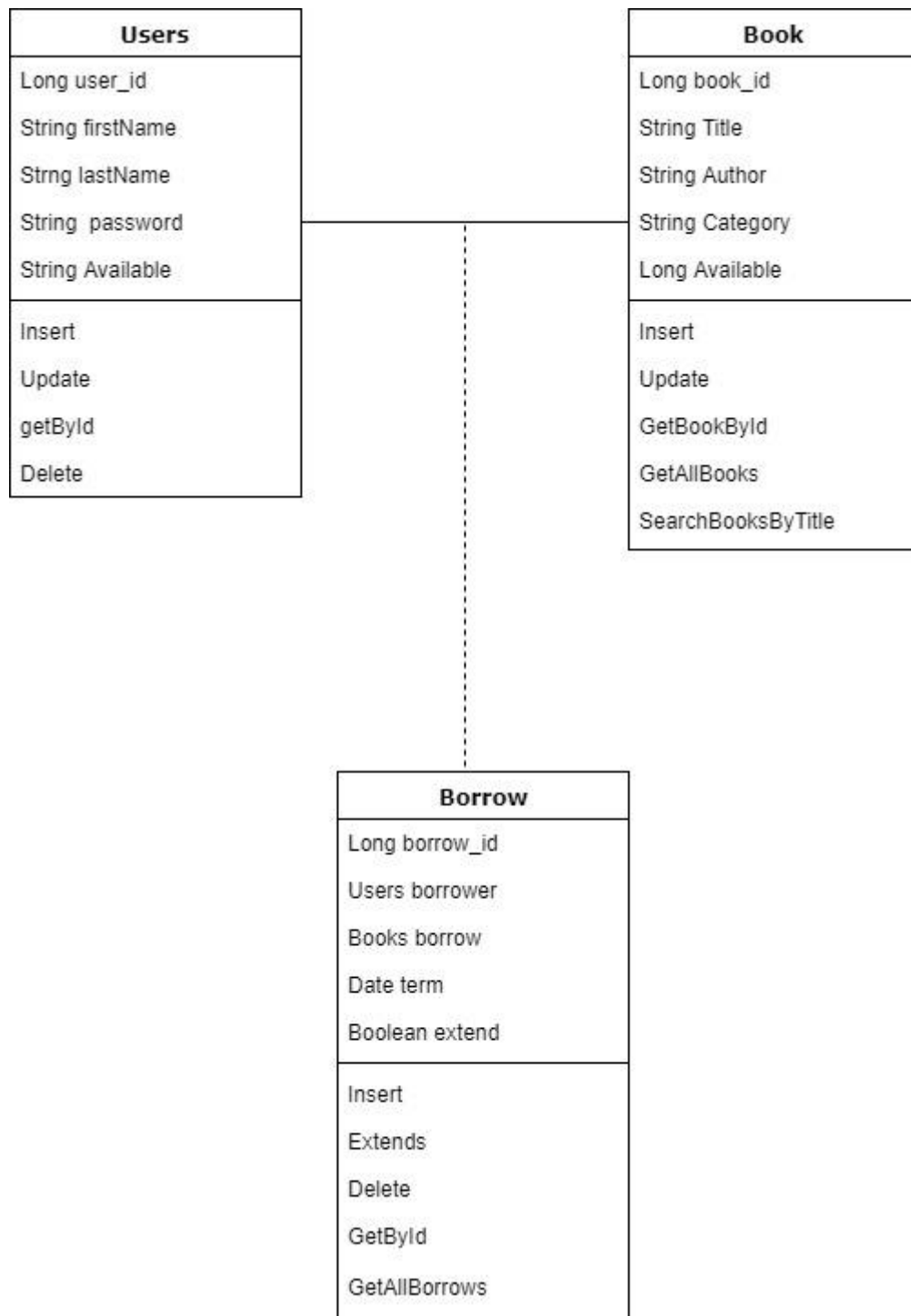


# PROJET 4

## SYSTEME D'INFORMATION BIBLIOTHEQUE

Le système d'information de la bibliothèque repose sur une architecture orienté service, il est dans cette première version constitué d'un web service SOAP, d'une application web et d'un batch. Le système de gestion de la base de données est PostgreSQL.

