

# Rapport du projet CY-BOOKS



Amine AIT MOUSSA  
Siham DAANOUNI  
Thomas HAUTION  
Pascal LE  
Amaury PROVENT

**Encadré par :** Eva ANSERMIN  
1ère année de cycle ingénieur **Génie Informatique**  
**Année universitaire :** 2023-2024

## **Sommaire :**

<b>Description du Projet.....</b>	<b>2</b>
<b>Informations Générales.....</b>	<b>2</b>
<b>Organisation de l'Équipe.....</b>	<b>4</b>
<b>Conception et Développement :.....</b>	<b>5</b>
<b>Fonctionnalités Implémentées.....</b>	<b>5</b>
<b>Problèmes Rencontrés et Solutions Apportées.....</b>	<b>7</b>
<b>Résultats.....</b>	<b>7</b>
<b>Conclusion.....</b>	<b>7</b>

# Rapport de Projet : CY Books

---

## Description du Projet

Le projet CyBooks a pour objectif de développer une application de gestion de bibliothèque. Les fonctionnalités principales incluent l'inscription des usagers, la gestion des livres, la gestion du stock, et la gestion des emprunts.

L'application est destinée aux bibliothécaires et doit être utilisable via une interface graphique. Les données des livres sont récupérées via l'API de la Bibliothèque Nationale de France (BNF), tandis que les informations sur les usagers et les emprunts sont stockées localement.

---

## Informations Générales

### Membres de l'équipe :

Amine AIT MOUSSA

Siham DAANOUNI

Thomas HAUTION

Pascal LE

Amaury PROVENT

### **Planification :**

Chaque semaine commence par une planification de nos prochaines étapes. Nous nous réunissons en équipe pour examiner les tâches à venir, établir les priorités et ajuster notre feuille de route si nécessaire. Cette planification hebdomadaire nous permet de rester alignés sur nos objectifs, afin d'optimiser notre efficacité opérationnelle.

Nous avons travaillé en groupe, de 2 à 3 personnes, en faisant des réunions quotidiennes en présentiel ou sur Discord afin de présenter les avancements qui ont été faits et pour des explications inter-groupes pour rester au courant des différentes parties que l'on n'a pas forcément codées.

Dans notre processus de développement, l'utilisation de GitHub est essentielle pour une gestion efficace du code. Nous avons instauré une norme de commits, avec chaque

version dans une branche différente, pour garder des traces des anciennes versions, et pouvoir y revenir si nécessaire. De plus, nous nous efforçons d'effectuer un commit tous les deux jours pour garder une dynamique de travail. Cette pratique régulière nous permet de suivre l'évolution du code, de détecter rapidement les erreurs et d'assurer une collaboration continue et régulière afin d'éviter de prendre du retard sur notre plan d'action, étant donné la courte durée du projet.

De plus, avant de commencer le développement de l'interface graphique, nous nous assurons d'avoir une version en ligne de commande fonctionnelle. Cette approche nous permet de valider les fonctionnalités essentielles de notre application et de résoudre les problèmes de base avant de passer aux aspects visuels. En consolidant d'abord la base fonctionnelle en console, nous garantissons une interface utilisateur cohérente et robuste à l'étape suivante du développement, ce qui nous a permis de gagner du temps et de pouvoir nous focaliser en grande majorité à l'adaptation de notre code.

En combinant une planification hebdomadaire rigoureuse, des réunions presque quotidiennes, une gestion efficace du code avec GitHub et une approche étape par étape du développement, nous maintenons un rythme soutenu et une qualité constante dans la réalisation de notre projet.

#### Versions de l'application :

- Version console :

Pour tester notre modèle de données, nous avons développé une version en ligne de commande en Java. Cette version permet de valider les fonctionnalités de base et de s'assurer que le modèle de données fonctionne correctement, notamment du côté des appels de commandes SQL.

En utilisant cette version simplifiée, nous pouvons identifier et corriger les éventuels problèmes de base, liés au SQL, avant de passer à une interface utilisateur plus complexe, mais plus agréable et intuitive à utiliser.

