

RAPPORT DE PROJET

CY-Books

Réalisation d'une application permettant la gestion d'une bibliothèque

Manâl BELCAÏD, Yasmina BOUZIANE
Sara GUENDOUZ, Maxime PAVIET,
Anis SOUSSA

26/05/2024

Projet GI 2024 – ING1 GI3

Sommaire

- I. Introduction**
- II. Cahier des charges**
 - Description des besoins fonctionnels
 - Description des besoins non fonctionnels
- III. Conception de l'application**
 - Architecture Générale
 - Diagrammes UML
 - 1. Diagramme de cas d'utilisation
 - 2. Diagramme de classes
 - Base de données
- IV. Développement de l'application**
 - Choix techniques
 - Description des classes
 - Interface utilisateur
- V. Fonctionnalités de l'application**
 - Authentification
 - Gestion des usagers
 - 1. Ajout et modification d'utilisateurs
 - 2. Gestion des emprunts et retours
 - Recherche des livres
 - Lancer le serveur
- VI. Tests et validation**
 - Résultats des tests et validation
 - Problème rencontré : API BNF
- VII. Organisation de groupe**
 - Communication et Collaboration en Temps Réel
 - Gestion du Code et Collaboration sur Git
 - Organisation et Planification avec la Méthode Scrum
- VIII. Conclusion**
 - Bilan du projet
 - Perspectives d'amélioration
 - Retour d'expérience
- IX. Annexes**
 - Structure Générale du code

Introduction

Le principal objectif du projet CY-Books est de créer une application graphique permettant de gérer une bibliothèque, offrant ainsi un outil pratique et efficace pour les bibliothécaires. L'application permet de gérer les usagers, les livres, ainsi que les différents emprunts, avec une interface utilisable à la fois à la souris et au clavier.

Cy-Books est une application Java/JavaFX conçue pour faciliter la gestion quotidienne d'une bibliothèque. Elle permet aux bibliothécaires de parcourir les utilisateurs et les livres, en filtrant les résultats selon différents critères. L'application intègre une base de données locale pour stocker les informations sur les usagers, les identifiants des livres et la liste des emprunts. Les informations détaillées sur les livres (titres, auteurs, année de publication, éditeurs, etc.) sont récupérées depuis un service externe, l'API de la Bibliothèque Nationale de France (BNF).

Cahier des charges

Description des besoins fonctionnels :

Gestion des usagers :

- Inscription des usagers : Permettre l'ajout de nouveaux usagers à la bibliothèque avec des informations de base telles que le nom, prénom, adresse, numéro de téléphone, adresse e-mail et date de naissance.
- Modification des informations des usagers : Permettre la mise à jour des informations des usagers existants.
- Recherche d'usagers : Offrir la possibilité de rechercher des usagers spécifiques à l'aide du mail.

Gestion des livres

- Recherche et filtrage des livres : Offrir la possibilité de rechercher des livres à l'aide de critères multiples tels que le titre, l'auteur et l'isbn

Gestion des emprunts et retours

- Enregistrement des emprunts : Permettre l'enregistrement des livres empruntés par les usagers.

- Enregistrement des retours : Permettre l'enregistrement des retours de livres empruntés.
- Suivi des emprunts : Gérer les délais d'emprunt, les alertes de retards, et les restrictions sur le nombre de livres pouvant être empruntés simultanément par un usager.

Interface utilisateur

- Utilisation intuitive : Interface graphique utilisable à la souris et au clavier.
- Affichage clair des informations : Présentation claire et structurée des informations des usagers et des livres.
- Navigation fluide : Faciliter la navigation entre les différentes sections de l'application (usagers et bibliothèque).

Description des besoins non fonctionnels :

Intégration de l'API de la Bibliothèque Nationale de France (BNF)

- Récupération des données des livres : L'application doit être capable de récupérer les informations des livres depuis l'API de la BNF. Cela inclut des données telles que le titre, l'auteur, l'année de publication, la langue et l'éditeur.

Performance

- Stabilité : L'application doit être stable et ne doit pas se fermer de manière intempestive.

