

Projet Bibliothèque

Lors de ce projet, nous avons eu à créer une application Java pour la gestion d'une bibliothèque.

Cette application devait comprendre :

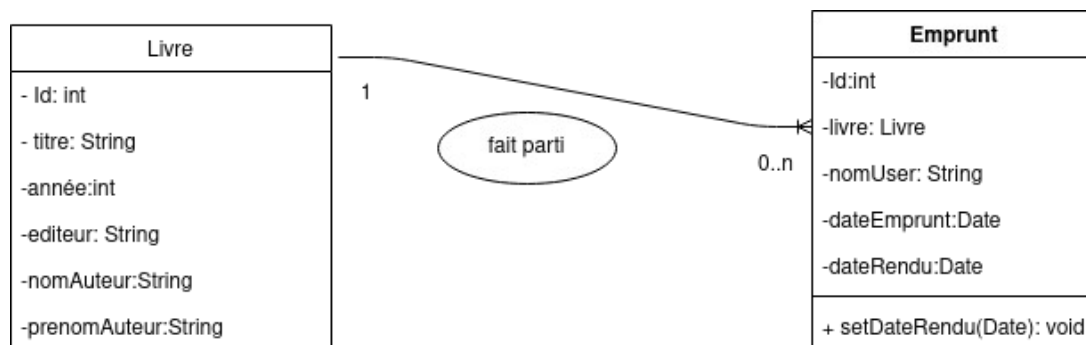
- une première page permettant d'enregistrer les livres de la bibliothèque ;
- une deuxième page permettant la gestion des emprunts (affichage de la liste des emprunts en cours, possibilité de créer un nouvel emprunt et de renseigner la date de retour du livre emprunté lors de son retour) ;
- une base de données pour permettre la persistance des entités créées ;
- la communication entre la base et les pages html en suivant le modèle type MVC.

Nous avons décidé de travailler sur l'IDE NetBeans v12.3, et avec un framework SpringBoot liée à une base SQLite par JDBC + Hibernate. Nos Javascript utilisent la bibliothèque JQuery.

Les auteurs de l'application sont : Claire De Giovanni, Jules Flores, Toutane Ndiaye.

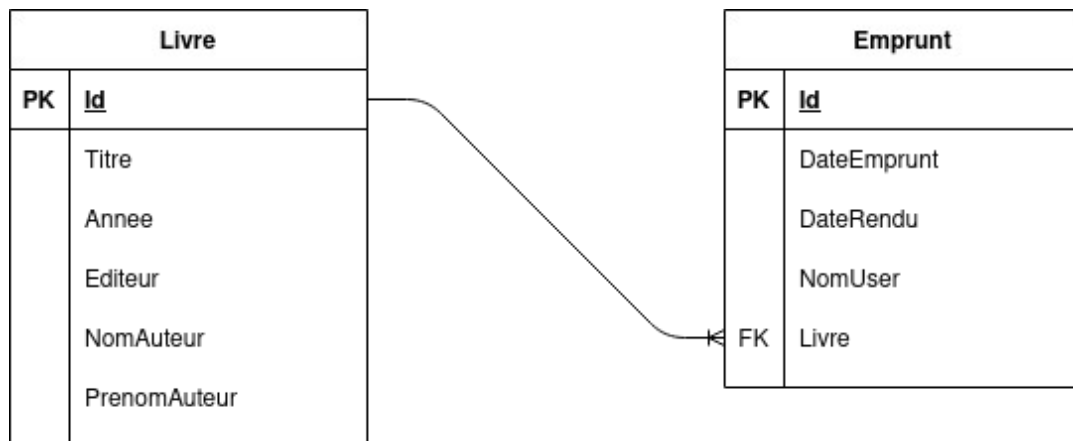
Nous avons développé cette application en télétravail. Nous nous sommes appuyés sur plusieurs outils informatiques : Mumble (audio), Discord (partage d'écran), Drawio (dessin des diagrammes de classe et de la base), LibreOffice Impress (présentation).

Diagramme de classes



Il est à noter que nous avons créé une classe intermédiaire LightEmprunt pour la communication entre le JavaScript et le Contrôleur.

Diagramme de la base



L'attribut Livre dans un Emprunt est annoté @ManyToOne.

Planning de réalisation

Voir en annexe le fichier intitulé « Planning.ods ».

