

# Modern Library

## Documentation Générale

### Table des matières

1.Principes généraux de l'application Modern Library.....	2
a.Contexte.....	2
b.Contraintes.....	2
c.L'application Modern Library.....	2
2.Architecture générale.....	4
3.Domaine fonctionnel et modèle physique de données.....	5
a.Domaine fonctionnel.....	5
b.Modèle physique de données.....	5
4.Cas d'utilisation - Diagrammes UML.....	6
a.Modern Library Web Services.....	6
Cas d'utilisation « Gestion de l'administration ».....	6
Cas d'utilisation « authentification ».....	7
Cas d'utilisation « Gestion des livres ».....	8
b.Modern Library Clients.....	9
Cas d'utilisation « Authentification ».....	9
Cas d'utilisation « Gestion des emprunts ».....	9
Cas d'utilisation « Gestion des livres ».....	10
5.Principales règles de gestion.....	11
a.Modern Library Web Services.....	11
b.Modern Library Web Clients.....	13
c.Modern Library Web Batch.....	14
6.Solution technique.....	15
a.Modern Library Web Services.....	15
b.Modern Library Clients.....	15
c.Modern Library Batch.....	16
7.Déploiement.....	17
a.Modern Library Web Services.....	17
b.Modern Library Clients.....	17
c.Modern Library Batch.....	17

# **1. Principes généraux de l'application Modern Library**

## **a. Contexte**

Le service culturel d'une grande ville souhaite moderniser la gestion de ses bibliothèques. Pour cela, elle désire mettre à disposition de ses usagers, un système de suivi des prêts de leurs ouvrages.

## **b. Contraintes**

Ce système comprendra :

- 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. Les prêts sont pour une période de 4 semaines (durée configurable)
  - de prolonger un prêt. Le prêt d'un ouvrage n'est prolongeable qu'une seule fois. La prolongation ajoute une nouvelle période de prêt (4 semaines, durée de prolongation configurable) à la période initiale
- 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

## **c. L'application Modern Library**

L'application Modern Library est actuellement composée de 3 parties principales.

Une partie Webservice, Modern Library Web Services, fournit un ensemble de webservices qui seront consommés par les applications tiers. Il s'agit du cœur de l'application Modern Library.

Une partie application web, Modern Library Clients, est destinée aux clients de la bibliothèque. Cette application doit permettre aux clients entre autre de rechercher des ouvrages, de prolonger ses emprunts, de consulter les emprunts en cours...

