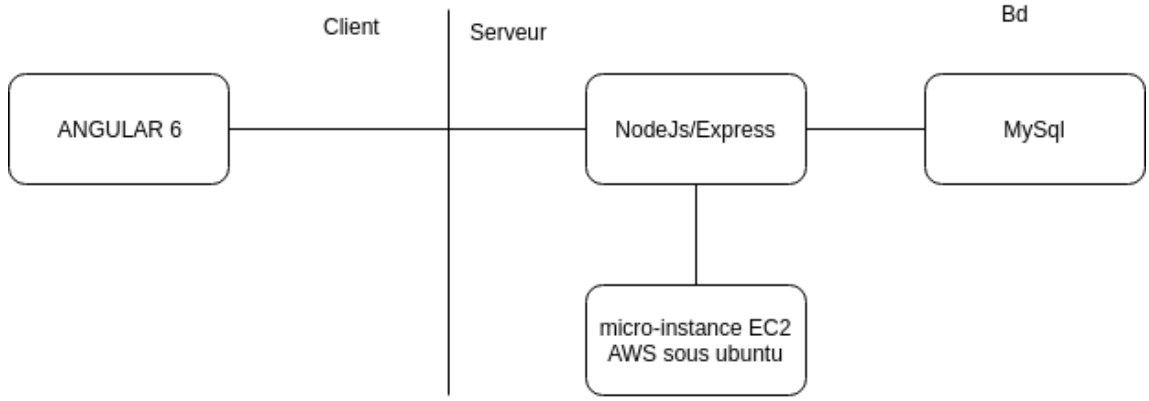

Figure 3 : Architecture matérielle (peut ou pas de trafic).

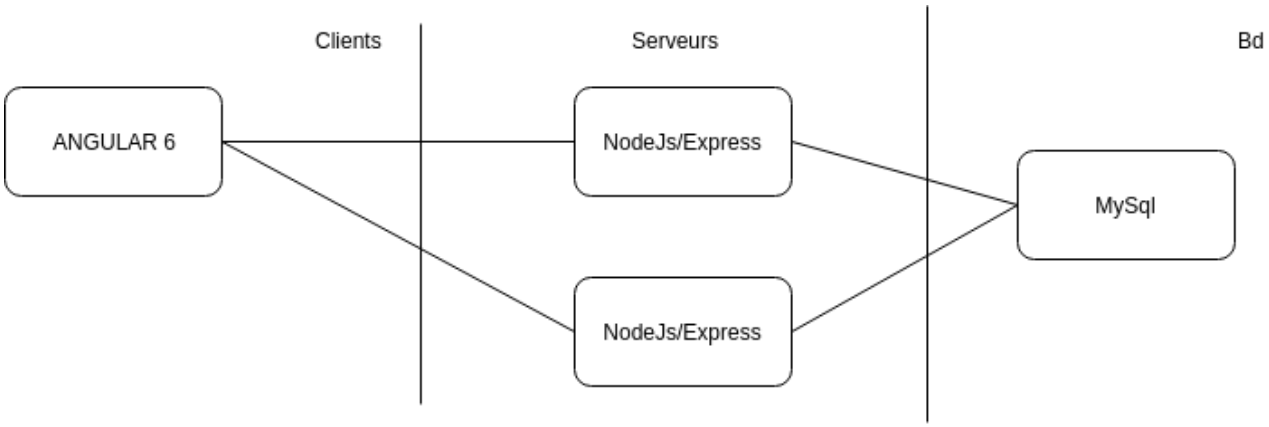

Figure 4 : Achitecture matérielle (beaucoup de trafic),

Mon application utilise « jwt authentification » qui permet de donner un token à l'utilisateur. Ce token est ensuite stocké dans le Store pour permettre l'authentification du client. Peu importe sur quel serveur l'application tourne, l'utilisateur pourra toujours être authentifié grâce à ce token.

IV Planification du projet :

4.1 : Conception cahier des charges :

Cette phase a été la plus longue. Elle devait durer initialement 5 jours, puis j'ai dû y revenir à cause des erreurs de conception que j'avais fait dû à manque d'expérience. J'ai dû y revenir ensuite deux fois pour retravailler la conception de fonctionnalités mals conceptualisées

4.2 Phase d'installation de l'environnement de développement :

J’avais déjà un environnement de developpement avant de commencer ce projet. Cette phase fût très courte et le deploiement de l’application sur le serveur distant prit peu de temps.

4.3 Phase de code :

La phase de code qui devait initialement durer 5 jours fût finalement entrecoupée de phases de conceptions. Elle a donc durée un peu moins longtemps que prévu, quelques fonctionnalitées n’ont pas eu le temps d’être implémentées.

V Fin de projet et retour sur la réalisation :

4.1 Les points positifs :

Ce projet m’a permis de découvrir de nouvelles choses dans le domaine du développement web et m’a donné envie d’approfondir mes connaissances ainsi que l’envie de continuer ce projet. Ce projet m’a appris à gérer mon sommeil et mon stress. J’ai appris à être productif et concentré pendant de longues périodes sous stress.

4.2 L’avenir du projet :

Cette application a été developpée en peu de temps. Il reste plusieurs fonctionnalités que j’aurais souhaité implémenter comme un tchat en dessous des ressources pour poser des questions ou la possibilité de rédiger directement des documents sur le site et encore plein d’autres.

L’aspect graphique doit encore être travaillé pour que l’application soit agréable à utiliser. Je pense apprendre à réaliser des animations pour que l’application paraît fluide et réactive.

4.3 Post-Mortem :

Avant de commencer ce projet, je n’avais aucune expérience sur l’utilisation d’une architecture Redux. J’ai perdu beaucoup de temps sur des erreurs trivialles et j’ai fait des erreurs de conception qui m’ont fait faire des rollbacks dans le code. J’ai dû recommencer plusieurs fonctionnalités qui n’était pas été correctement implémentée.

J’ai toujours envie d’apprendre et de découvrir de nouvelles technologies. Je pense que la prochaine fois je prendre une technologie que je connais mieux pour pouvoir avancer plus rapidemment et ne pas faire de grosses erreurs liés à mon manque d’expérience, lorsqu’il y a une « deadline » très proche.

Conclusions :

J’étais plustôt confiant avant le début du projet et jusqu’à la fin de la phase de conception. Quelques problèmes m’ont retardé et cela m’a permis de prendre du recul.

J’ai trouvé cette remise en question très enrichissante et j’aborderai les futurs projets de manière différente. Je trouve que le résultat est encourageant mais mérite d’être approfondi.