

# Rapport du projet bibliothèque

*LEROUGE - MARGUIER*  
*Groupe D - Promotion 2020*



Projet en open-source à l'adresse  
[https://github.com/TimRD/LEROUGE\\_MARGUIER\\_bibliotheque](https://github.com/TimRD/LEROUGE_MARGUIER_bibliotheque)

Le but de ce projet était de proposer une application de tri d'une bibliothèque, que cela soit pour tout type de documents (romans, manuels, revues). Cette application permet d'afficher la liste des différents documents, elle permet aussi d'ajouter un ouvrage particulier ou bien d'en supprimer un. Ce projet nous a permis de nous familiariser, un semestre de plus, en java avec notamment l'utilisation de l'architecture MVC (model, view, controller) mais aussi avec l'interface graphique swing.

Dans un premier temps nous verrons les objectifs de ce projet avant de passer à la réalisation pour enfin conclure sur les difficultés que nous avons rencontrés durant ce projet.

# I. Objectifs

## A. Les attentes du projet

Les attentes de ce projet étaient multiples. Il fallait dans un premier temps s'imprégner des paquetages déjà fournis mais aussi coder différentes classes et actions de notre programme. Nous avons pu aborder dans un premier temps les relations d'héritage grâce au diagramme des classes qui nous a été fournis mais aussi développer des fonctionnalités particulières comme l'ajout et la suppression de certains documents.

Enfin, nous avons eu l'objectif énoncé dans le cahier des charges de réaliser une interface graphique ludique afin de faciliter l'utilisation de notre programme.

## B. L'utilisation de l'architecture MVC

Notre projet ainsi que l'archive qui nous a été fournis utilise le modèle d'architecture MVC pour "model", "view", "controller". Cela nous a permis de séparer distinctement notre travail mais aussi de donner une portabilité à notre projet.

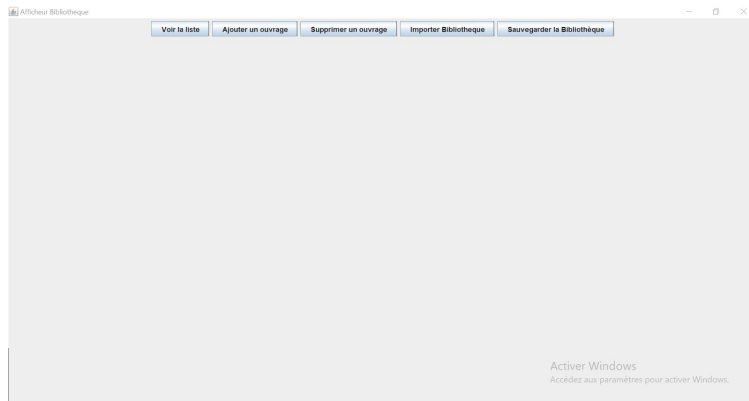
Ces trois parties sont cruciales et ont toutes une utilité particulière. La partie Model contient toutes les classes de notre projet, la partie view gère la partie graphique en enfin, la partie controller fait le pont entre ces deux dernière parties puisque c'est elle qui est appelée au lancement de l'application pour faire le lien entre view en model.

# II. Rendu

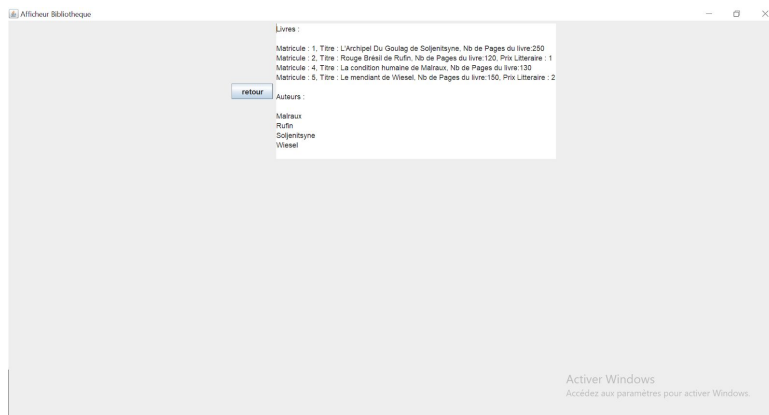
## A. Maquette de l'application

Voici le rendu de notre application sous forme de vues.

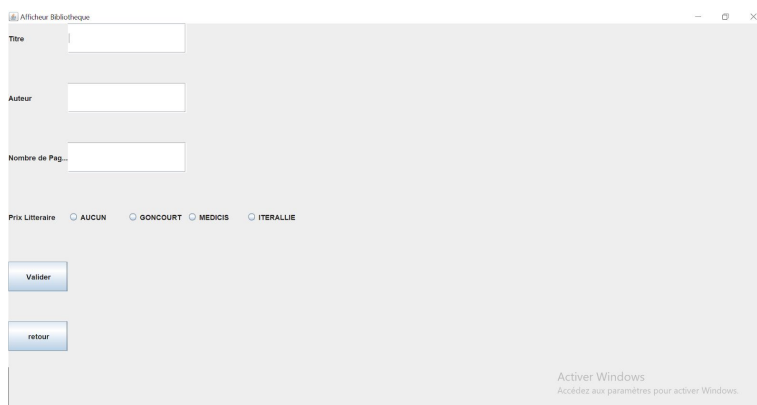
### Page d'accueil avec toutes les fonctionnalités



