**Parcours « Expert Java EE » - Projet 4 - Développez le nouveau système d’information de la bibliothèque d’une grande ville.**

**Jérémy LOOTENS**

**Contexte**

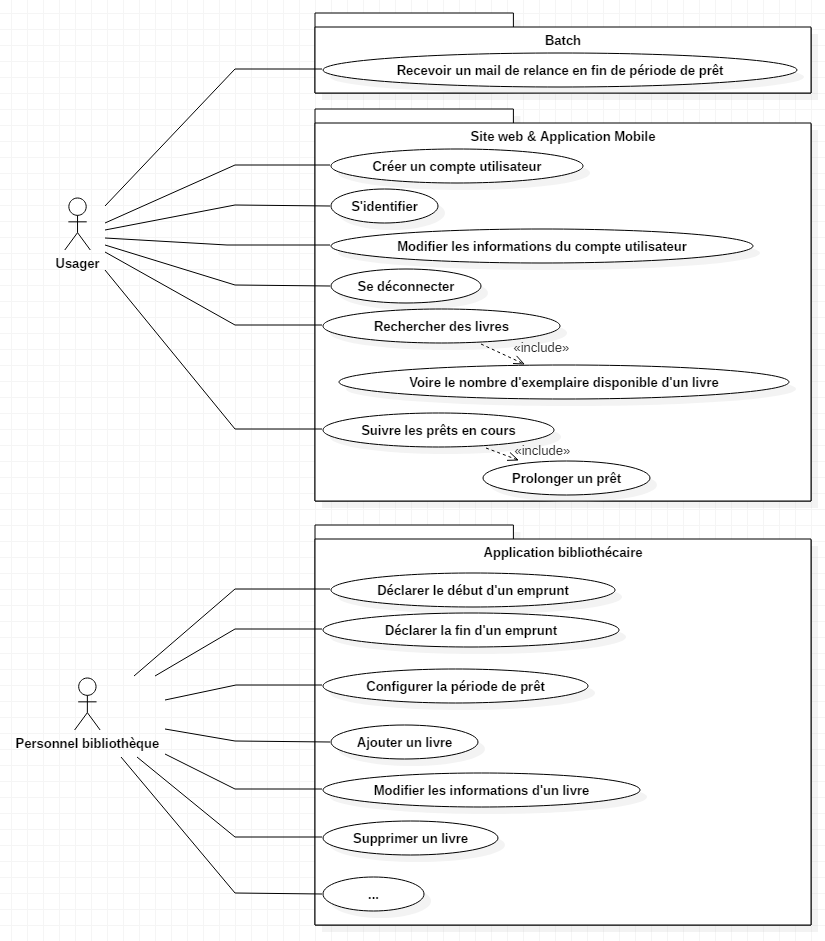
Le service culturel de « Trifoully-les-oies » souhaite moderniser la gestion de ses bibliothèques. Pour cela, il désire mettre à disposition des usagers, un système de suivi des prêts de leurs ouvrages.

Ce système doit comprendre :

* un site web en responsive design accessible aux usagers et permettant :
  + de rechercher des ouvrages et voir le nombre d’exemplaires disponibles ;
  + de suivre leurs prêts en cours ;
  + de prolonger un prêt ;
* une application mobile iOS et Android fournissant les mêmes services que le site web ;
* une application spécifique pour le personnel des bibliothèques permettant, entre autres, de gérer les emprunts et les livres rendus ;
* un batch lancé régulièrement et qui enverra des mails de relance aux usagers n’ayant pas rendu les livres en fin de période de prêt.

