

# Share My Photos - Rapport

## TAULEIGNE Samuel

### Retour d'Expérience & Présentation des Choix Technologiques

L'application Share My Photos que j'ai développée permet à un utilisateur de :

- Déposer, modifier, détruire des photos (chacune d'elles étant associée à un nom et une description) photo par photo ou par lots
- Rechercher une photo (ou des photos) en fonction d'un mot présent dans la description ou dans le nom (la recherche n'est pas sensible à la casse)
- Afficher les photos en mode diaporama
- Consulter le diaporama en mode synchronisé entre plusieurs clients

Ce projet est pour moi une réussite car j'ai trouvé les ressources pour le mener au bout en cochant toutes les cases des fonctionnalités demandées alors que je ne m'en sentais pas capable au départ.

De plus, certaines particularités de mon code me rendent assez satisfait d'avoir trouvé des ressources dans mes propres connaissances sans avoir besoin du travail d'autres personnes (sans importer d'autres bibliothèques toutes faites). Je sais que c'est une étape parfois obligatoire pour gagner en rapidité, mais ça a été beaucoup plus formateur et apprenant pour moi de procéder de la sorte.

Je vois tout de même certains points d'amélioration dans ce travail tels que les échanges avec le client (que je trouve très intéressants et primordiaux mais que j'ai eu de la difficulté à prévoir étant donnée ma difficulté à démarrer sur ce projet) et le design de l'application que j'ai gardé très simple et intuitif mais qui aurait été amélioré par des échanges avec le client qui aurait eu ses propres exigences.

**La vidéo de démonstration est ici : <https://youtu.be/GrpUxcvJvMI>.**

**Le code source se trouve sur mon GitHub :**

**<https://github.com/SamuelTauleigne/ShareMyPhotos>.**

#### ➤ Base de Données & Stockage des Photos

Concernant le stockage des données, nous n'avions essentiellement à traiter uniquement un lot de photos accompagnées d'un commentaire. Le but de l'application étant de partager des photos de vacances, il peut y avoir une quantité assez grande de données. Alors, j'ai choisi d'utiliser une base de données MongoDB pour mettre en pratique le cours correspondant vu à Télécom Saint-Etienne : il permet de stocker aisément ce type de données et en quantité si cela devenait nécessaire.

Concernant la structure de la base de données, je me suis permis d'ajouter un nom à chaque photo qui pourrait être plus succinct et correspondre à une catégorie de photos (une année et un lieu de voyage par exemple). Ainsi la structure globale comprend un identifiant unique, une url unique (correspondant à l'adresse où l'on pourra retrouver l'image qui n'est pas directement stockée en base de données), un nom et une description (le commentaire associé à la photo).

Concernant le stockage des photos, j'ai préféré le faire en dehors de la base de données afin d'épargner ce travail à la base de données et de séparer les différentes fonctions qui pourraient nous être demandées. J'ai choisi de stocker les images via Apache puisque je n'avais pas encore pu expérimenter de solution, elle a paru adaptée et a convenu à sa mission.

Ainsi, lorsqu'un utilisateur importe une image dans l'application, l'image est stockée sous Apache et ses informations associées sont ajoutées en base de données MongoDB. Lorsqu'un utilisateur modifie une image, il peut modifier le nom et la description d'une image : son identifiant et son url ne sont pas modifiables par l'utilisateur. Lorsqu'un utilisateur supprime une image, la ligne associée est supprimée en base de données et elle est aussi supprimée du serveur Apache (ce n'est qu'une copie de l'image initiale).

### ➤ Back-End & Front-End

Concernant le choix des technologies Back et Front, j'ai voulu me perfectionner dans les technologies vues à Télécom Saint-Etienne : Spring Boot et Angular. En effet, ces technologies correspondent bien à la mission qu'on leur donne ici et je me sens maintenant bien plus confiant sur ces technologies.

Les particularités de mon travail se trouvent dans le fait que j'ai voulu pousser l'utilisation des bases de Spring Boot et d'Angular pour réaliser toutes les fonctionnalités. Ainsi, le diaporama synchronisé a été fait simplement à l'aide d'un endpoint de l'API qui présente une image à la fois que chaque client vient interroger pour être notifié du passage à l'image suivante ou précédente par un autre client. Cet endpoint répond à la méthode GET pour l'affichage de l'image mais aussi à des requêtes POST lors de la navigation par un client dans le diaporama (image précédente ou suivante).