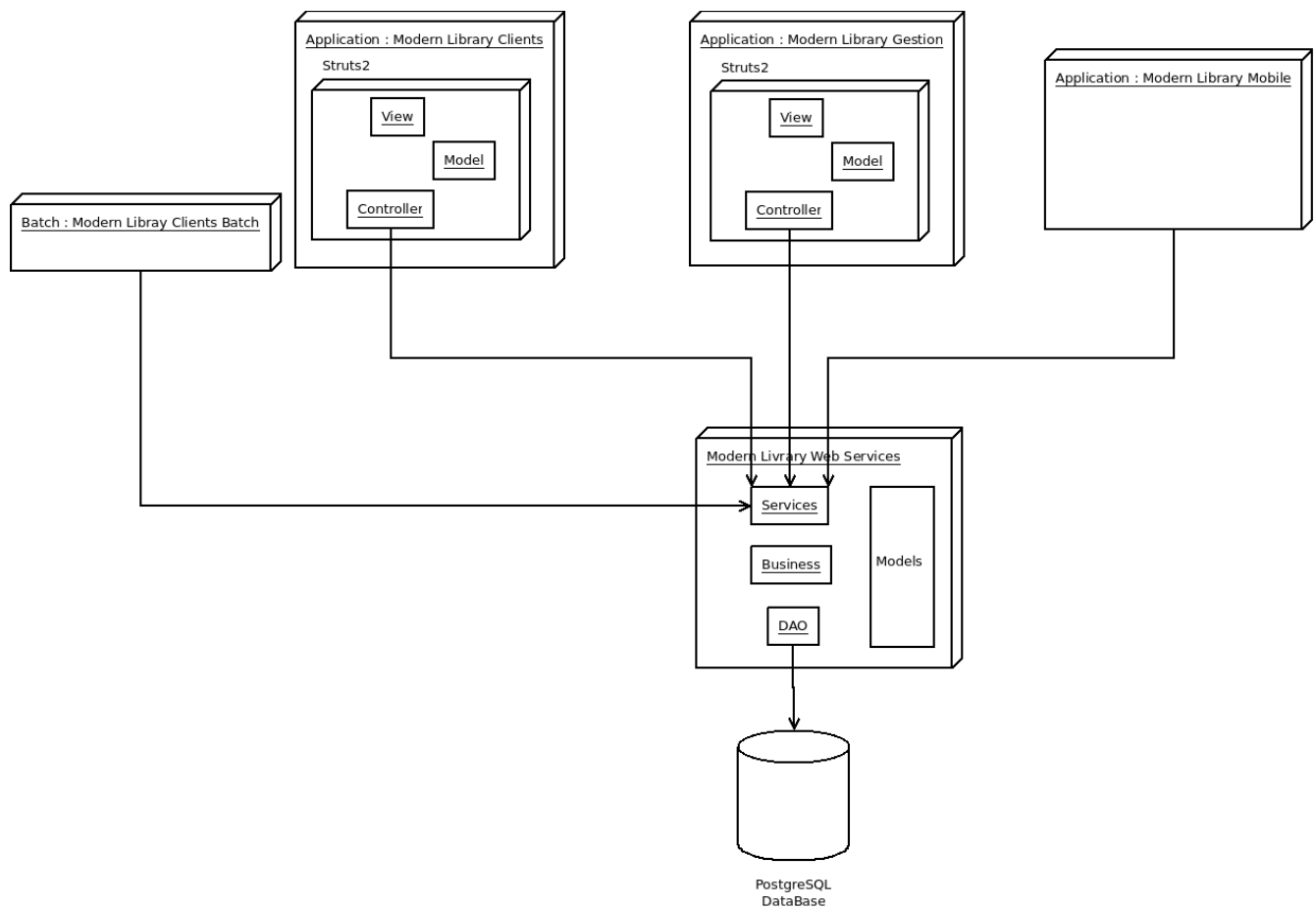
Une partie batch, Modern Library Batch, qui permet l'envoi d'un mail de relance aux usagers n'ayant pas rendu les ouvrages dont la période de prêt est terminée.

Dans l'avenir, deux parties supplémentaires seront ajoutées dans une prochaine livraison.

Une partie mobile, Modern Library Mobile, qui sera l'interface destinée aux utilisateurs sur appareil mobile.

Une partie gestion, Modern Library Gestion, qui sera l'interface à destination des employés de la bibliothèque à des fins de gestion.

