



Université
de Limoges

PROJET DE POO

OUARAFANA / TARCY / TURIKUMWE

JANVIER 2022

Table des matières

1	CONCEPTION	2
1.1	Maquette	2
1.2	Présentation générale	3
1.3	Architecture des classes	4
2	DÉVELOPPEMENT	5
2.1	Picture.java	5
2.2	PictureDaoImpl.java et PictureDaoImpl.java	6
3	BILAN	7
4	SOURCES	8

CONCEPTION

Durant ce projet, nous devons réaliser un outil de gestion de bibliothèque d'images simplifié, implémentant quelques fonctions de sécurité.

1.1 Maquette

En début de projet, nous avons commencé par établir une maquette de notre application.

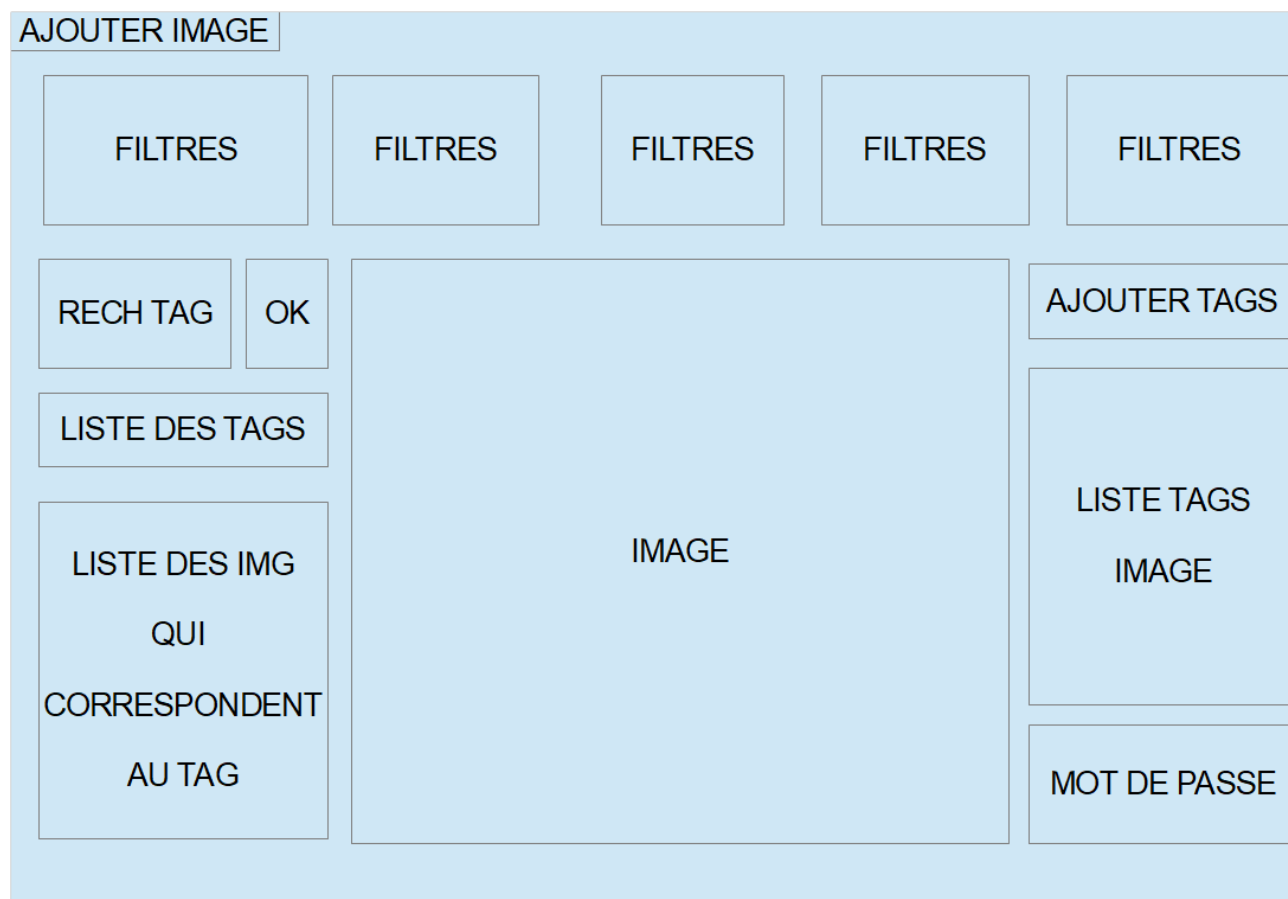


FIGURE 1 – Maquette initiale du projet

Tout au long du projet, nous nous sommes basé sur cette disposition pour mettre en place notre application.

1.2 Présentation générale

Nous avons nommé notre outil "Photogénie". Lors de l'ouverture de l'application, l'utilisateur ne dispose que d'une seule fenêtre pour gérer sa bibliothèque d'image.

Photogénie permet : de prévisualiser le contenu des filtres à appliquer aux images, d'ajouter des tags et images à la bibliothèque, de sauvegarder les modifications apportés aux images et de protéger une image par un mot de passe. L'ajout d'une image et d'un mot de passe se fait par l'intermédiaire de la barre d'outils.