Sécurité

- Authentification : Implémenter un système d'authentification pour restreindre l'accès aux fonctionnalités de gestion de l'application aux seuls bibliothécaires autorisés.

Conception de l'application

Architecture Générale :

Les principaux packages Java utilisés sont :

Package Server: Contient trois packages qui sont : Model, Manager et Data. Les classes représentant les entités principales de l'application telles que Book, User, Borrow et Library sont dans le package Model. Les classes contenant les méthodes permettant de gérer les différentes entités telles que BookManager, UserManager, BorrowManager et LibraryManager sont dans le package Manager. Enfin la classe permettant de communiquer avec l'API de la BNF se nomme API BNF se trouvant dans le package Data.

Package Client : Contient les classes JavaFX ,responsables de l'interface utilisateur, ainsi que la fiche de style.

Diagrammes UML :

Diagramme des cas d'utilisation :

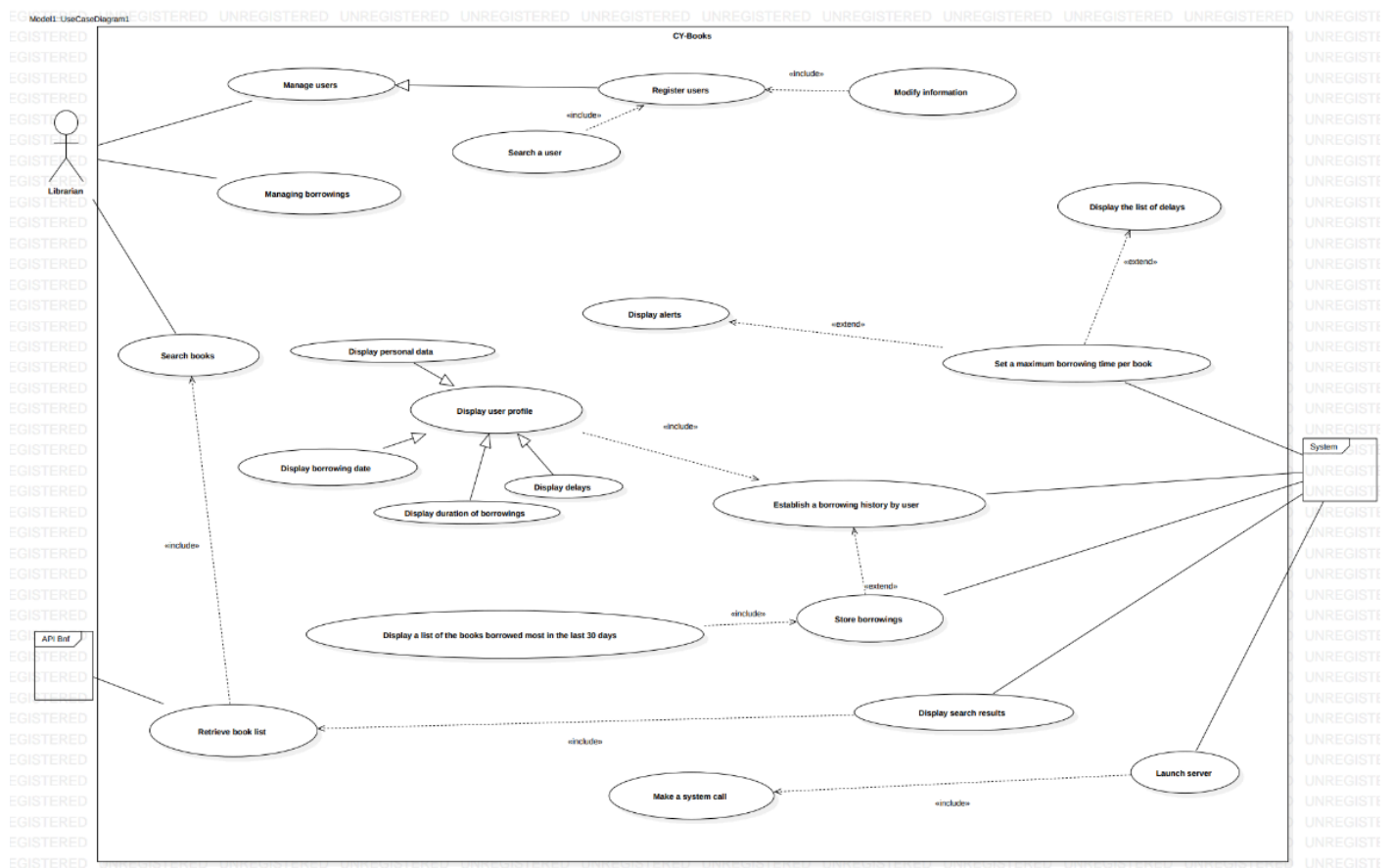
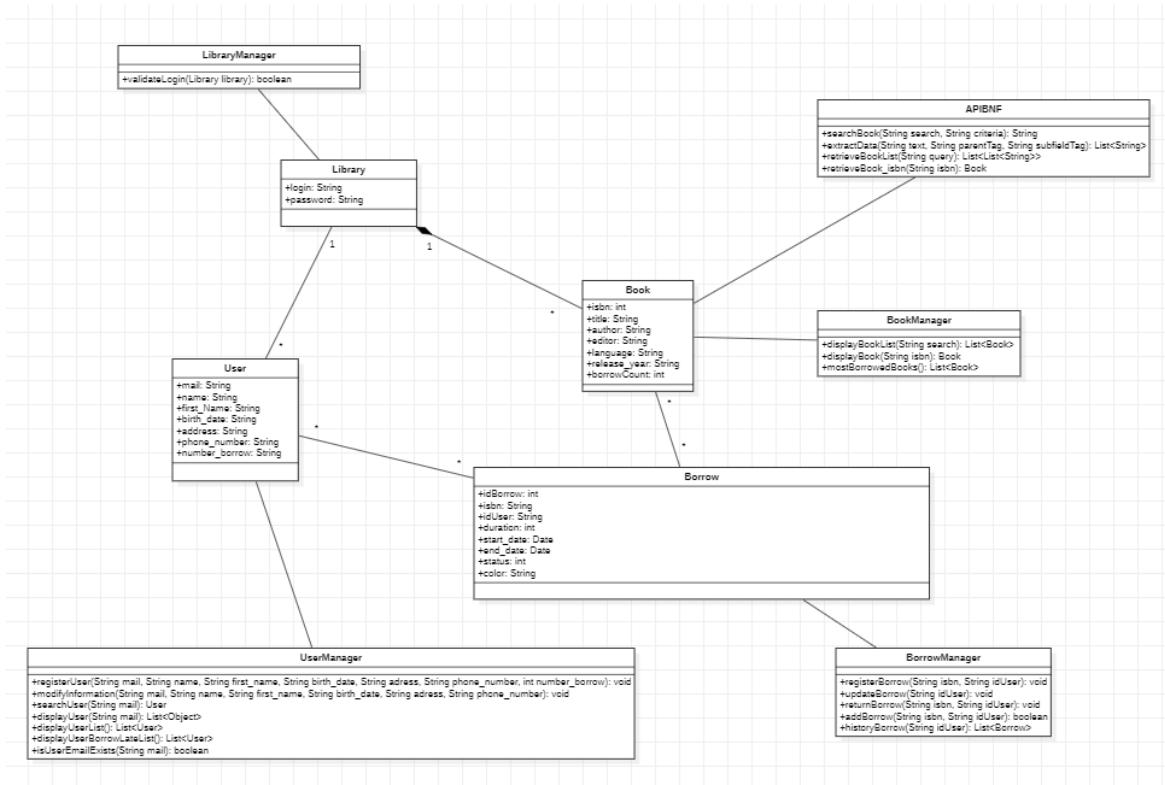


Diagramme de classes :



Base de données :

La base de données est composée de trois tables :

user(mail, name, first_name, birth_date, address, phone_number, number_borrow)

borrow(idBorrow, isbn, idUser, duration, start_date, end_date, status)[status vaudra 0 si l'emprunt est en cours et 1 si l'emprunt est retourné]

library(login, password)

Développement de l'application :

Choix techniques :

Choix de l'outil de développement : IntelliJ IDEA

Nous avons fait le choix d'utiliser IntelliJ IDEA car il s'agit de l'outil de développement avec lequel chacun d'entre nous a appris à coder en Java. Nous étions donc tous déjà familiarisés avec cet outil. De plus, IntelliJ IDEA offre diverses fonctionnalités qui nous apparaissent comme essentielles : une compilation simplifiée, une complétion automatique intelligente et la détection d'erreurs en temps réel.