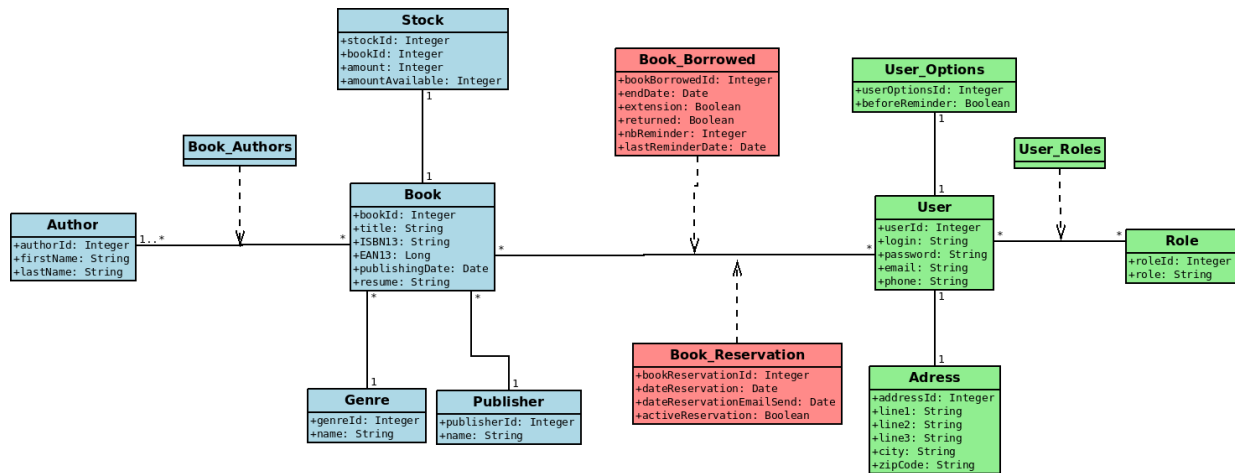
## 2. Architecture générale



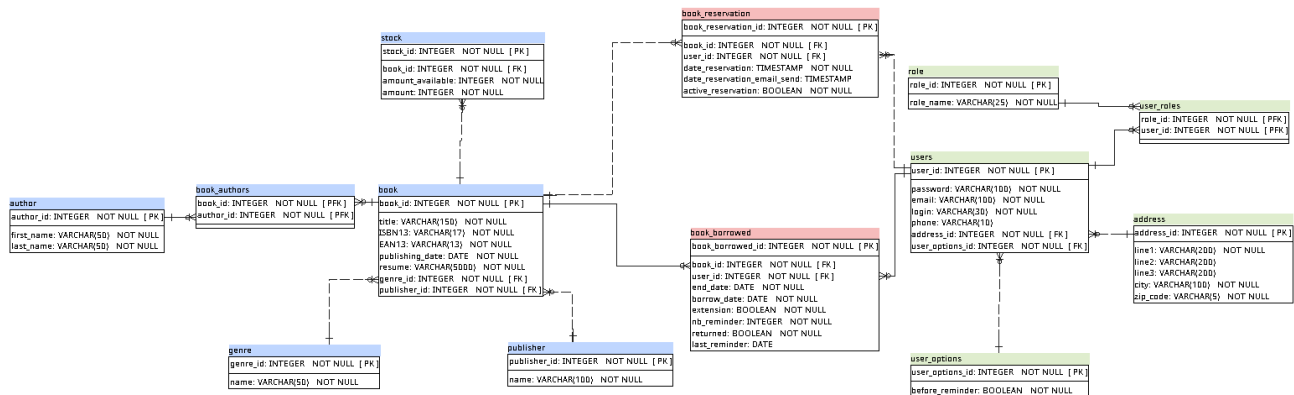
Comme indiqué dans la partie précédente, les applications Modern Library Mobile et Modern Library Gestion ne sont pas développés actuellement.