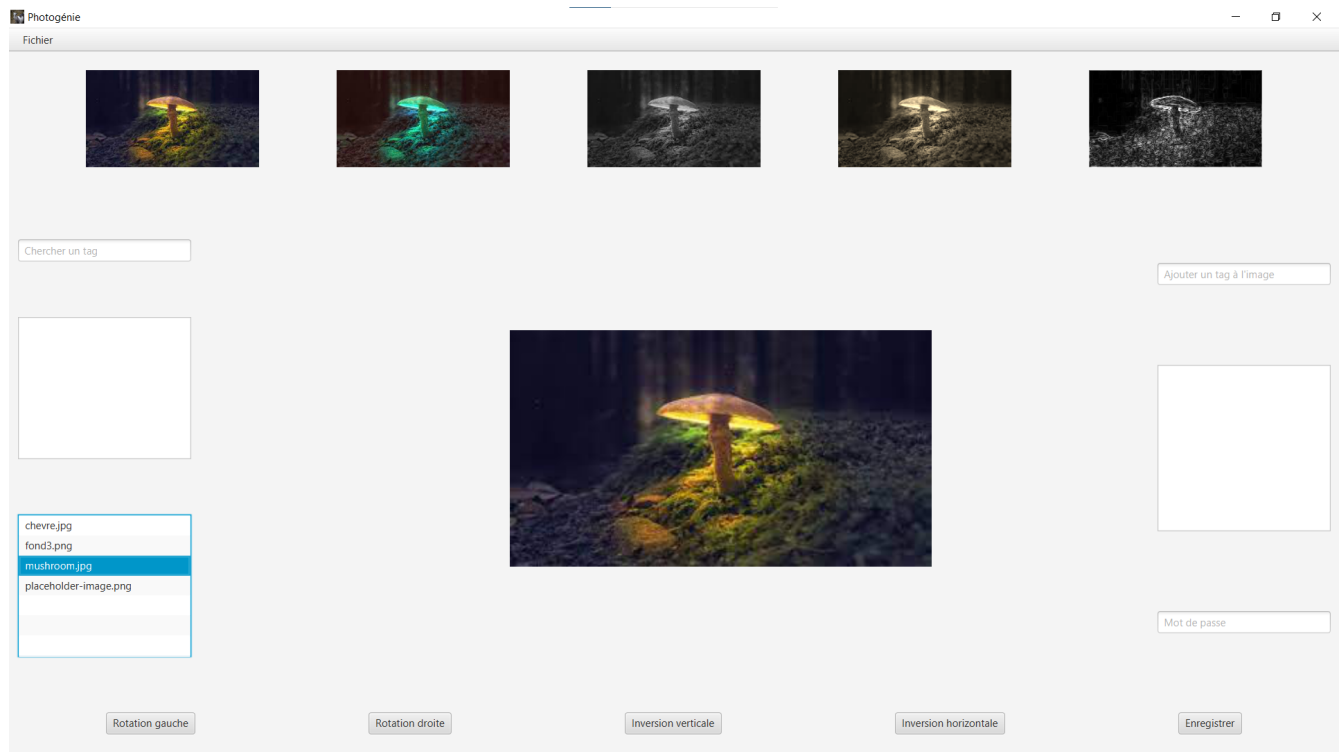


FIGURE 2 – Rendu final de l'application

1.3 Architecture des classes

Pour implémenter nos classes, nous avons tenté de suivre le DAO suivant :

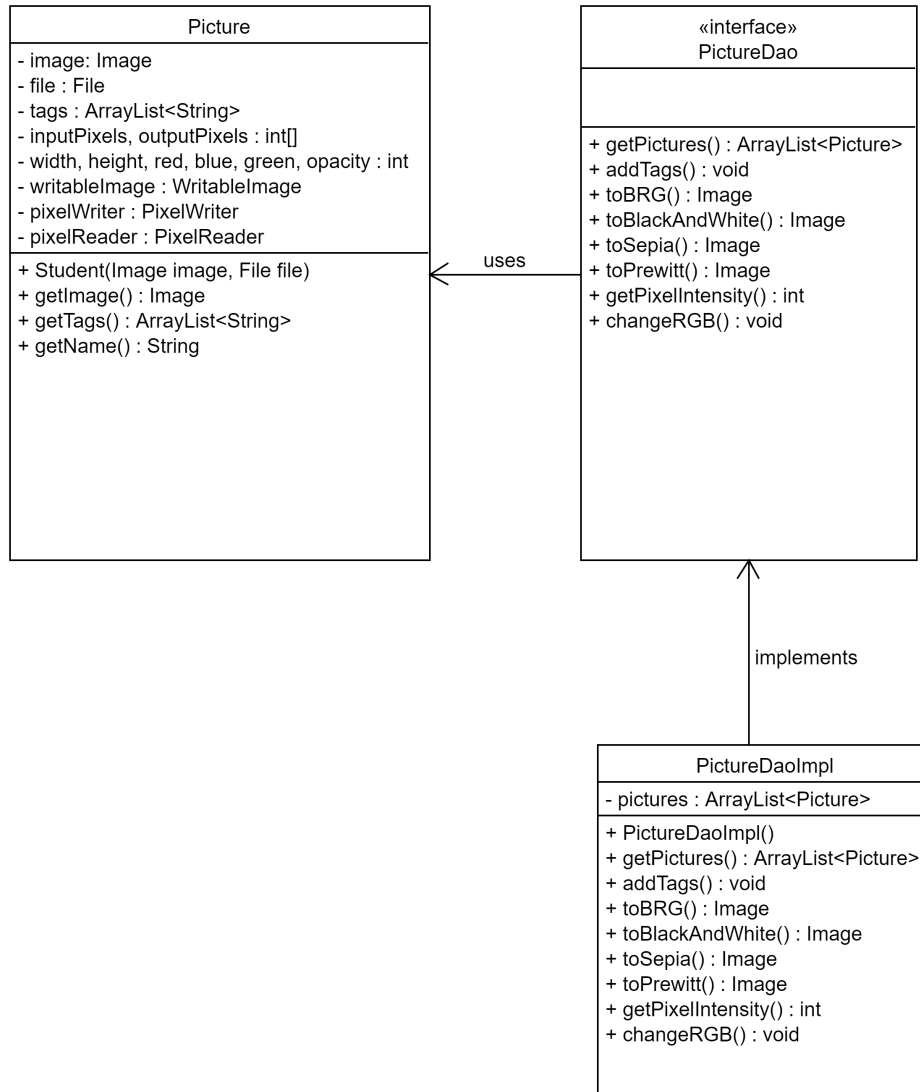


FIGURE 3 – Patron de conception de notre application

Nous avons trouvé ce patron sur le site [tutorialspoint](http://tutorialspoint.com). L'objectif étant d'organiser notre code de façon à simplifier la sauvegarde d'informations.

