

DOCUMENTATION TECHNIQUE

PROJET 4 : MISE EN PLACE DU SYSTEME D'INFORMATION DE LA BIBLIOTHEQUE

▪ **Introduction**

Dans le cadre du quatrième projet, le présent document vise à présenter l'ensemble des fonctionnalités du système d'information de la bibliothèque ainsi que l'analyse conceptuelle, cette dernière est représentée par le diagramme de classe UML.

Ce document met en avant les choix techniques effectués : l'architecture du système ainsi que les différents outils permettant de la concrétiser. En effet, l'objectif principal de ce projet est d'employer les web services qui font partie de l'architecture SOA.

▪ **Objectifs du projet**

Les objectifs du système de bibliothèque à réaliser sont :

- La consultation de la liste de tous les ouvrages en ayant la possibilité de savoir l'état de leur disponibilité.
- La recherche multicritère des ouvrages.
- La connexion et la déconnexion du bibliothécaire.
- L'accès au tableau de bord qui décrit l'état de tous les emprunts effectués par l'utilisateur.
- La possibilité de prolonger un prêt.

▪ **Structure du projet**

Le présent projet est construit de trois parties distinctes :

- Les web services qui sont considérés comme des briques indépendantes servant à communiquer avec la base de données, c'est grâce à cette partie qu'on concrétise l'architecture SOA.
- L'application web qui, au lieu d'interroger directement la base de données, utilise ce qui est généré depuis les fichiers WSDL pour reproduire les résultats souhaités.
- Le batch qui est responsable de l'envoi automatique des emails aux utilisateurs n'ayant pas rendu les ouvrages avant que le délai soit terminé.

- **Rôles**

Après avoir analysé l'ensemble des fonctionnalités demandées, il est possible d'identifier deux rôles différents qui sont décrits ci-dessous :

- L'utilisateur : peut, à travers l'application, consulter tous les ouvrages proposés par la bibliothèque municipale et savoir leur état de disponibilité.
- L'utilisateur avancé : doit disposer d'un identifiant et mot de passe pour pouvoir accéder à son espace personnel, ce dernier permet de lister tous les prêts effectués et de savoir leur état (en cours, rendu ...). Le bibliothécaire peut également prolonger un emprunt donné à condition qu'il ne soit pas déjà prolongé.

- **Fonctionnalités du système d'information**

Le système de gestion de la bibliothèque fournit les fonctionnalités suivantes, regroupées en trois catégories principales :

- La gestion des ouvrages :

Cette catégorie permet de lister les ouvrages disponibles dans la base de données et d'effectuer une recherche multicritère en fonction du titre de l'ouvrage et du nom d'auteur.
- La gestion des auteurs :

Permet de lister les auteurs disponibles dans la base de données et d'avoir des informations sur leurs ouvrages.
- La gestion des utilisateurs :

Permet de récupérer les informations d'un utilisateur en fournissant son email et de lister tous les utilisateurs disponibles dans la base de données.
- La gestion des emprunts :

Comprend toutes les fonctionnalités concernant la consultation des emprunts, leur suivi et la possibilité de prolonger un prêt. Ces fonctionnalités sont offertes uniquement aux utilisateurs avancés disposant d'un identifiant et mot de passe.

- La relance des prêts :

Permet de relancer les prêts en retard c  d envoyer des emails aux utilisateurs n'ayant pas rendu les ouvrages au bon moment selon une fr  quence d  termin  e par la configuration du batch.

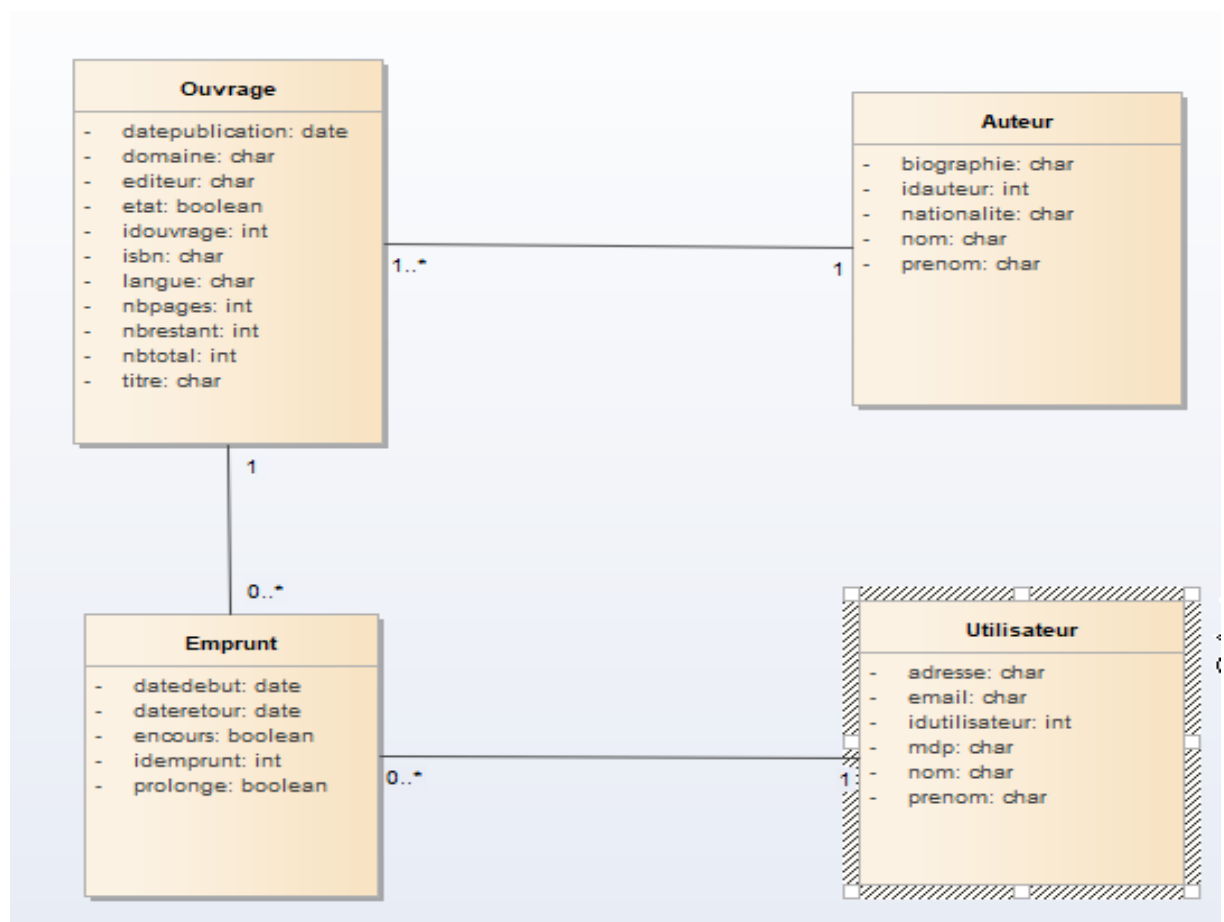
- L'authentification :