Choix du serveur et de la base de données : XAMPP/phpMyAdmin

Nous avons choisi d'utiliser XAMPP pour mettre en place notre serveur. Ce choix s'est fait car nous avons tous déjà utilisé XAMPP lors de notre projet en développement web. De plus, nous avons opté pour XAMPP car il permet l'utilisation de phpMyAdmin pour gérer notre base de données MySQL.

Choix de la plateforme logicielle : JavaFX

Nous avons choisi JavaFX pour notre plateforme logicielle. Ce choix s'est fait naturellement car il s'agit de la seule plateforme logicielle que nous avons étudiée en cours de programmation Java, et donc la seule que nous connaissons. De plus, son intégration avec l'écosystème Java et sa richesse de fonctionnalités de l'UI en fait une plateforme adéquate pour notre projet.

Description des classes :

Voir la Javadoc (chemin du fichier en annexe)

Interface utilisateur :

Avant de commencer à implémenter notre application, nous avons réalisé une maquette à partir du logiciel Figma afin de définir un design sur lequel nous nous sommes appuyés.

Voir la maquette (chemin du fichier en annexe)

Fonctionnalités de l'application

Authentification :

Cette page est la page d'authentification. Elle permet au bibliothécaire de se connecter à l'application à l'aide d'un identifiant et d'un mot de passe unique.

CYBOOKS

Identifiant :

Mot de passe :

Mot de passe

Se connecter

Gestion des usagers:

Cette page affiche la liste des usagers inscrits sur l'application. Elle permet également de rechercher un usager en particulier, d'inscrire un usager et d'afficher les retardataires.

CYBook

Usagers

Rechercher un usager

Rechercher

+

Afficher les retardataires

Name	First Name	Email	Birth Date	Address	Phone Number	Borrows
Ash	Kelchum	ash@pokemon.com	1996-02-27	Pallet Town	0678901234	5
Eren	Yeager	eren@attack.com	2009-09-09	Shiganshina	0670123456	1
Goku	Son	goku@dbz.com	1984-12-03	Earth	0679012345	2
Kirby	Dream	kirby@popstar.com	1992-04-27	Dream Land	0678678901	1
Link	Hyrule	link@hyrule.com	1986-02-21	Hyrule	0678345678	2
Luffy	Monkey D.	luffy@onepiece.com	1997-07-22	East Blue	0674567890	1
Luigi	Mario	luigi@nintendo.com	1983-07-14	Mushroom Kingdom	0678234567	2
Mario	Mario	mario@nintendo.com	1981-07-09	Mushroom Kingdom	0678123456	2

<

>

Ces pages permettent l'inscription d'un utilisateur sur l'application s'il n'est pas présent dans la base de données. S'il existe déjà, le second formulaire permet de modifier toutes ses informations hormis le mail car nous l'avons défini comme clé primaire dans notre base de données.

Inscrire un usager

Nom :

Prénom :

Date de naissance :

Mail :

Téléphone :

Adresse :

Ajouter

Modifications des informations

Nom:

Prénom:

Date de naissance:

Mail:

Téléphone:

Adresse:

Valider

Dans le profil de chaque utilisateur on peut retrouver une rubrique “Emprunts” dans laquelle s’affiche tous ses emprunts. En vert, nous retrouvons les livres empruntés et dont la date de retour n’est pas encore dépassée. En rouge, nous retrouvons les livres empruntés et dont la date de retour est dépassée. Et en gris, les livres déjà rendus. En appuyant sur le bouton “Retour Emprunt”, le livre est rendu et passe en gris.