### 3. Domaine fonctionnel et modèle physique de données

#### a. Domaine fonctionnel



#### b. Modèle physique de données

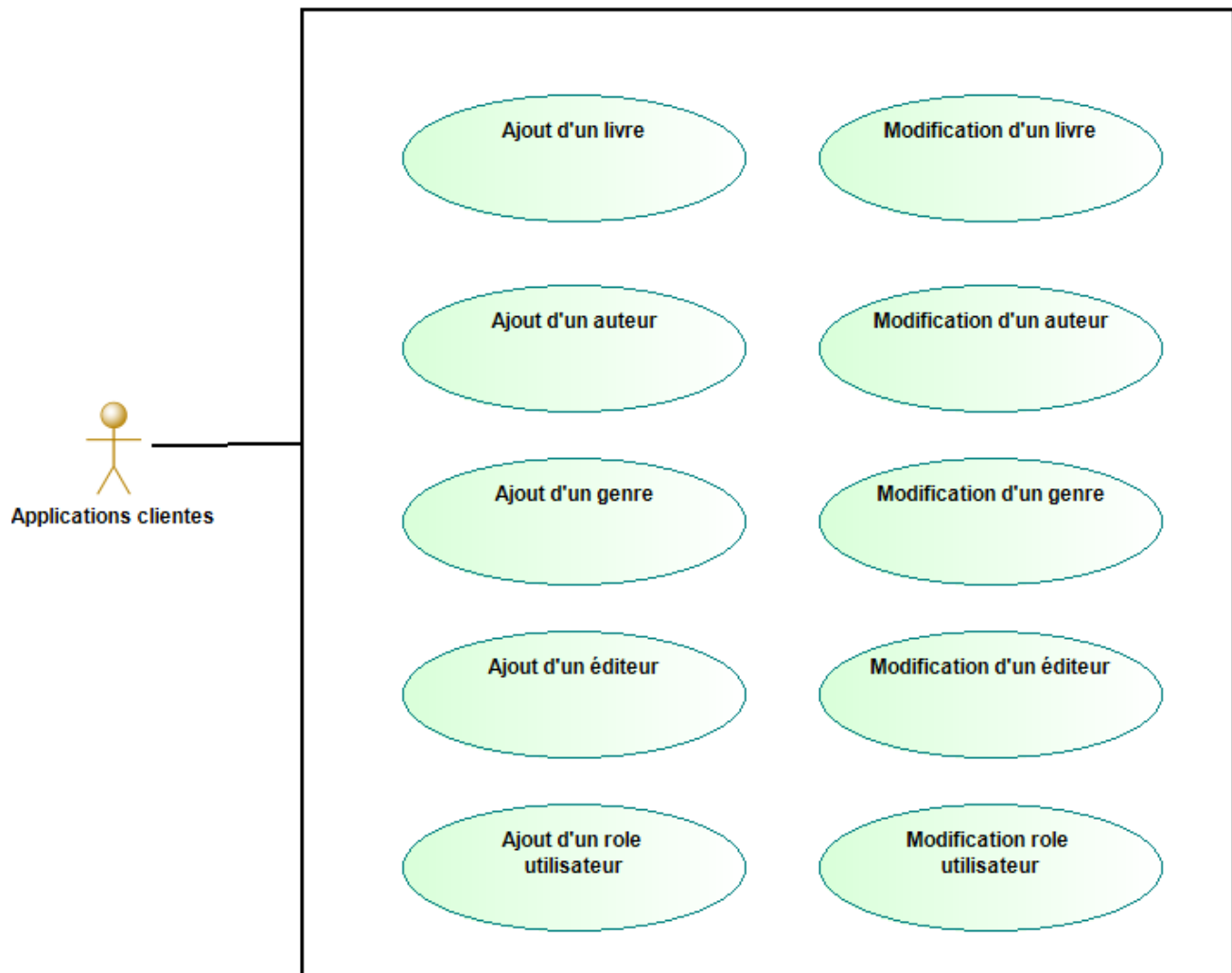


Pour une meilleure lisibilité, le modèle physique de données est également joint en document pdf à part.

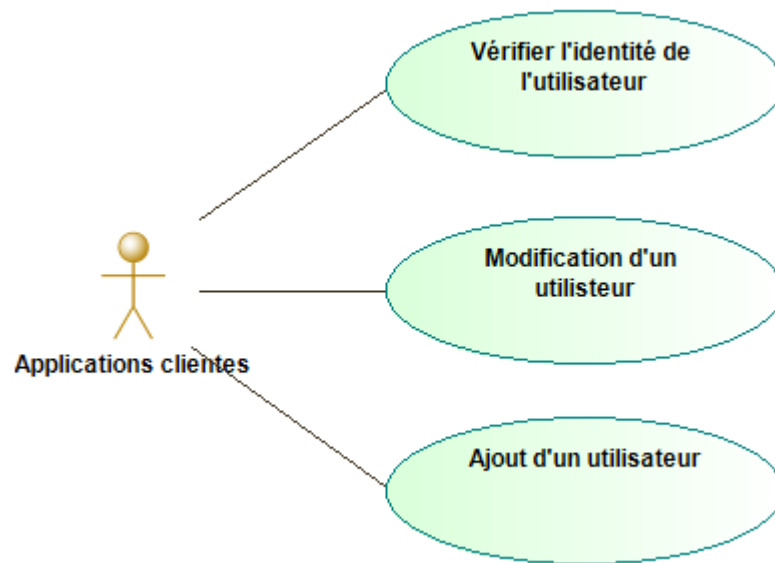
## **4. Cas d'utilisation - Diagrammes UML**