Sert    faire l'authentification des utilisateurs sur l'interface web du syst  me d'information. Elle concerne les fonctionnalit  s de connexion et de d  connexion des utilisateurs.

- **Diagramme de classe**

Apr  s avoir relev   toutes les fonctionnalit  s requises du syst  me d'information, le diagramme de classe est con  u de la mani  re suivante :

Les classes qui composent ce diagramme sont :



- **Ouvrage** : est une classe qui regroupe toutes les informations qui décrivent l'ouvrage, elle est liée à deux classes qui sont auteur et emprunt.
- **Auteur** : décrit les informations concernant l'auteur de l'ouvrage, un seul auteur écrit un ou plusieurs ouvrages.
- **Emprunt** : est la classe qui détaille les informations de l'emprunt d'un ouvrage concernant un utilisateur, elle est donc liée à deux classes qui sont ouvrage et utilisateur.
- **Utilisateur** : est la classe qui contient les informations de l'utilisateur qui va se servir de l'application de bibliothèque, elle est liée à la classe emprunt.

Choix techniques

- **Architecture multi-modules**

Afin de gérer les dépendances utilisées, Maven s'avère un outil pratique de build et de packaging. Il permet la mise en place d'une architecture basée sur plusieurs modules (architecture multi-modules).

Dans le cadre de ce projet, deux projets Maven ont été créés :

- Le webservice : est découpé en quatre modules Maven.
- L'interface web qui comprend aussi le batch : est découpé en cinq modules Maven.

Les modules Maven communs sont :

- **Modèle** :
Ce module contient les classes décrivant le modèle fonctionnel de la solution.
- **Consumer** :

Au niveau du webservice : ce module se connecte directement à la base de données pour récupérer les informations nécessaires. Ce module est accessible depuis le module business.
- **Business** :

Ce module contient la logique métier du système d'information. Ce module est accessible depuis les classes webservices créés.
- **Webservice**:

Ce module décrit la logique d'exposition des webservices qui utilise l'api Jaxws.

- Web interface :

Définit les actions Struts (le contrôleur) ainsi les différentes pages web (html, css, bootstrap).

Le module supplémentaire qui concerne le batch définit le step qui consiste à envoyer des emails à un utilisateur d'une façon récurrente, il permet aussi la configuration de Spring batch.

▪ **Gestion des entités**

La manipulation des données dans la base de données est faite grâce à l'utilisation du driver JDBC de PostgreSQL. D'une autre part, le framework Spring fournit une interface nommée « RowMapper ». Cette dernière permet de simplifier le mapping des objets java (les entités) en fonction des données en base de données.

Spring fournit également une classe nommée « NamedParameterJdbcTemplate », elle permet de définir l'ensemble des opérations CRUD nécessaires pour la réalisation des besoins demandés.

Au niveau des entités, le projet emploie la notion des DTO (Data Transfer Object).

Comme le webservice est censé envoyer des données à l'application web, il est recommandé de les transformer (les entités) en DTO surtout que ces derniers aident à réduire la quantité des données transférées au système externe (le client).

De plus, des classes de conversion qui correspondent à chaque entité ont été prévues pour réaliser la transformation d'entités en DTO.

▪ **Architecture webservices**

Le développement des webservices et leur exposition est fait en suivant la méthode bottom-up qui consiste à générer le fichier wsdl qui joue le rôle d'intermédiaire entre le client et les webservice.

L'api Jaxws a beaucoup facilité la mise en place des webservices grâce à l'utilisation des annotations (@webservice, @webmethod...)

▪ **Mise en place du batch**

C'est grâce à l'utilisation de Spring batch que le module batch a pu être développé. Il sert à décrire le job et de configurer son exécution.

Dans ce projet, un seul step est associé au job en question. Le step sert à envoyer l'email à l'utilisateur qui a tardé à rendre son ouvrage emprunté.