Notre application est donc composée de ces trois classes, que nous observerons sommairement dans la partie suivante.

DÉVELOPPEMENT

Comme expliqué précédemment, nous gérons les données de l'application au travers des classes `Pictures`. Le lien entre les parties programmatoires et graphiques de notre application s'effectue au niveau du fichier `HelloController.java`. L'application démarre avec la fonction `main` du fichier `HelloApplication.java`.

Regardons maintenant les informations importantes contenues dans ces fichiers.

2.1 `Picture.java`

La classe `Picture` contient l'ensemble des données nécessaires à la gestion des images contenues dans l'application. On y stocke les images ainsi que les tags et les filtres qui leurs sont associés.

Passons les très importants accesseurs/mutateurs, et concentrons nous sur les fonctions principales de la classe.

Nous implémentons deux constructeurs à la classe `Picture`. Le premier, privé ne sert qu'à la sauvegarde de données en XML, et le deuxième, public nous permet d'instancier un objet de la classe en lui conférant une image, un id et un url ;

Pour ne pas trop nous étendre sur les filtres, nous exposerons le fonctionnement de la fonction `toBRG`, qui implémente le filtre `Blue,Red,Green` (faisant passer Mars pour l'Azerbaïdjan). Blague à part, cette fonction, résume l'idée de toutes nos fonctions filtres. Nous récupérons les pixels d'une image (sous forme d'entier de 32 bits) avec la méthode `getPixels` de la classe `PixelReader`, que nous décomposons en quatre variables qui représentent respectivement l'opacité, le rouge, le vert, et le bleu.

Nous appliquons les transformations désirées sur chaque pixel avant de retourner une image transformée. La complexité de la transformation a donc des influences sur le temps de génération de la nouvelle image, notamment pour le filtre de Prewitt.

Pour permettre l'annulation des filtres appliqués, nous conservons une "copie" de l'image d'origine.

Les fonctions de chiffrement et de déchiffrement d'une image reposent sur la classe `SecureRandom`. Nous mélangeons les pixels de l'image de façon prédictive, et les indices de ces derniers.

Lorsque le mot de passe entré est correct, nous trions les indices de l'image mélangée pour retrouver l'image d'origine. (Cette méthode a un certain coût en temps algorithmique).

Enfin, lors de la sauvegarde d'une image, nous ne conservons que les informations essentielles à la transformation de l'image, à savoir, un tableau recensant les modifications appliqués à une image, le chemin de l'image, et les tags appliqués.

2.2 PictureDaoImpl.java et PictureDaoImpl.java

L'interface PictureDao, définit l'ensemble des fonctions que doit contenir la classe PictureDaoImpl.

Cette classe simplifie la sauvegarde et la restauration des images, car elle comporte une liste de Picture, qui sera sauvegardée, et restaurée sous un unique objet.

BILAN

Ce projet était très intéressant, il nous a permis d'appliquer nos connaissances en programmation orientée objets sur un exemple concret. Le travail de groupe ainsi que la courte de période de temps pour mettre en place le tout ont été les principaux défis que nous avons dû relever. Néanmoins, même sans être allé jusqu'à l'implémentation de la base de donnée, nous estimons avoir mené un travail intéressant et sommes satisfait du rendu de notre projet.

SOURCES

Voici les sources utilisées durant notre projet :

- [Application des filtres](#)
- [Algorithme de Fisher-Yates](#)
- [DAO pour le XML](#)
- [Sauvegarde des images](#)
- ainsi que la documentation disponible sur [java](#), [javafx](#), et [xml](#)