**Cas d’utilisation**

L’analyse du besoin exprimé a permis de déterminer le diagramme de cas d’utilisation suivant pour l’application :

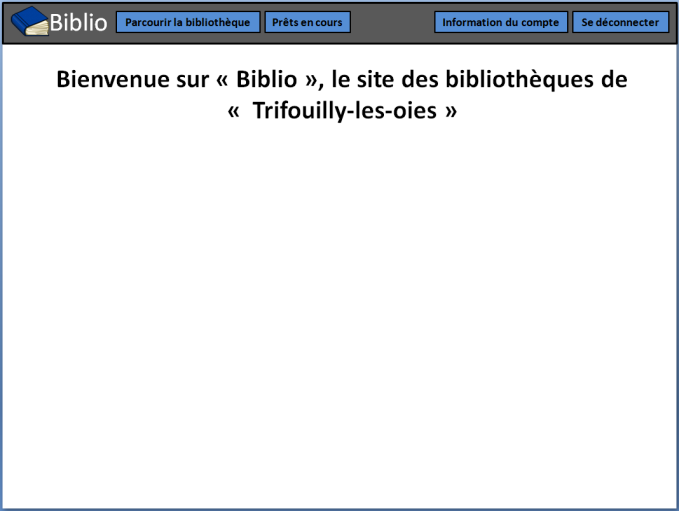
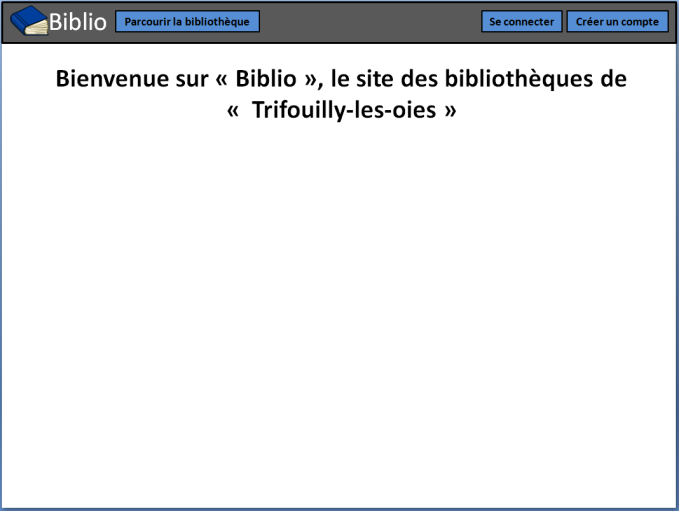


*Remarque : Ici seul le site web et le batch sont à développer, l’analyse des cas d’utilisation de l’application bibliothécaire (zone grisée) n’est donc pas développée et reste partielle.*

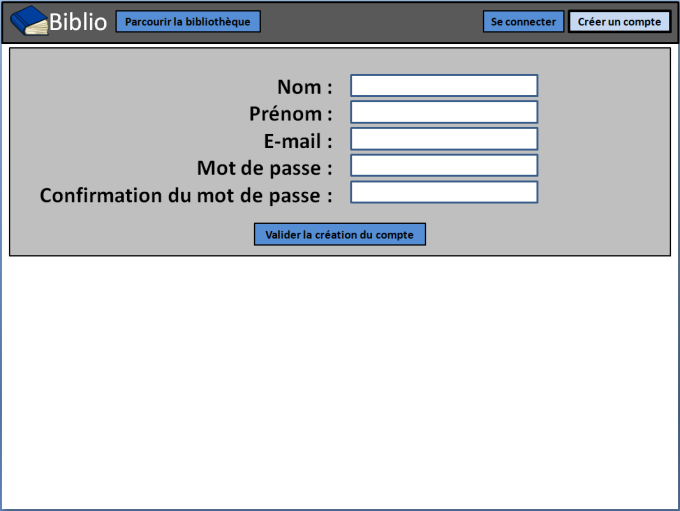
**Wireframes du site web**

A partir des cas d’utilisations ci-dessus, les wireframes ci-dessous ont été réalisées afin de répondre au mieux aux besoins avec une facilité d’utilisation maximale. La navigation à travers les vues se fait principalement grâce à la barre de navigation.