- Version Finale avec Interface Graphique :

La version finale de notre application intègre une interface graphique développée avec JavaFX, utilisant des fichiers FXML et CSS pour une expérience utilisateur plus optimale. C'est la version qui sera présentée lors de la démonstration finale. L'interface graphique permet une interaction plus intuitive et visuelle avec l'application, rendant l'utilisation plus accessible et agréable pour les bibliothécaires. En combinant ces technologies, nous visons à offrir une application robuste et conviviale, répondant aux attentes et besoins des bibliothécaires.

---

## Organisation de l'Équipe

### Outils utilisés :

- GitHub pour le Stockage du Code :

Notre code source est hébergé et géré via GitHub. Cette plateforme nous offre une visibilité complète sur les modifications apportées au code, facilite la collaboration d'équipe et assure une traçabilité précise des versions. Les fonctionnalités de pull requests et de revues de code garantissent la qualité et la cohérence du travail de chacun.

- IntelliJ IDEA pour le Développement :

Nous avons opté pour IntelliJ IDEA comme environnement de développement principal. Cet IDE riche en fonctionnalités nous permet de coder, déboguer et tester efficacement. La fonctionnalité Code With Me d'IntelliJ IDEA facilite la collaboration en temps réel, même à distance, renforçant ainsi notre productivité, par exemple lorsque nous codons des méthodes sur les mêmes classes, nous permettant de rester cohérent de d'avoir accès aux mêmes informations.

- Discord et WhatsApp pour la Communication Interne :

Nous utilisons Discord et WhatsApp pour maintenir une communication fluide et constante au sein de l'équipe. Discord est privilégié pour les réunions en ligne, les

échanges de groupe et le partage de fichiers, tandis que WhatsApp est utilisé pour les discussions rapides et informelles. Ces plateformes nous permettent de rester connectés, d'échanger des idées et de résoudre rapidement les problèmes, contribuant ainsi à la réussite de notre projet.

### Flux de travail :

Nous avons mis en place plusieurs pratiques de gestion et de développement structurées et efficaces.

Tout d'abord, nous organisons des réunions hebdomadaires pour planifier les tâches. Ces réunions permettent de définir clairement les objectifs à court terme, de discuter des progrès réalisés et d'ajuster notre stratégie si nécessaire. Elles sont essentielles pour maintenir une vision claire et partagée de l'avancement du projet.

Pour assurer la qualité du code, nous pratiquons des revues de code et des tests réguliers. Les revues de code permettent de détecter et de corriger les erreurs, d'améliorer la qualité

du code et de partager les connaissances entre les membres de l'équipe. Les tests réguliers, quant à eux, garantissent que le code fonctionne comme prévu et respecte les normes de qualité établies.

Ces pratiques combinées nous permettent de maintenir une progression régulière dans notre projet, tout en favorisant la collaboration au sein de l'équipe.

### Répartition des tâches :

Pour commencer, la partie console a été priorisée pour s'assurer du bon fonctionnement de l'application et pour se familiariser avec le code.

La recherche dans la BNF via l'API a été codée par Pascal et Thomas, tandis que les autres classes ont été composées par Siham, Amaury et Amine.

La partie JavaFX a été élaborée par Pascal, Amaury, Siham et Amine.

Siham et Amine se sont occupés de la partie CSS, tandis que Thomas s'est assuré du bon fonctionnement de la version console.

Les annexes, les diagrammes de classes et de cas d'utilisations ont été faits par Amine.

La génération de la Javadoc a été faite au fur et à mesure, mais a été révisée par Thomas et Siham.

---

## **Conception et Développement :**

### Architecture du Projet :

- Modulaire pour éviter la concentration de code dans un seul fichier, de faciliter la compréhension du code à sa lecture par une personne qui n'a pas participé à le concevoir, de faciliter le débogage et de respecter les normes du codage afin d'avoir le code le plus propre possible.
- Nous avons fait le choix de laisser 1 fichier CSS par FXML afin de garantir une bonne maîtrise de toute modification du front après utilisation par les bibliothécaires et après des tests du terrain. Le code CSS peut certes être factorisé, mais nous avons choisi de laisser une totale indépendance entre les différentes scènes pour une meilleure flexibilité s'il y a besoin de modifications.

En annexes : le diagramme de Cas d'Utilisation et le diagramme de Classe UML.

