RAPPORT
Architecture logicielle

2023

MARHERAOUI Amin | 205 & IBOUDA Yasser | 206

Sommaire

Introduction

Rapport

Contexte

Au sein de ce projet, nous devons développer une application permettant de gérer les interactions d'une bibliothèque avec ses abonnés. Cette application a plusieurs services : Un service de réservation pour que l'abonné puisse mettre un document de côté le temps de venir le chercher ; Un service d'emprunt afin de pouvoir récupérer un livre réservé ou libre et un service de retour afin de gérer les retours des documents.

Les différents types de client pouvant interagir avec le logiciel sont : Les postes de la bibliothèque à partir desquelles nous pouvons emprunter ou retourner un livre et à partir des ordinateurs personnels des adhérents, nous pouvons uniquement réserver un livre.

Notre équipe

- IBOUDA Yasser
- MARHERAOUI BERRAHHOU Amin

Nos outils

Icon clickable

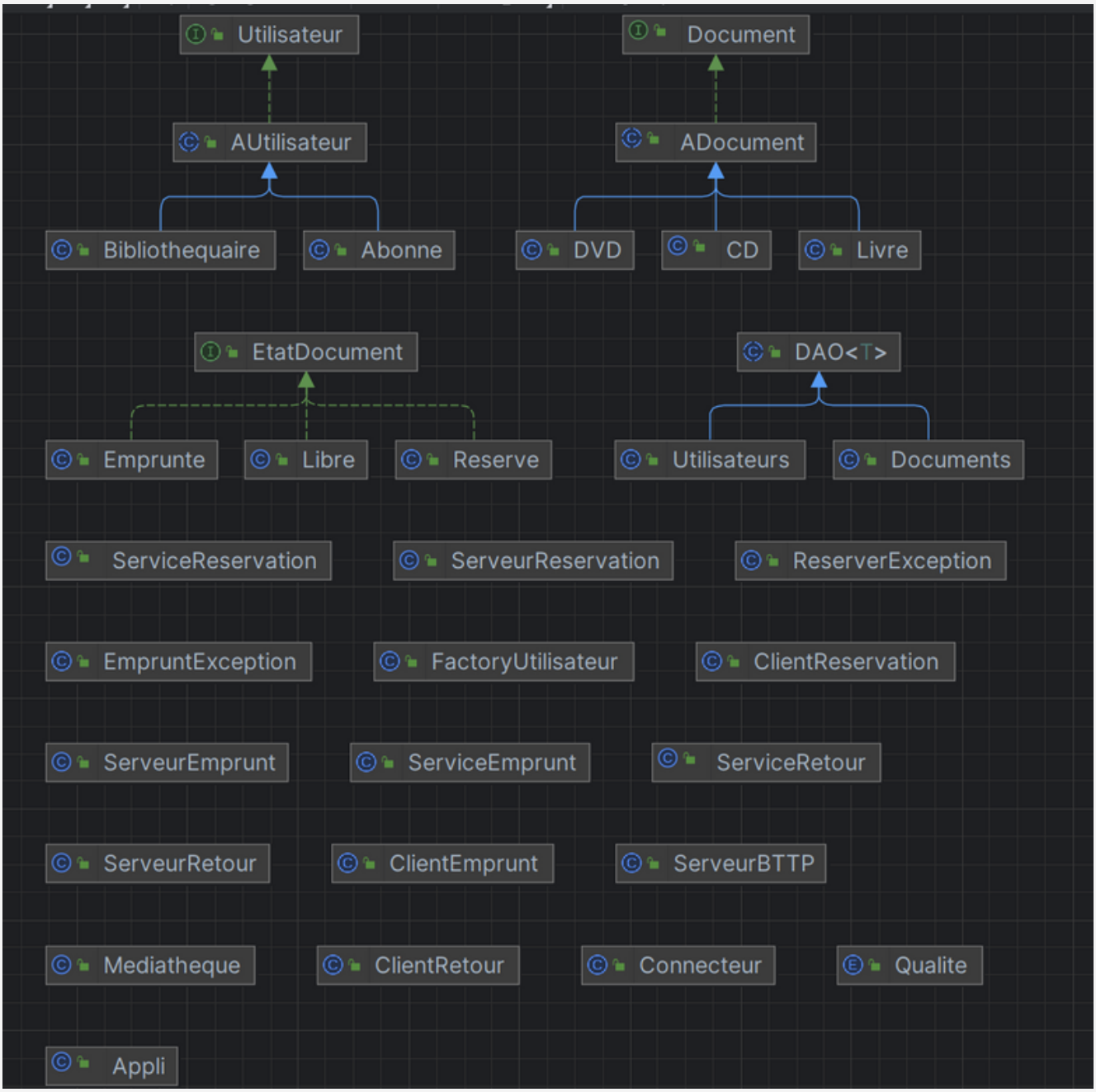
Ressources partagées

Les différentes ressources partagées sont :

Normalement la liste d'abonnés et la liste de documents : Afin de les protéger, nous avons essayé d'utiliser la méthode statique appelée synchronizedList() de la classe Collections qui permet de construire une liste Thread-Safe depuis une factory. Pour que les différents Threads puissent accéder chacun à leur tour à ces listes, cependant nous n'avons pas vraiment réussi car nous avons eu des problèmes au niveau des méthodes static.

Les instances de ADocument : Certaines fonctions de ADocument comme reserver() ont besoin d'être Thread-Safe. Par exemple, un document ne peut pas être réservé par deux personnes en même temps. Si deux instances d'Abonné font cette requête sur un même document, alors ces 2 requêtes doivent être traité l'une après l'autre afin de soulever une exception lors de la 2ème demande de type ReserveException. C'est pourquoi nous avons synchronized des méthodes et utilisé des blocs de code synchronized (this){ ... } pour verrouiller l'accès concurrent au document courant grâce à son moniteur de Hoare. Nous avons essayé de réduire au maximum la taille de ces blocs.

Diagramme de package



Conclusion

Pendant ce projet, nous avons découvert la programmation client-serveur. Nous avons ainsi pu expérimenter l'utilisation de nouveaux composants comme les threads et les sockets et de nouveaux concepts comme la communication synchrone et la thread safety.

Afin d'améliorer notre productivité et notre confort de travail collaboratif, nous avons utilisé le système de gestion de version Git et discord.

Le sujet, à la fois ludique et intéressant, nous a permis de ressentir l'esprit du développeur. De plus, la création d'une application client-serveur était pour nous une expérience nouvelle, tout à fait enrichissante.

Enfin, l'aspect collaboratif de ce projet fait partie de ses points forts. En effet, la motivation mutuelle est un moteur puissant qui permet de surmonter tout type de difficulté.
De plus, la résonnance de plusieurs esprits focalisés sur le même objectif, communiquant sur un problème, est une des formes les plus efficaces de réflexion.
Chacun comblant les faiblesses des autres, nous avons pu nous tirer mutuellement vers le haut. Ayant déjà l'expérience du travail en commun, nous avons renforcé notre niveau de collaboration et cela nous a permis d'avancer encore plus, chacun connaissant les spécificités des autres.

Remerciements

Nous vous remercions pour la lecture de notre travail et nous espérons avoir répondu à toutes vos questions.

Amin Mahreraoui
Yasser Ibouda

Coordonnées

AMIN MARHERAOUI : amin.marheraoui-berrahhou@etu.u-paris.fr
YASSER IBOUDA: yasser.ibouda@etu.u-paris.fr