Le diagramme de classe de l'application :



Le Webservice connectée à la base de données PostgreSQL permet à la fois de gérer les persistances en BDD et produit les services consommables par les autres API. Le web service est développé avec le framework SpringBoot.

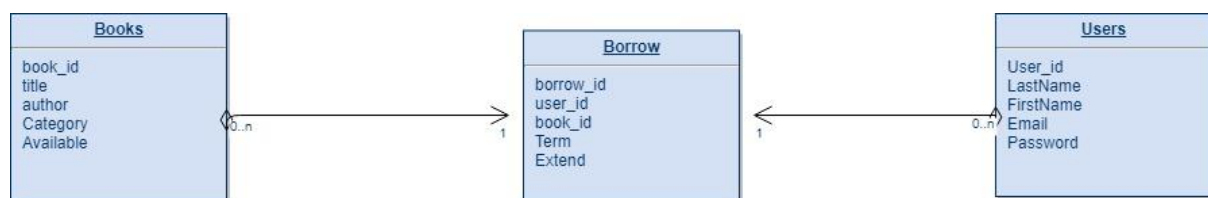
L'application suit un des modèles d'architecture les plus répandus, le modèle MVC pour Model View Controller. L'application est divisée en trois parties :

- La partie vue, qui représente ce que l'utilisateur va voir et avec laquelle il pourra interagir. Ici la vue est minime et dirige vers les wsdl.
- La partie contrôleur, qui fait le lien entre la vue et le reste de l'application. Cette partie contrôle les entrées en les orientant comme un aiguillage et contrôle aussi les sorties vers la vue
- La partie modèle, qui est la partie métier de l'application possédant toutes les fonctions de l'API.

L'architecture MVC est matérialisée par une structure multi-tiers, l'application est divisé en plusieurs modules gérant chacun une logique métier différente :

- Le module Model contient les entités. Elles sont constituées d'un ensemble de POJO persistés en base. Elles sont utilisées par la couche DAO pour la persistance, et par le contrôleur pour pouvoir convertir les données d'entrée, venant de la vue, en objet.
- Le module Consumer, la couche de persistance ou DAO. Cette couche est appelée par la couche service et permet de persister en base de données les entités envoyées en paramètre.
- Le module Business, la couche service ou bien métier. Cette couche est appelée par la couche contrôle et définit plusieurs méthodes qui permettent de définir les aspects fonctionnels de l'API.
- Le module Webservice est enfin celle qui contient les Endpoint qui jouent le rôle de contrôleur ici.

Les relations entre les entités en BDD sont mappées grâce à l'interface JPA.



L'application Web permet aux usagers de suivre leurs prêt, de les prolonger, de voir les livres disponibles en prêt, de rechercher un livre. Pour cela elle consomme les services provenant du webservice. Elle a été développée avec les framework Struts2 et Spring.

L'application est aussi sur une architecture MVC matérialisé aussi découpé par une struture multi-tiers :

- La couche Model

-La couche Consumer, à la différence du webservice ne se connecte pas à la base de données mais consomme les services provenant de ce dernier. Les classes générées à partir des WSDL y figurent ainsi que les clients qui permettent de consommer le service.

-La couche business, elle contient la logique métier qui sera appelée dans le contrôleur.

-La couche Webclient, elle contient les classes Actions de struts qui jouent le rôle de contrôleur ainsi que le fichier Struts.xml qui mappent les différentes sorties et les vues JSP correspondantes.

Le batch se charge de consulter les prêts dont la date est expirée et d'envoyer un email aux utilisateurs concernés. L'application a été développée avec le framework spring-boot-batch.