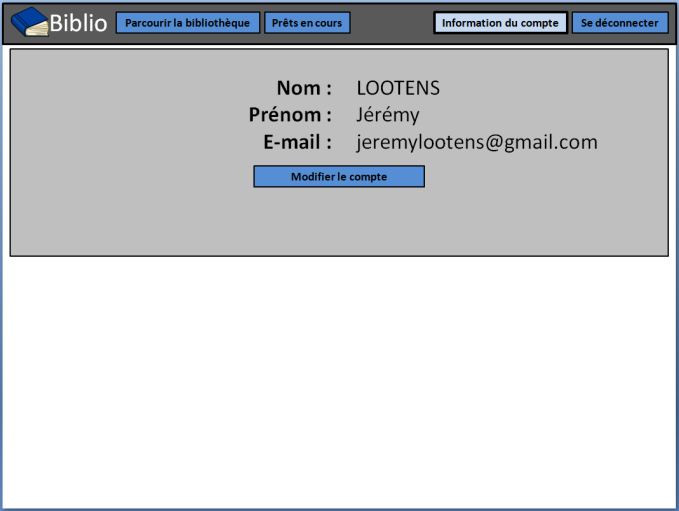
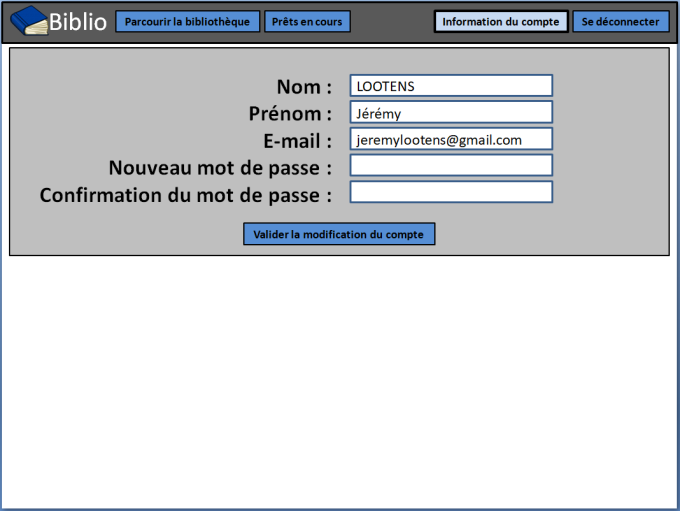
Page d’accueil (avant et après connexion de l’utilisateur)