### Page qui permet d'afficher tous les document enregistrés

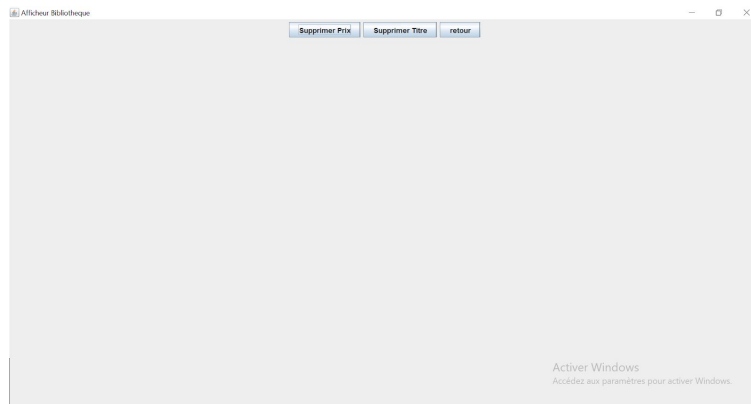


### Page qui permet l'ajout d'un nouvel ouvrage

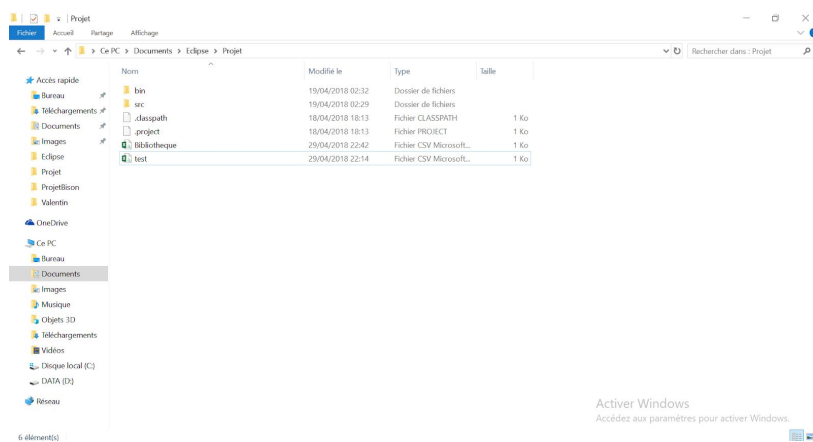


The screenshot shows the same application window with a form to add a new book. The form has fields for "Titre", "Auteur", and "Nombre de Pag...". Below these fields, there are radio buttons for "Prix Littéraire" with options "AUCUN", "GONCOURT", "MEDICIS", and "ITERALLIE". There are "Valider" and "retour" buttons at the bottom. The "Activer Windows" message is still present at the bottom right.

### Page qui permet de supprimer un prix littéraire ou un livre



*A la sauvegarde d'un nouvel ouvrage, celui-ci va être enregistré dans un fichier nommé "Bibliothèque" présent dans le projet.*



## B. Les fonctionnalités

Les fonctionnalités présentes dans le cahier des charges sont toutes présentes, que cela soit :

- L'affichage des documents
- Le tris lexicographique des documents à l'affichage
- L'ajout, la sauvegarde et la suppression des documents et des prix littéraires
- L'utilisation d'une interface graphique

Cependant, pour l'affichage des documents, l'ordre des documents est uniquement lexicographique grâce à l'appel de la fonction sort, ici ligne 25 de *TestDocuments.java*.

```
B.getDocuments().sort(Document.compDoc); // Tri des documents de la bibliotheque
```

### III. Difficultés Rencontrées

Durant ce projet, nous avons rencontrés de nombreuses difficultés notamment à cause d'un niveau très moyen du binôme en java, cependant ce projet nous a permis de progresser grâce à un cahier des charges qui nous guidait tout au long de l'avancée du projet.

Voici certaines difficultés que nous avons rencontrées :

Faute de JDK et de temps, l'implémentation de la javadoc a été compromise, cependant, nous savons comment procéder pour l'implémentation de la JavaDoc pour un futur projet.

La plus grosse difficulté que nous avons rencontré a été l'écriture des fonctions "ecrireFile" et "lireFile", ces fonctions utiles à l'importation de la bibliothèque dans un fichier CSV et de l'ajout de livres à partir d'un fichier CSV sont des notions que nous avons eu du mal à assimiler, car nous ne les avons pas étudiés en cours. La partie SWING nous a d'ailleurs été difficile à implémenter, car il fallait bien manipuler les boutons retour et effacer le panel d'avant.

Enfin, les exceptions ont été sources de nombreuses recherches sur internet afin de bien comprendre leurs fonctionnements et l'implémenter de manière correcte dans notre projet.

Pour finir, la gestion des échéances a été un obstacle, car les séances de TP étaient assez éloignées les unes des autres et nos différentes activités extrascolaires ont fait que nous n'étions pas disponibles la semaine précédente du rendu. Cependant, nous prenons cela comme une leçon pour nos prochains projets afin de planifier en amont des éventuels problèmes de planning.

### Conclusion

Ce projet nous a permis de progresser dans le langage Java mais aussi de savoir gérer le travail en duo avec notamment l'utilisation de git hub pendant ce projet. Le vrai apport de celui-ci est la maîtrise de l'utilisation de l'interface graphique que nous avons développé.

### Bibliographie :

<https://stackoverflow.com/>

<https://openclassrooms.com/dashboard>

<https://java.developpez.com/>

<https://www.mkyong.com/java/how-to-read-and-parse-csv-file-in-java/>

<https://github.com/iSweaZzy/JavaProject/settings/collaboration/>