

PROJET.NET

Application d'archivage et de consultation de documents



E-TAGGER


ARIOI Vincent - HENSELMANN Pierre - MEKHILEF Cassandra

M1 informatique - La plateforme .NET - 2021/2022



**UNIVERSITÉ
DE LORRAINE**

**UFR MATHÉMATIQUES INFORMATIQUE
MÉCANIQUE ET AUTOMATIQUE**



INTRODUCTION	3
PRÉPARATIONS	3
CHOIX TECHNIQUES ET STRUCTURELS	3
PREMIÈRE MAQUETTE	4
DÉVELOPPEMENT	5
LE PATTERN DAO	5
AMÉLIORATIONS ET RENDU FINAL	6
CONCLUSION	7

INTRODUCTION

Le but de ce projet est de développer pour le client, M Benoît MARTIN, une application d'archivage et de consultation de documents. L'idée de base repose sur le fait d'importer des documents de différents types et de pouvoir les taguer pour les retrouver plus facilement par la suite. Ainsi, l'application devra dans un premier temps permettre de gérer les tags dans une base de données de tags (ajout, suppression et modification). Ensuite, ces tags seront gérés de manière hiérarchique. Enfin, l'utilisateur pourra importer des documents (de type PDF ou JPG), les visualiser et y ajouter des tags.

Notre équipe a décidé de développer cette application sous le nom de E-TAGGER signifiant "taguer en ligne", tout en faisant allusion au mot "étagère", si on le lit à la française, sur laquelle nous allons ranger nos documents.

PRÉPARATIONS

CHOIX TECHNIQUES ET STRUCTURELS

Pour permettre une visualisation des **tags** de façon hiérarchique, nous avons décidé d'associer à chaque tag, un tag père. Les tags qui n'ont pas de père seront alors considérés comme tout en haut de cette hiérarchie. Pour ce faire, un tag sera composé en base de données de son identifiant, son nom, sa couleur (car nous verrons par la suite que nous avons choisi de représenter les tags avec des couleurs) et l'identifiant du tag père. Nous avons ainsi la table suivante :

TAG	
<u>idt_tag</u>	Identifiant
nom	Nom
clr	Couleur
idt_pere	FK TAG (père)

En ce qui concerne les **documents**, nous avons fait le choix de stocker en base de données un chemin relatif vers le fichier d'origine. Ce choix a ses avantages et ses inconvénients. En effet, si le document d'origine est déplacé, le chemin stocké sera alors erroné. Mais, si le document d'origine est modifié, celui qui s'affiche dans l'application le sera également, et cela nous permet de gérer des fichiers beaucoup plus volumineux. Un document est alors uniquement composé de son identifiant et du chemin d'accès absolu:



Ces boutons doivent ainsi permettre l'ajout et la suppression de tags ainsi que la modification de la hiérarchie.

Au centre de l'application nous trouverons la recherche et l'affichage des différents documents uploadés dans l'application.

Enfin à droite, l'utilisateur pourra visualiser le document sélectionné ainsi que les différents tags associés.

Nous avons choisi de représenter les tags avec des couleurs pour rendre la manipulation plus visuelle et pour rappeler l'utilisation des tags sous Macintosh. Ainsi, à la création d'un tag, mais aussi à sa modification, l'utilisateur pourra choisir la couleur associée.