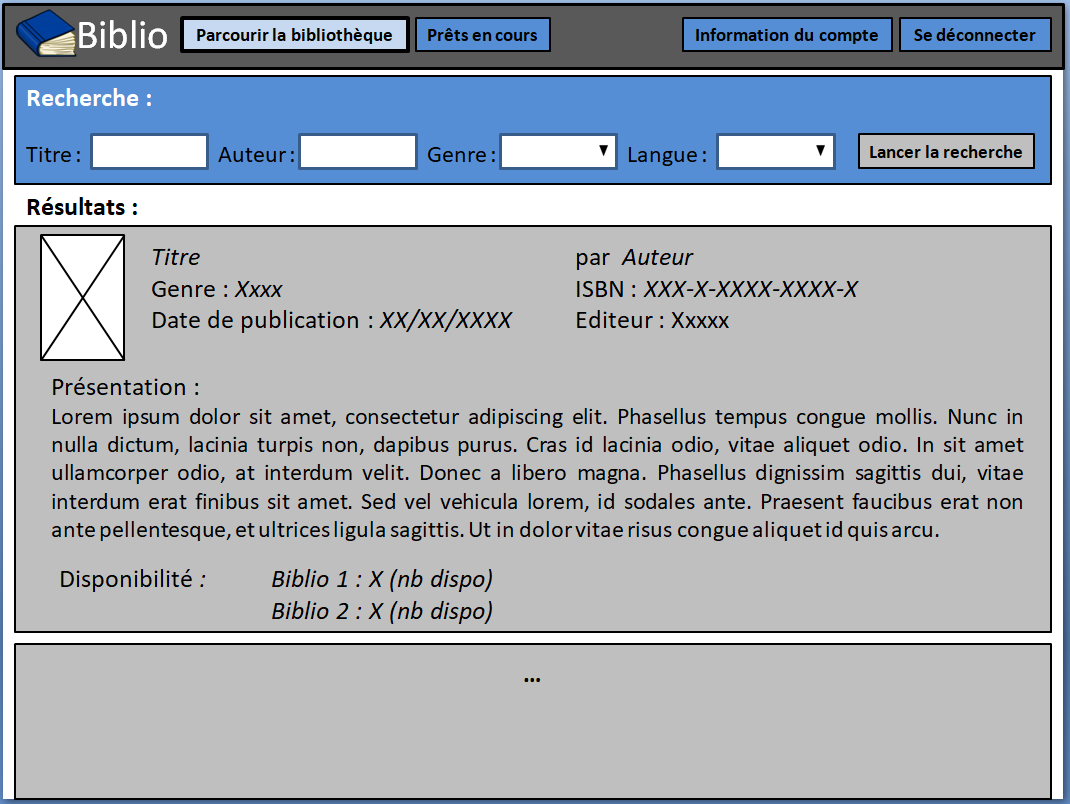
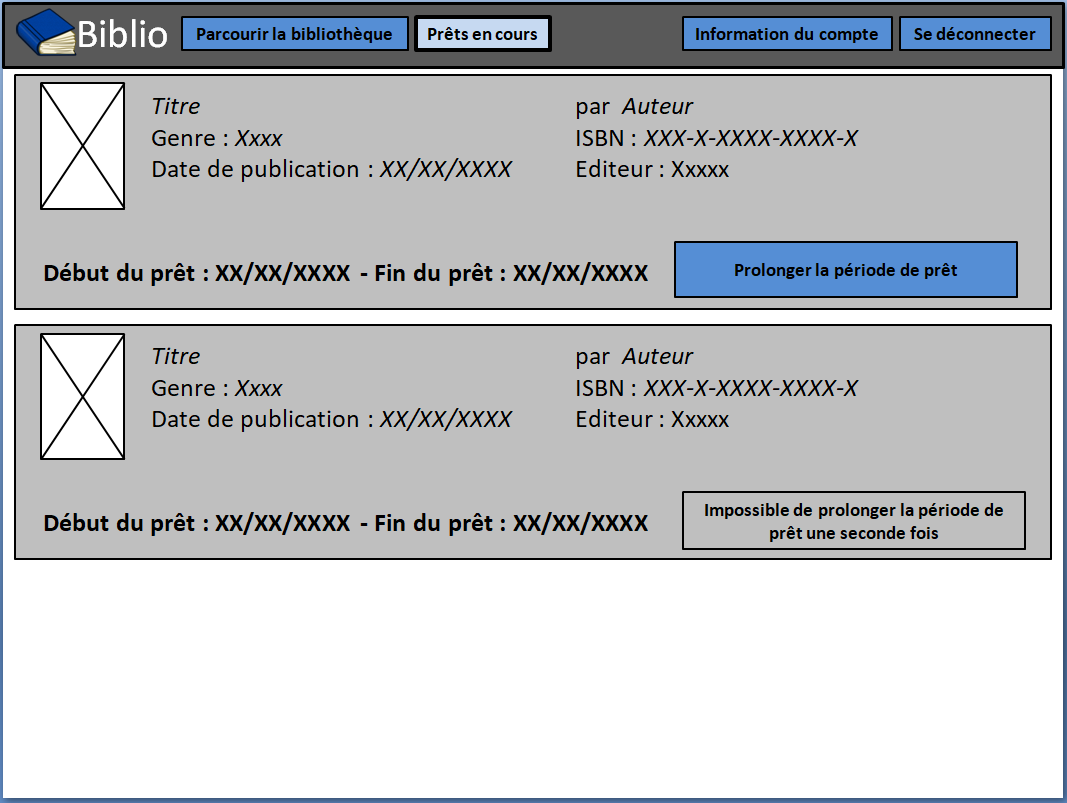
Création de compte utilisateur Connexion