---

## Fonctionnalités Implémentées

- Gestion des Usagers :

- Inscription des usagers : Permet aux nouveaux utilisateurs de s'inscrire en fournissant les informations nécessaires à la bibliothécaire, qui les inscrit dans la base de données des clients de la bibliothèque.

- Modification des informations : La bibliothécaire peut mettre à jour les informations personnelles des clients, sauf leur adresse mail qui fait office d'identifiant de recherche dans nos différentes bases de données.

- Recherche d'usagers : La bibliothécaire a la possibilité de lancer la recherche d'utilisateurs spécifiques dans le système, permettant ainsi de gérer toutes les données les concernant.

- Gestion des Livres :

- Recherche multi-critères : La bibliothécaire peut effectuer des recherches selon plusieurs critères (titre, auteur, genre, etc.), en récupérant les données des livres depuis l'API de la Bibliothèque nationale de France.

- Affichage des résultats par page : En cas de grand nombre de résultats, les résultats sont affichés par groupe de 10 au fur et à mesure pour une meilleure lisibilité.

- Connexion à l'API de la BNF : Intégration avec l'API de la Bibliothèque nationale de France pour récupérer des informations détaillées sur les livres.

- Gestion des Emprunts :

- Suivi des emprunts avec historique : Les emprunts sont marqués, et un historique détaillé est conservé pour chaque usager. Ces informations sont disponibles pour la bibliothécaire après avoir recherché le client.

- Limitation du nombre de livres empruntés : Les utilisateurs ne peuvent emprunter qu'un nombre limité de livres simultanément, que l'on peut changer dans le code (initialement fixé à 5), avec une durée maximale d'emprunt définie par la bibliothécaire lors de l'emprunt du livre. Elle choisit, la date et l'heure de rendu du livre concerné.



- **Stockage des Données :**

- Base de données locale : Utilisation d'une base de données locale pour stocker les informations des usagers et des emprunts.

- Lancement automatique du serveur de base de données : Le serveur de base de données se lance automatiquement si nécessaire, assurant la disponibilité des données à tout moment.

Ces fonctionnalités assurent une gestion complète et efficace des usagers, des livres et des emprunts, tout en garantissant une expérience fluide pour la bibliothécaire.

---

## **Problèmes Rencontrés et Solutions Apportées**

Problème : Difficulté de connexion avec l'API de la BNF.

Solution : Utilisation de la bibliothèque `HttpURLConnection` de Java pour gérer les requêtes HTTP et des bibliothèques JSON pour le traitement des réponses.

Problème : Gestion des erreurs et exceptions.

Solution : Implémentation d'une gestion des erreurs robuste pour éviter les interruptions intempestives de l'application. Utilisation de logs pour le suivi des erreurs.

---

## **Résultats**

L'application CY Books répond aux exigences du cahier des charges avec toutes les fonctionnalités principales implémentées et testées.

L'interface graphique est intuitive et permet une gestion efficace des usagers, des livres, et des emprunts par les bibliothécaires. La version finale est stable, après différents tests, elle ne présente pas de défauts majeurs à notre connaissance.