Emprunts

Titre : One Piece Durée : 2 Date d'emprunt : 2024-05-23 Date de retour prévue : 2024-06-22 Retour Emprunt	Titre : La promesse de l'aube Durée : 20 Date d'emprunt : 2024-05-05 Date de retour prévue : 2024-06-04 Retour Emprunt	Titre : Naruto uzumaki !! Durée : 10 Date d'emprunt : 2024-05-15 Date de retour prévue : 2024-06-14 Retour Emprunt
Titre : Les contemplations du pauvre Durée : 27 Date d'emprunt : 2024-04-28 Date de retour prévue : 2024-05-28 Retour Emprunt	Titre : Désir d'exode Durée : 31 Date d'emprunt : 2024-04-24 Date de retour prévue : 2024-05-24 Retour Emprunt	Titre : Harry Potter ha maen ar furien Durée : 15 Date d'emprunt : 2024-01-23 Date de retour prévue : 2024-02-22

En sélectionnant un livre après une recherche, vous pouvez voir toutes ces informations et en appuyant sur le bouton “Emprunter” vous serez redirigé vers la seconde page qui vous permettra l’emprunt après avoir saisi le mail (valide) de l’usager.

One Piece

Titre : One Piece

Auteur : Eiichiro Oda

Edition : Glénat

Date de parution : 2011

Langue : FR

Emprunter

Emprunter

Titre : One Piece

Mail de l'usager:

Valider

Cette page permet de rechercher un livre selon un critère de recherche: il est possible de rechercher un livre par titre, par auteur ou par ISBN. Les résultats de recherche sont affichés par page de 8 livres. Il est évidemment possible de naviguer entre les pages. Il est également possible d’afficher les livres les plus empruntés.

←

BIBLIOTHEQUE

harry potter

⌕

Les plus empruntés

PAR TITRE

PAR TITRE

PAR AUTEUR

PAR ISBN

ISBN	Titre	Auteur	Editeur	Langue	Année
9782363120083	Harry Potter ha maen ar furien	J. K. Rowling	an Amzer embanner	FR	cop. 2012
9782070541270	Harry Potter à l'école des sorciers	J. K. Rowling	Gallimard	FR	1999
9782070543588	Harry Potter et la coupe de feu	J. K. Rowling	Gallimard	FR	2000
9782070615360	Harry Potter et les reliques de la mort	J. K. Rowling	Gallimard	FR	DL 2007
9782070541294	Harry Potter et la chambre des secrets	J. K. Rowling	Gallimard	FR	1999
9782070541300	Harry Potter et le prisonnier d'Azkaban	J. K. Rowling	Gallimard	FR	1999
9782070588268	Harry Potter et la chambre des secrets	J. K. Rowling	Gallimard	FR	DL 2016
9782070624553	Harry Potter et la coupe de feu	J. K. Rowling	Gallimard	FR	DL 2016

1

2

3

4

5

6

7

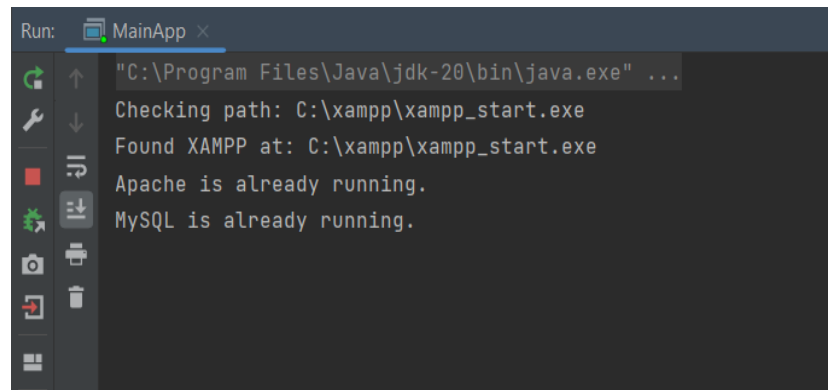
8

9

10

1/17

Nous avons implémenté une méthode nous permettant de lancer le serveur à chaque lancement de l'application, si ce n'est pas déjà fait. Cette méthode a été implémentée dans la classe Serveur du package Java.



```
Run: MainApp x
"C:\Program Files\Java\jdk-20\bin\java.exe" ...
Checking path: C:\xampp\xampp_start.exe
Found XAMPP at: C:\xampp\xampp_start.exe
Apache is already running.
MySQL is already running.
```