DÉVELOPPEMENT

LE PATTERN DAO

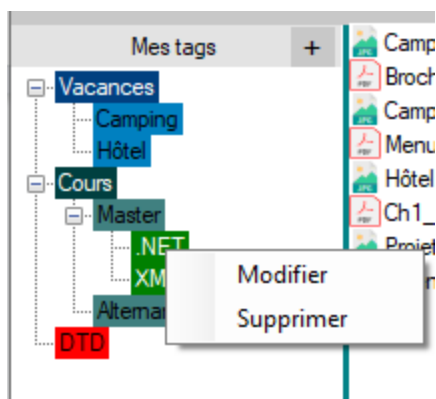
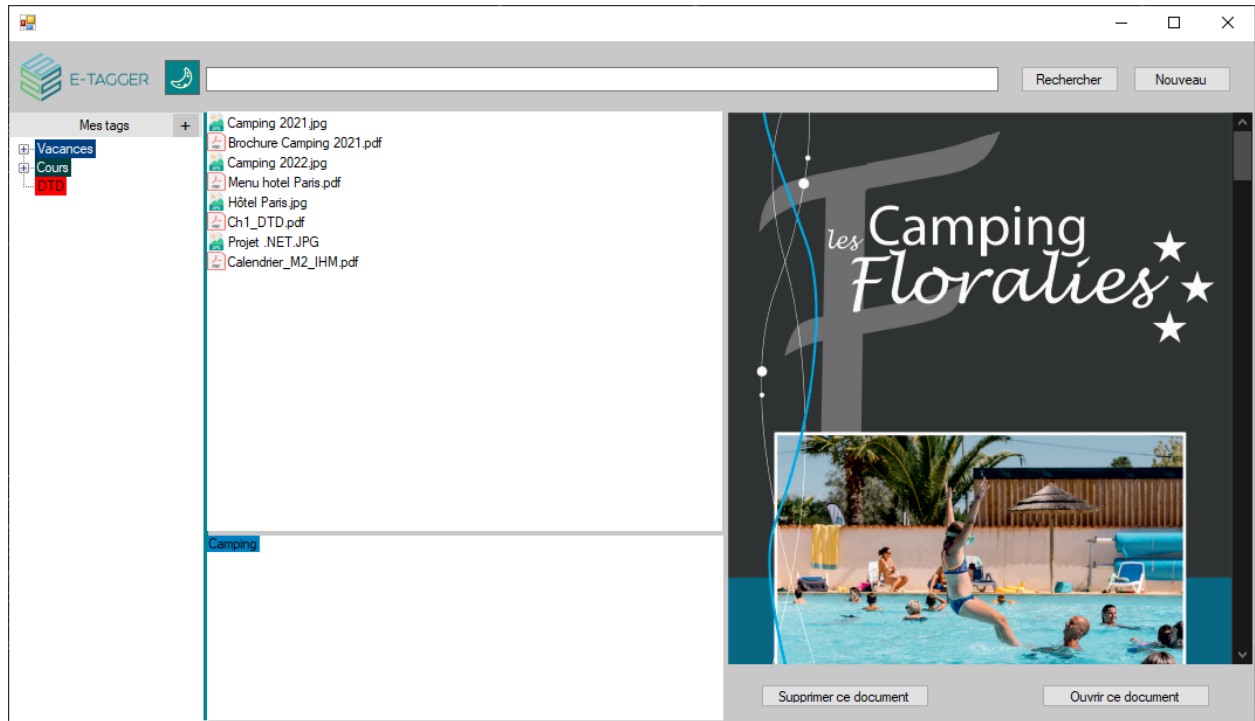
Afin de faciliter au mieux le stockage des données de l'application en base de données, nous avons réalisé un mappage des données qui consiste à faire correspondre les attributs d'une fiche du système de stockage (c'est à dire la base de données) avec les attributs d'un objet affiché sur l'application. Pour ce faire, nous avons donc créé les classes Document, Tag et Liaison composées principalement des attributs de la base de données et des accesseurs.

Le pattern DAO (Data Access Object) permet de faire le lien entre ces classes de mappage et la base de données. Ce pattern est représenté dans notre application par les classes DocumentDAO, TagDAO et LiaisonDAO. Elles contiennent les différentes requêtes INSERT, UPDATE, DELETE et SELECT correspondantes.

Ainsi, l'interaction entre l'interface (et donc le client) et la base de données est nettement facilitée.