### **a. Modern Library Web Services**

- Cas d'utilisation « Gestion de l'administration »

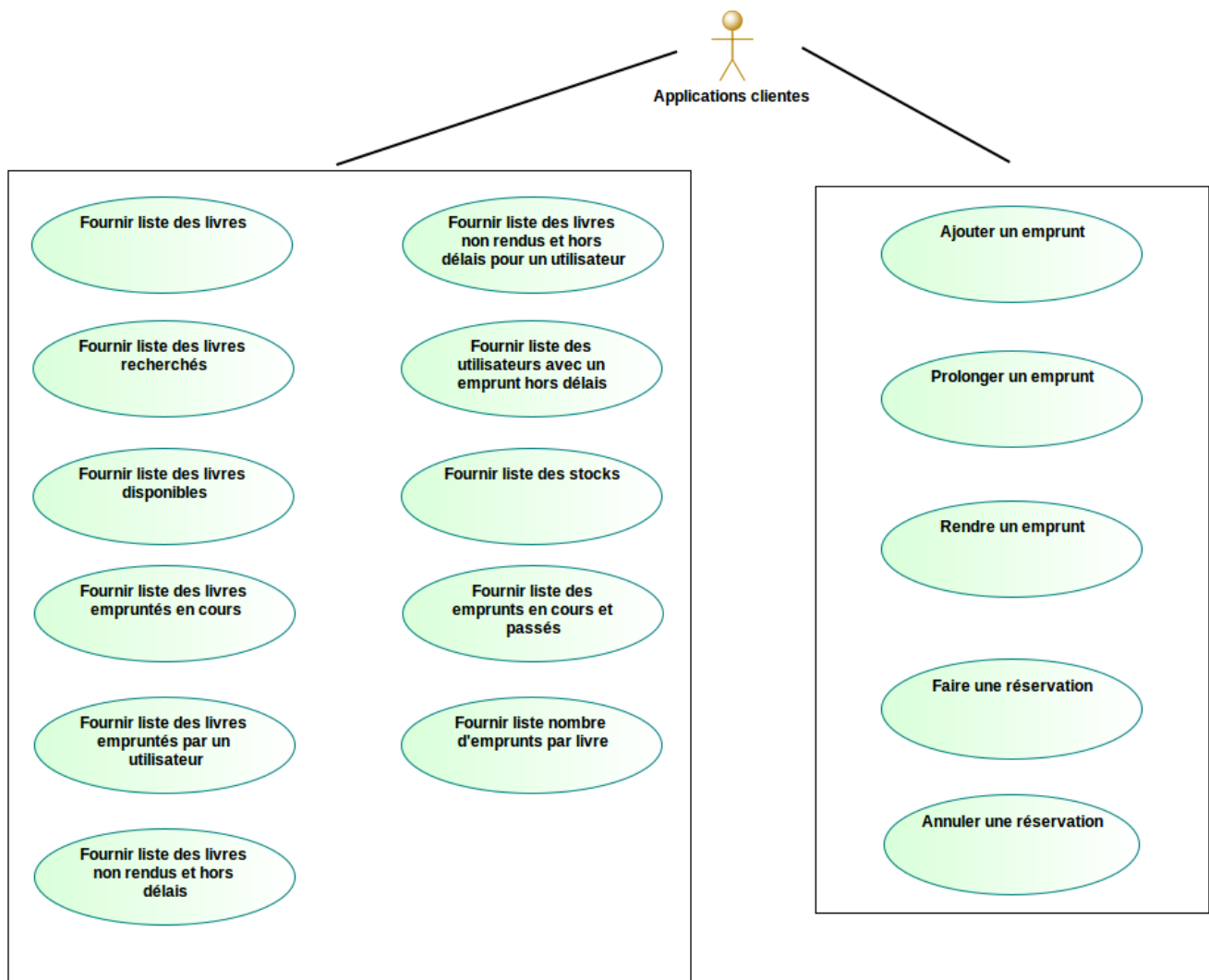


- Cas d'utilisation « authentification »