Résultats des Tests et Validation

Les tests et la validation de l'application CY-Books ont été réalisés afin d'assurer la conformité aux exigences fonctionnelles et techniques définies dans le cahier des charges. Voici un résumé des résultats obtenus :

Tests Unitaires :

- Modules de gestion des utilisateurs : Les tests unitaires ont couvert l'ajout et la modification des utilisateurs. Tous les tests ont été validés avec succès.
- Modules de gestion des emprunts : Les tests ont porté sur la création, la mise à jour, et la clôture des emprunts. Toutes les fonctionnalités ont été validées avec des résultats positifs.

Tests d'Intégration :

- Connexion à l'API BNF : La récupération des informations des livres via l'API de la Bibliothèque Nationale de France a été testée pour divers critères de recherche. Les résultats ont montré une récupération des données correcte.
- Interaction entre modules : Les tests ont vérifié la cohérence des interactions entre les modules utilisateurs, livres, et emprunts. Aucune incohérence ou erreur n'a été relevée.

Tests Fonctionnels :

- Interface utilisateur : Les tests fonctionnels ont été effectués pour s'assurer que l'interface utilisateur répond aux exigences en termes de navigation, d'accessibilité, et d'ergonomie. Les tests ont confirmé une expérience utilisateur fluide et intuitive.
- Critères de recherche : Les différentes options de filtrage et de recherche des livres ont été testées. Les résultats de recherche sont conformes aux attentes pour ce qui est des filtres par titre et par auteur mais le filtrage par ISBN ne fonctionne pas toujours.

Tests de Performance :

- Temps de réponse de l'API : Les temps de réponse des appels à l'API BNF ont été mesurés et se sont avérés ne pas toujours être dans des limites acceptables, c'est pourquoi nous avons fixé un temps de recherche maximal qui une fois dépassé interrompt la recherche.

Problèmes Rencontrés : API BNF :

Pendant les tests et la validation, plusieurs problèmes liés à l'utilisation de l'API de la Bibliothèque Nationale de France ont été identifiés :

Incohérences de Données :

Certains champs renvoyés par l'API contenaient des données manquantes ou incorrectes.

Solution : Mise en place d'un filtre pour ne récupérer que les livres dont les informations sont complètes.

Lorsque l'on procède à une recherche par titre ou par auteur, on obtient une liste de livres avec chacun un ISBN. Mais lorsque l'on recherche par ISBN avec ceux que l'on a obtenu, il ne retrouve pas toujours le livre. Cela nous empêche donc de récupérer un livre par son ISBN, et donc ses informations, ce qui nous limite pour procéder à des emprunts.

Temps de Réponse Variable :

Les temps de réponse de l'API étaient parfois variables, affectant la performance globale de l'application.

Solution : Mise en place d'un temps de recherche maximal qui une fois dépassé interrompt la recherche.

En conclusion, malgré les défis rencontrés, les tests et la validation de l'application CY-Books ont démontré que l'application est fonctionnelle et répond aux exigences du projet.

Organisation de groupe

Pour mener à bien notre projet, nous avons mis en place une organisation de groupe rigoureuse et collaborative, reposant sur plusieurs outils et méthodes de travail en équipe.

Communication et collaboration en temps réel :

Afin de faciliter la communication et le travail en temps réel, nous avons choisi d'utiliser Discord. Ce choix s'est avéré particulièrement pertinent car il nous a permis de nous connecter tous ensemble pour des séances de travail synchronisées. Nous avons tiré profit de la fonctionnalité de partage d'écran et des appels vocaux pour organiser des sessions "codes with me" sur IntelliJ. Ces sessions ont permis à tous les membres de participer activement au développement du code, d'échanger des idées instantanément et de résoudre les problèmes rencontrés en temps réel.