AMÉLIORATIONS ET RENDU FINAL

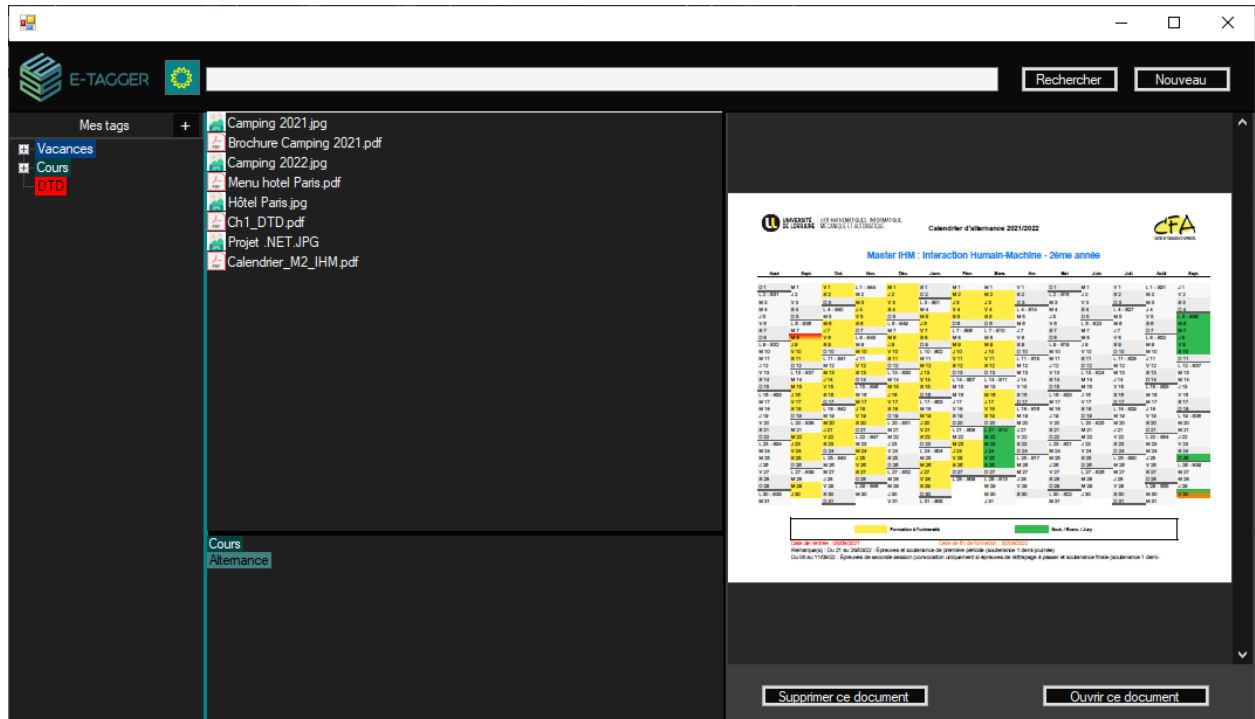
Afin de mettre le plus à l'aise possible le client lors de l'utilisation de notre application, et après quelques tests, nous avons opté pour quelques changements sur l'interface par rapport à la maquette de départ même si globalement, elle s'en rapproche fortement :



Le premier changement majeur concerne l'arborescence des tags. En effet, nous l'avons finalement simplifiée en retirant les différents boutons situés sur la maquette juste au-dessus de l'arbre. Seul le bouton "+", permettant de créer de nouveau tag, a été conservé. La modification et la suppression des tags s'effectue finalement à l'aide d'un simple clic droit sur le tag destinataire de l'action. Ce choix permet à l'utilisateur d'agir directement sur le tag qu'il souhaite modifier ou supprimer et donc de rendre l'action plus naturelle.

Nous avons ensuite largement agrandi la fenêtre principale et plus particulièrement la zone de visualisation des documents. En effet, nous nous sommes rendus compte rapidement que la zone était trop étroite pour accueillir de grandes images. Ainsi, l'agrandissement du visualiseur a entraîné le décalage de l'affichage des tags du document sélectionné, directement sous la liste des différents documents de l'application. De la même manière, il est possible d'en ajouter ou en supprimer à l'aide d'un clic droit sur le tag concerné.

De plus, afin de rendre l'utilisation plus agréable aussi bien de jour comme de nuit, nous avons intégré un "mode sombre", interchangeable grâce à un petit bouton situé à droite de notre logo :



Enfin, l'application permet à l'utilisateur de retrouver facilement ses documents de deux manières différentes :

- par tags : un double clic sur un des tags de l'arborescence de droite affichera les documents tagués par celui-ci dans la zone centrale;
- par nom : en tapant un nom dans la barre de recherche située en haut de l'écran, les documents contenant cette chaîne de caractère s'afficheront dans la zone centrale.

CONCLUSION

Finalement, ce projet aura été l'occasion d'appréhender un nouveau langage de programmation et de prendre conscience d'un certain nombre de difficultés posées par la conception d'interfaces graphiques, notamment du point de vue de l'ergonomie ; il nous aura aussi poussés à faire des choix de conception dans les espaces de liberté laissés par le cahier des charges, par exemple pour la forme de la base de données. Nous avons pu faire l'expérience de la complexité des interactions entre les différents composants d'une application, et de la grande marge de progression entre une application fonctionnelle et une application agréable et facile d'utilisation.