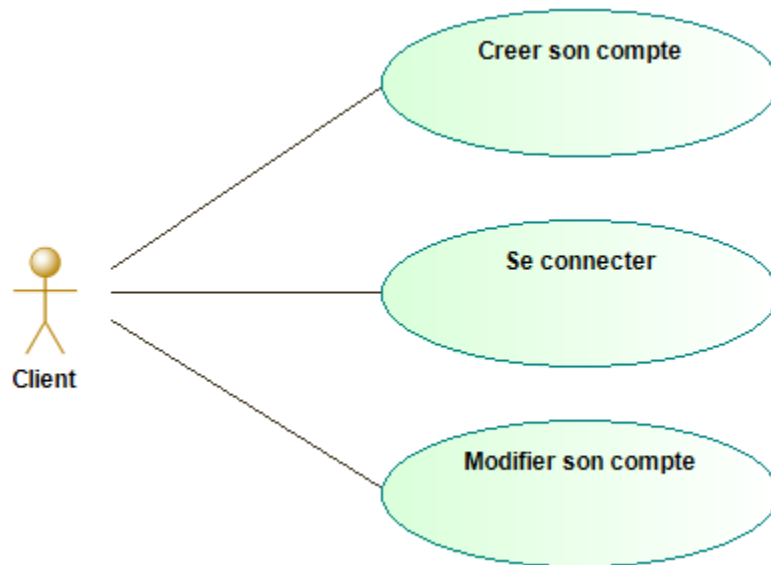


## • Cas d'utilisation « Gestion des livres »

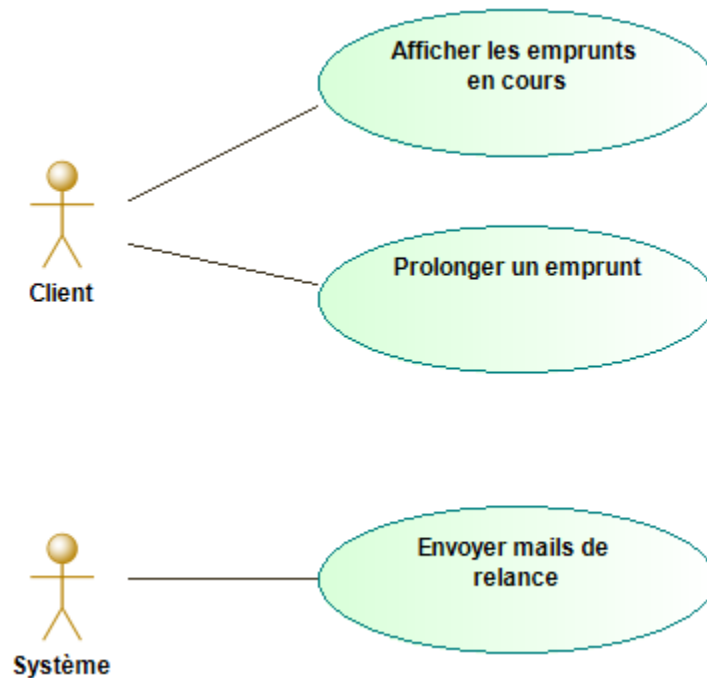


## b. Modern Library Clients

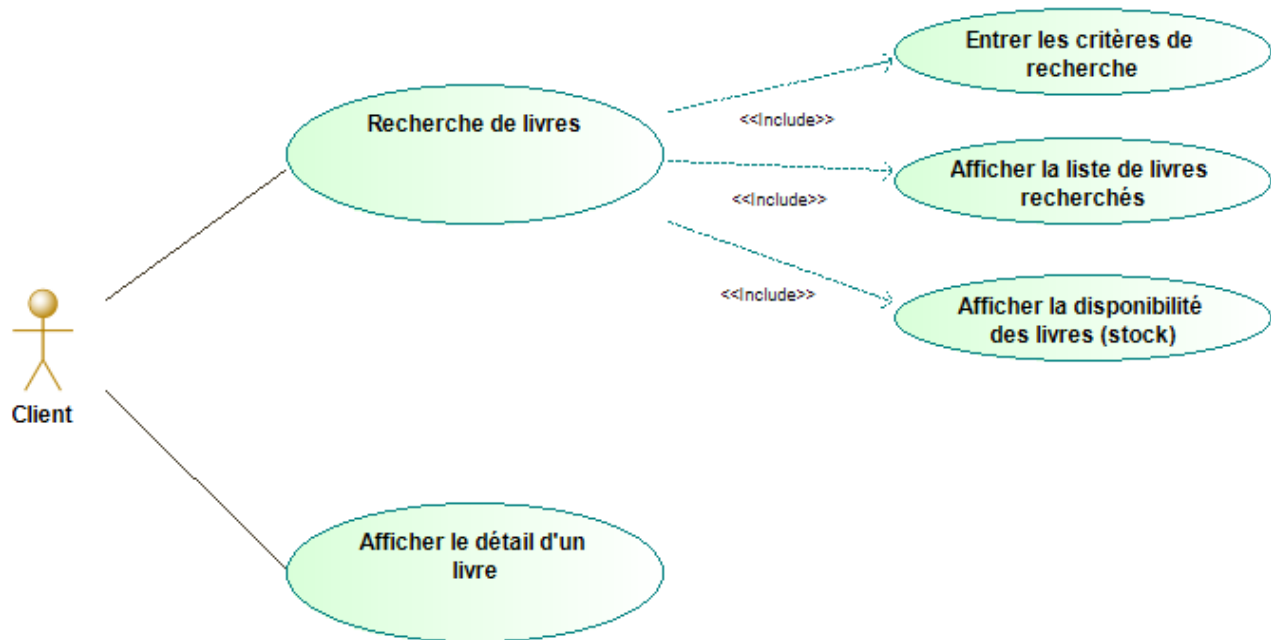
- Cas d'utilisation « Authentification »



- Cas d'utilisation « Gestion des emprunts »



- Cas d'utilisation « Gestion des livres »



## 5. Principales règles de gestion

L'intégralité des règles de gestion seront exhaustivement détaillées dans les spécifications fonctionnelles détaillées (SFD) et/ou les spécifications techniques détaillées (STD).

Ci-dessous nous listons les principales règles de gestion.

### a. Modern Library Web Services

Partie authentification :

- La web méthode « login » doit vérifier si le login et le mot de passe correspondent et renvoyer toutes les informations concernant l'utilisateur avec son adresse et ses rôles. Dans tous les autres cas, une SOAP Fault est renvoyée.
- La web méthode « modifyUser » doit modifier les informations de l'utilisateur à l'exception de son mot de passe et de ses rôles. Renvoie true si la modification a bien été fait. Dans tous les autres cas, une SOAP Fault est renvoyée.
- La web méthode « addUser » doit permettre l'ajout d'un utilisateur, adresse comprise. Le rôle de l'utilisateur n'est pas pris en compte. Renvoie l'identifiant du nouvel utilisateur si l'ajout s'est bien fait. Dans tous les autres cas, une SOAP Fault est renvoyée.
- La web méthode « addRoleToUser » doit permettre d'ajouter un rôle à l'utilisateur. Renvoie true si l'ajout a bien été fait. Dans tous les autres cas, une SOAP Fault est renvoyée.
- La web méthode « loginAvailable » doit permettre de vérifier si un login est disponible. Renvoie true si le login est disponible. Une SOAP Fault est renvoyée dans tous les autres cas.