Gestion du Code et Collaboration sur Git :

Pour éviter les problèmes liés aux opérations de pull et push sur Git, nous avons adopté une approche collaborative pendant nos sessions de codage sur IntelliJ. Grâce à cette méthode, nous avons pu travailler sur le même fichier de code en temps réel, minimisant ainsi les conflits et les erreurs de fusion. Cette collaboration en direct a également amélioré notre productivité et notre efficacité en réduisant les interruptions liées à la gestion des versions.

Organisation et Planification avec la Méthode Scrum :

Pour structurer notre travail et assurer une progression régulière, nous avons utilisé la méthode Scrum. Cette méthodologie agile nous a permis de diviser le projet en sprints, chacun avec des objectifs et des échéances clairs. Chaque membre de l'équipe avait des tâches spécifiques à accomplir, définies lors des réunions de planification des sprints. Cette organisation nous a permis de suivre l'avancement du projet de manière précise, de respecter les deadlines et de maintenir un haut niveau de productivité.

En utilisant Scrum, nous avons pu :

- Définir des objectifs clairs pour chaque sprint, ce qui a facilité la gestion des priorités et la distribution des tâches.

- Tenir des réunions quotidiennes (daily meetings) pour discuter des progrès, des obstacles rencontrés et des prochaines étapes.
- Maintenir une transparence totale au sein de l'équipe, chacun sachant exactement ce qu'il devait faire et ce que les autres membres faisaient.

Notre organisation de groupe a été un élément clé de la réussite de notre projet. L'utilisation de Discord pour la communication en temps réel, les sessions collaboratives sur IntelliJ pour le développement du code, et l'application de la méthode Scrum pour la planification et la gestion des tâches ont permis à notre équipe de travailler de manière efficace et cohérente. Cette approche intégrée nous a aidés à éviter les problèmes de coordination, à respecter les délais et à atteindre les objectifs fixés.

Conclusion

Bilan du projet :

Le développement de l'application de gestion de livres et d'utilisateurs a donné des résultats positifs. Cette application a été conçue pour optimiser la gestion des utilisateurs d'une bibliothèque et leurs emprunts, avec un accent particulier sur la simplicité d'utilisation et l'efficacité des opérations quotidiennes. Grâce à une interface conviviale, la bibliothécaire peut désormais gérer les informations des utilisateurs et des livres de manière efficace. L'intégration avec l'API de la Bibliothèque nationale de France (BnF) a permis de récupérer automatiquement les données bibliographiques, ce qui a considérablement réduit le temps nécessaire pour emprunter une pluralité de livres et a amélioré la précision des informations.

Perspectives d'amélioration :

Bien que l'application remplisse efficacement ses fonctions actuelles, plusieurs améliorations peuvent encore être envisagées pour la rendre encore plus performante et utile. Par exemple, l'exploration de l'API de la BnF a révélé certaines limites, notamment l'incapacité à intégrer des filtres tels que le genre, le type ou la langue des livres.

Retour d'expérience :

Le retour d'expérience sur ce projet a été positif. La collaboration efficace entre les membres du groupe a été déterminante dans la réalisation d'une application fonctionnelle et intuitive. Grâce à des tests utilisateurs approfondis, nous avons pu garantir la fluidité de la recherche, ainsi que la simplicité et la fiabilité de l'enregistrement des données. Cela a conduit à une interface client sans faille, exempte de bugs.

Annexes :

Structure du git

Java :

Client (Interface utilisateur/Javafx) : cy-books14/src/Java/Client

Server (code java) : cy-books14/src/Java/Server

Ressources :

Javadoc : cy-books14/src/Ressources/Javadoc/allpackages-index.html

Maquette : cy-books14/src/Ressources/Maquette CY-Books.pdf

Répertoire balise API :cy-books14/src/Ressources/Maquette CY-Books.pdf

UseCase et diagramme de classe : cy-books14/src/Ressources/UML.pdf

SQL :

Création table de données : cy-books14/src/SQL/bibli.sql

Intégration des données : cy-books14/src/SQL/bibli_data.sql

Choix techniques : cy-books14/src/README.md