Information du compte utilisateur Modification du compte utilisateur

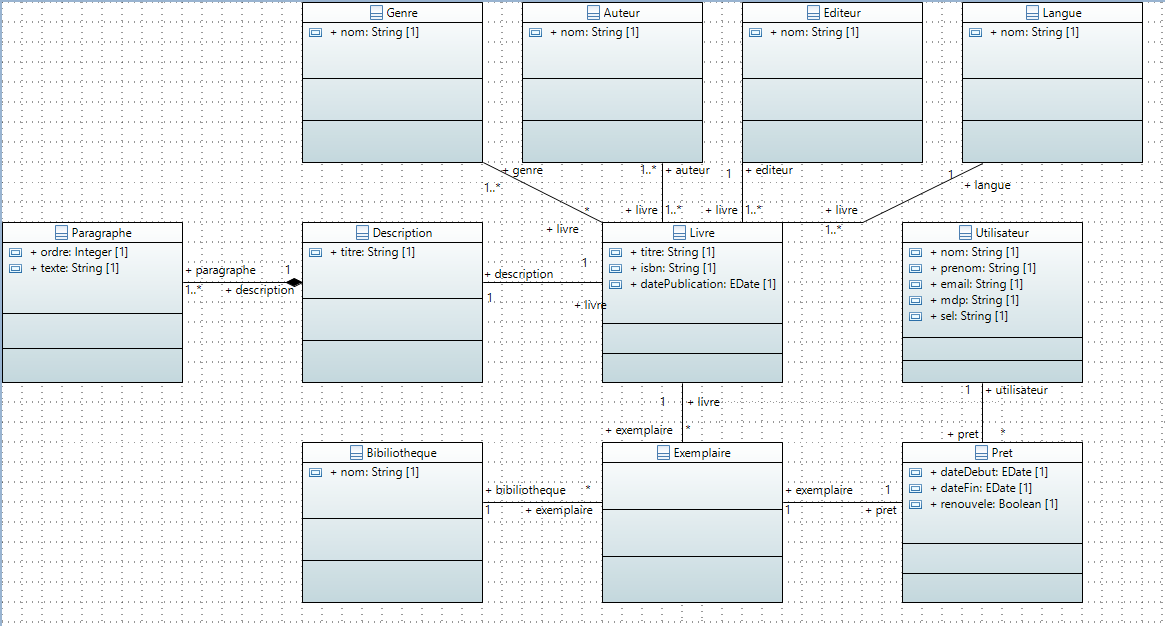
 

Recherche de livre Prêt en cours par l’utilisateur

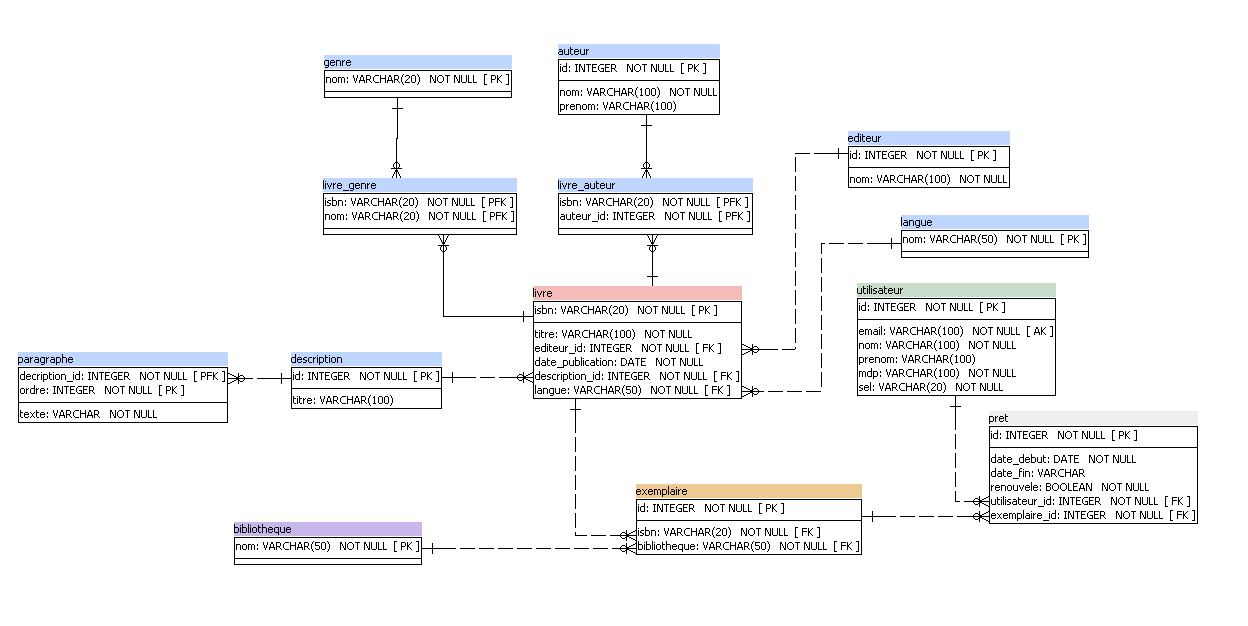
 

**Domaine fonctionnel et modèle physique