---

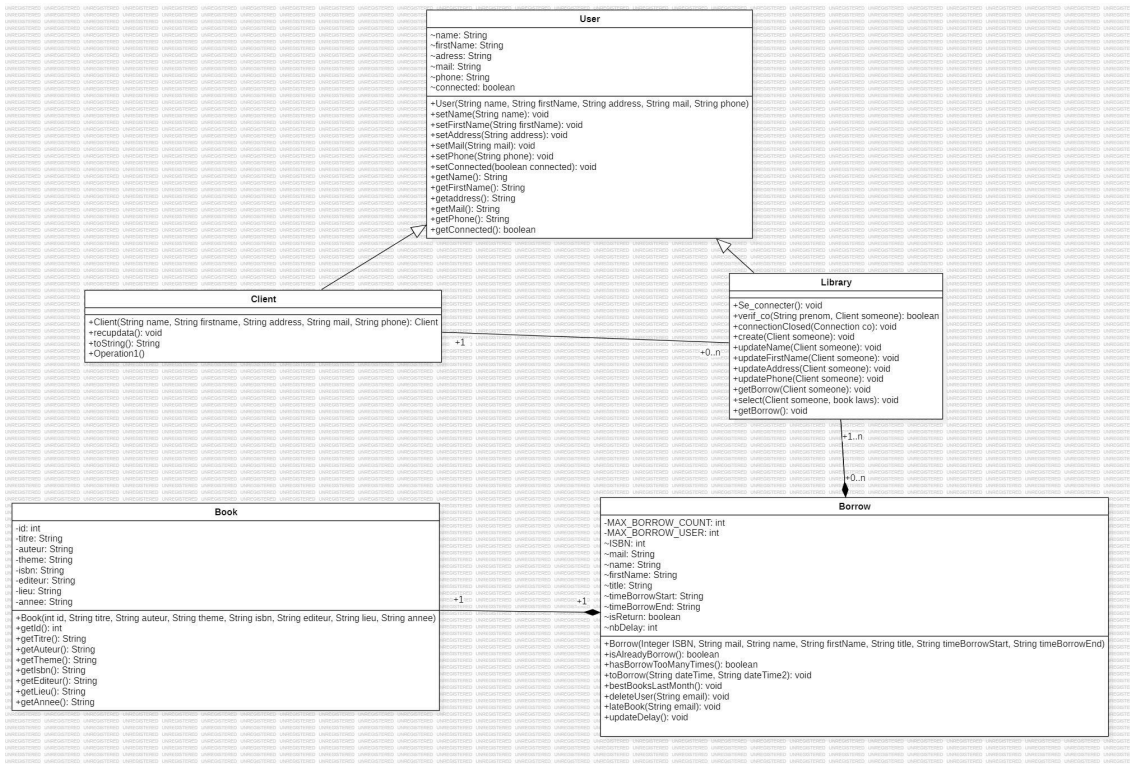
## **Conclusion**

Le projet CYBooks a permis de nous former en JavaFX, de mettre en pratique nos compétences en développement logiciel en équipe, en utilisant des outils de gestion de projet et des méthodologies de développement. Le projet a été un succès grâce à une bonne organisation de l'équipe et une planification rigoureuse.

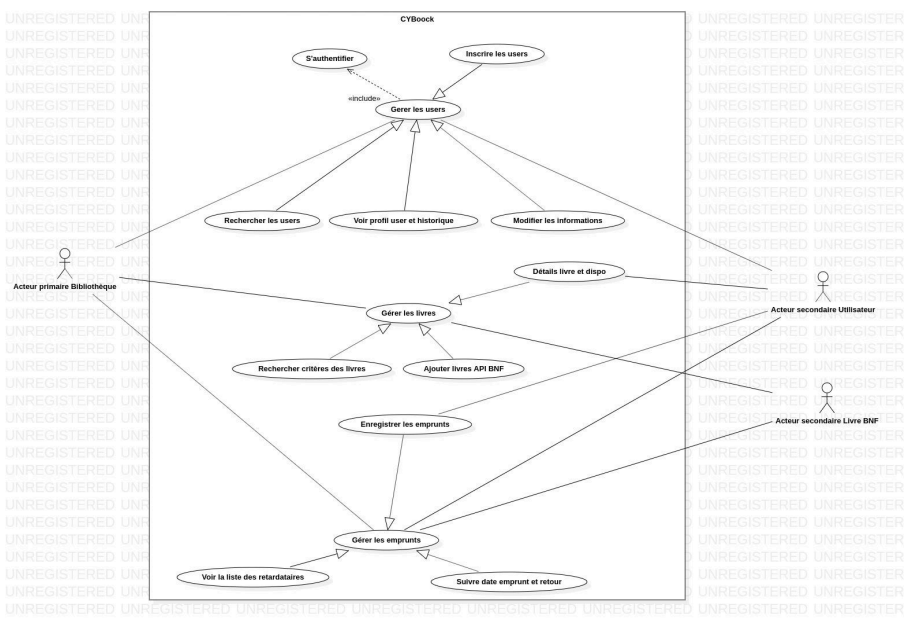


# Annexes

## 1. Diagramme de Classe UML



## 2. Diagrammes de Cas d'Utilisation



Dépôt Git : <https://github.com/SihamDaanouni/CY-Books>