Maquette de la page 1

Gestion de la bibliothèque

Ajouter un nouveau livre à la base de données

Titre :

Année:

Editeur :

Nom Auteur:

Prénom de l'auteur:

[Ajouter un nouveau livre à la base de données :](#)

Cette page permet de créer un objet livre et de le persister dans la base de données. L'utilisateur insère les informations demandées et valide la saisie en utilisant le bouton « Enregistrer ».

Une ligne de texte apparaît alors en dessous des boutons pour informer l'utilisateur du résultat (si échec, la cause de l'échec comme un champ non rempli ou un problème de connexion au backend, sinon pour confirmer que l'action s'est bien effectuée).

Le bouton « Créer un emprunt » mène sur la deuxième page.

Maquette de la page 2

Gestion des emprunts
Ajouter un livre

Création d'un emprunt

Titre du livre :
Id :
Nom de l'emprunteur :
Date d'emprunt :

Enregistrer

Id	Titre	Auteur	Année	Disponible
1	Titre1	Jules	2021	Oui
2	Titre1	Jules	2021	Non

Id de l'emprunt	Nom utilisateur	Date d'emprunt
1	Toutane	15/06/2021
2	Claire	30/06/2021

Ajout de la date de retour d'un livre

Id de l'emprunt :
Date de rendu :

Enregistrer

Cette page comporte plusieurs parties :

Création des emprunts :

Sur cette page, dans le premier cadre, l'utilisateur peut générer un nouvel emprunt.

Le champ « Titre du livre » permet, lors de la saisie, l'affichage en direct de la liste des livres correspondants (fonction de recherche simple). Cette liste indique la disponibilité de chaque livre et permet de renseigner l'Id du livre qui va être emprunté.

Cette méthode permet de sélectionner le bon livre lorsque plusieurs ont le même titre (plusieurs exemplaires du même livre ou homonymie) et voir s'ils sont disponibles ou non.

Le bouton « Enregistrer » permet de valider la saisie. Une ligne de texte apparaît alors en dessous des boutons pour informer l'utilisateur du résultat (si échec, la cause de l'échec comme un champ mal rempli, livre non disponible ou un problème de connexion au backend, sinon pour confirmer que l'action s'est bien effectuée).

Liste des emprunts :

Dans la partie intermédiaire, on affiche la liste de tous les emprunts en cours. Lorsqu'un livre est emprunté (bouton « Enregistrer » du haut), cet emprunt s'ajoute automatiquement à cette liste.

Ajout de la date de retour d'un livre :

Lors du retour d'un livre par un emprunteur, l'utilisateur retrouve l'id de l'emprunt grâce à la liste au-dessus et le renseigne dans le champ correspondant ainsi que la date de rendu.

Le bouton « Enregistrer » permet de valider la saisie. Une ligne de texte apparaît alors en dessous pour informer l'utilisateur du résultat (si échec, la cause de l'échec comme un champ mal rempli, date non valide ou un problème de connexion au backend, sinon pour confirmer que l'action s'est bien effectuée).

La mise à jour de la base de données comprend la modification de l'objet emprunt et l'attribut disponible du livre correspondant. La liste des emprunts est aussi mise à jour.

Un bouton de retour à la page 1 est aussi présent.

Gestion de projet : utilisation de Git en lien avec GitHub

Afin de faciliter la mise en commun du code de notre application et de la documentation du projet, nous avons utilisé GitHub.

Pour une question de facilité, nous n'avons pas créé de branche additionnelle à la branche main. Nous devons alors être vigilant à ne pas travailler simultanément sur les mêmes fonctionnalités afin de ne pas créer de conflit. Une autre restriction imposée était de partager uniquement des codes fonctionnels. Cette règle nous permettait de garder sur le dépôt distant une version toujours fonctionnelle de l'application et ainsi d'assurer que tous les membres du groupe puissent continuer de travailler simultanément et de tester l'application.

Au démarrage de chaque nouvelle journée de travail, nous vérifiions que notre dépôt local était à jour avec le dépôt distant à l'aide de la commande `git status`. A chaque nouvelle fonctionnalité implantée avec succès, nous partagions cette dernière sur le dépôt distant.

Tableau de tests d'acceptation

Voici un tableau recensant tous les cas possibles d'utilisation de notre application afin de vérifier leurs bons fonctionnements.

Voir en annexe le fichier intitulé « Les_tests.ods ».