de données**

L’analyse du domaine fonctionnel à mener à la création du diagramme UML de classes suivant :



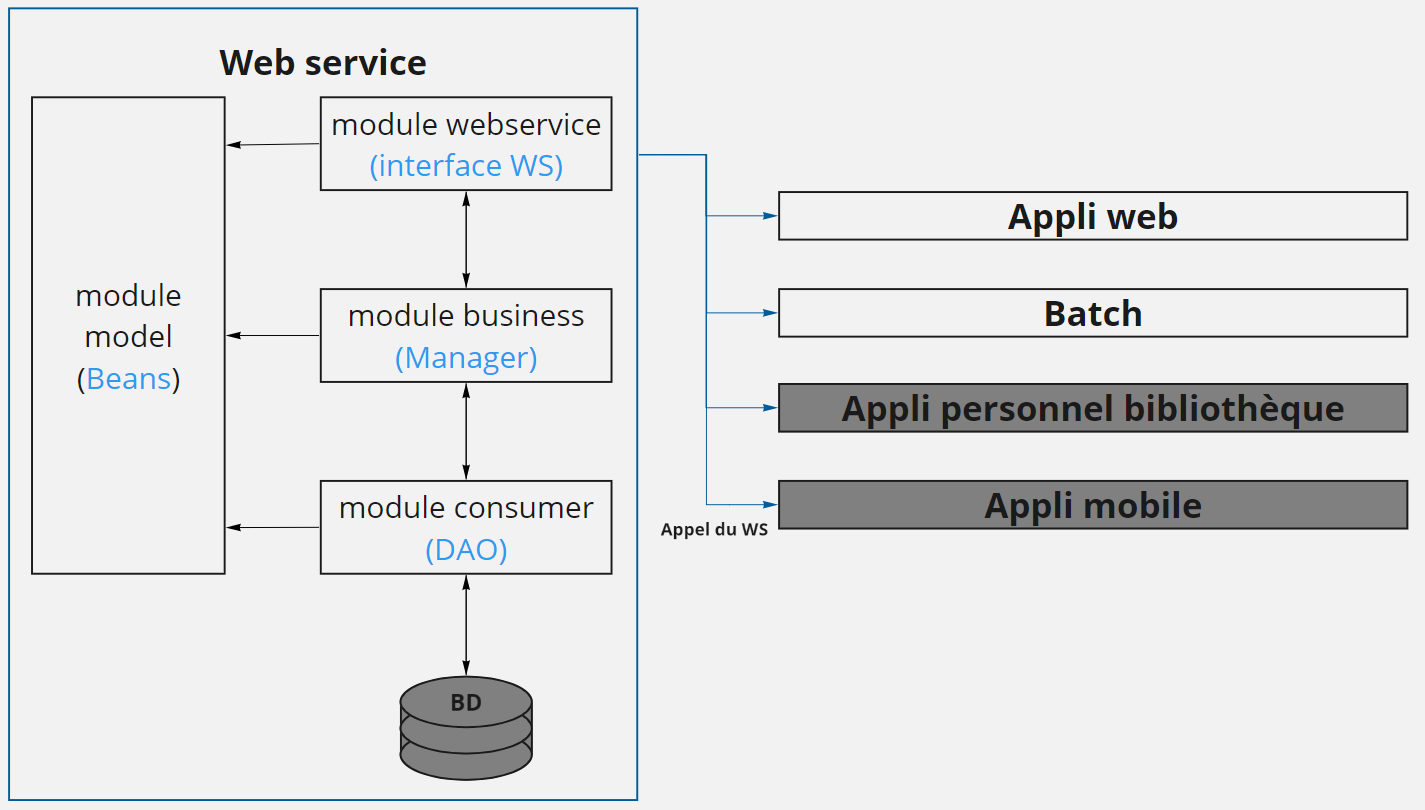
A partir de ce diagramme de classes, il en a été déduit le modèle physique de données suivant :



**Solution technique**

Le choix a été pris, afin de centraliser la gestion métier des différentes applications, de travailler autour d’une architecture orientée services (SOA pour Service Oriented Architecture).

L’architecture suivante a alors été fixée :



On retrouve à gauche le web service avec sa structure interne basée sur une architecture multi-tiers :

* Le module consumer compose la couche persistance et contient les DAO réalisant l’accès à la base de données (lecture et écriture).
* Le module business compose la couche métier du web service et donc de l’ensemble des applications.
* Le module model compose comme sont nom l’indique la couche modèle de l’architecture multi-tiers et contient les objets métiers du web service
* La couche webservice compose la gestion et l’interface du web service.

On retrouve également à droite les différentes applications qui feront appel au web service pour effectuer les opérations métiers nécessaires.

Les différents éléments ci-dessus reposent sur les technologies suivantes :

* Java 8 ;
* Apache Tomcat 9 ;
* PostgreSQL 10 ;

et les Frameworks Java suivants :

* Maven
* Spring (IoC, JDBC, TX, Mail,…) ;
* Struts2 ;
* Apache CXF.

**Mise en œuvre (configuration et déploiement)**

L’ensemble des éléments nécessaires à la configuration et déploiement du web service et des applications est disponible dans le dossier intitulé « livrables ». La procédure suivante permet le déploiement et la configuration de la base de données, du webservices, de l’application web et du batch.

1. Créer la base de données PostgreSQL (10)
   1. Créer dans PgAdmin la base de données
   2. Au choix :
      1. Restaurer la base de données à partir du fichier de "Backup" de pgAdmin suivant : livrables/BaseDeDonnees/backup.sql (clic droit > Restore > filename puis chercher le fichier backup.sql).
      2. Utiliser le script de création des tables et des données de démonstration livrables/BaseDeDonnees/creationBD.sql.
2. Déploiement du Web Service (sur serveur Tomcat 9)
   1. Configurer les ressources JNDI dans le fichier apache-tomcat-9.X.X/conf/context.xml en ajoutant les ressources suivantes (voir l'exemple Projet4/livrables/WebServices/context.xml) :
      1. DataSource pour configurer l'accès à la base de données (url, username, password) ;
      2. Durée maximale d'un prêt.
   2. Déployer sur Tomcat le fichier war suivant : Projet4/livrables/WebServices/biblioService.war
3. Déploiement du batch d'envoi de mail de relance
   1. Décompresser le fichier Projet4\livrables\Batch\biblioBatch-1.1.zip
   2. Créer une variable d'environnement nommée "biblioBatch\_Home" et contenant le chemin d'accès au dossier "biblioBatch-1.1" contenu dans le dossier décompressé.
   3. Modifier le fichier biblioBatch-1.1\conf\config.properties pour configurer les éléments suivant du Batch :
      1. Message et périodicité des mails de relance ;
      2. Configuration SMTP ;
      3. Adresse du Web Service LivreService.
   4. Exécuter biblioBatch-1.1.jar
4. Déploiement de l'application web (sur serveur Tomcat 9)
   1. Configurer les ressources JNDI dans le fichier apache-tomcat-9.X.X/conf/context.xml en ajoutant les ressources suivantes (voir l'exemple Projet4/livrables/WebApp/context.xml) avec l'adresse des Web services ;
   2. Déployer le fichier Projet4\livrables\WebApp\biblioWebapp.war