Partie gestion des livres :

- La web méthode « currentlyBooksBorrowedList » doit fournir une liste contenant les informations sur les emprunts en cours avec le login des utilisateurs emprunteurs et les titre des livres empruntés. Dans tous les autres cas, une SOAP Fault est renvoyée,
- La web méthode « currentlyBorrowForUser » doit fournir une liste contenant les informations sur les emprunts en cours et les titres des livres empruntés pour un utilisateur donné. Dans tous les autres cas, une SOAP Fault est renvoyée.

- La web méthode « returnedBorrowsForUser » doit fournir une liste contenant les informations sur les emprunts et les titres des livres empruntés et rendus pour un utilisateur donné. Dans tous les autres cas, une SOAP Fault est renvoyée.
- La web méthode « currentlyDeadlineExpired » doit fournir une liste contenant les informations sur les emprunts, les titres des livre empruntés et les login et les mails des utilisateurs emprunteurs dont la date de retour a expiré. Dans tous les autres cas, une SOAP Fault est renvoyée.
- La web méthode « currentlyDeadlineExpiredForUsers » doit fournir une liste contenant les informations sur les emprunts, les titres des livre empruntés dont la date de retour a expiré pour un utilisateur donné et le login et le mails de l'utilisateur emprunteur. Dans tous les autres cas, une SOAP Fault est renvoyée.
- La web méthode « allBooks » doit fournir la liste de toutes les informations concernant les livres à l'exception des stocks et des emprunts. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode « booksSearchedList » doit fournir une liste de toutes les informations concernant les livres à l'exception des stocks et des emprunts. Cette liste doit correspondre aux critères de recherche renseignés ou non par l'utilisateur : titre du livre, genre du livre, éditeur du livre, ean13, nom de l'auteur, livre disponible ou non. Si aucun critère n'est fourni, la liste des informations de tous les livres doit être fournit. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode « availableBooksList » doit fournir une liste contenant les informations (à l'exception des stocks et des emprunts) concernant les livres disponibles pour un emprunt. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode « currentlyDeadlineExpiredUsers » doit fournir une liste contenant les informations sur les utilisateurs ayant un emprunt dont la date de retour est dépassé. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode « sendReminderToUsers » doit envoyer un email de rappel à tous les utilisateurs ayant un emprunt dont la date de retour est dépassée. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode « sendReminderBeforeExpirationToUsers » doit envoyer un email de rappel à tous les utilisateurs ayant un emprunt dont la date de retour est inférieur à 5 jours. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode « bookStockList » doit fournir une liste contenant les informations sur les stocks et les titres des livres. Renvoie une SOAP Fault en cas de problème technique.

- La web méthode « borrowingHistoryList » doit fournir une liste contenant les informations sur tous les emprunts (rendus ou en cours) avec le login des utilisateurs emprunteurs et les titre des livres empruntés. Cette liste doit être triée par date d'emprunt décroissant. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode « nbBorrowingByBooks » doit fournir une liste contenant le nombre d'emprunts réalisés pour chaque livre ainsi que le titre et l'identifiant des livres. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode « bookBorrow » doit permettre d'enregistrer l'emprunt d'un livre particulier pour un utilisateur particulier. Cette méthode doit retourner l'identifiant de l'emprunt enregistré. Renvoie une SOAP Fault si l'enregistrement de l'emprunt n'a pu être réalisé.
- La web méthode « extendBorrowing » doit permettre de prolonger un emprunt particulier. Renvoi true si l'extension a bien été réalisée. Le nombre de jour d'extension doit pouvoir être changé via un fichier de configuration. Si la date de retour est déjà dépassé ou si une extension a déjà été faite, une SOAP Fault doit être renvoyé. Renvoie une SOAP Fault en cas de problème technique ou si l'extension ne peut se faire.
- La web méthode « borrowingReturn » doit permettre d'enregistrer le retour d'un livre emprunté. Renvoie true si l'enregistrement du retour s'est bien déroulé. Dans tous les autres cas, une SOAP Fault est renvoyée.
- La web méthode « bookDetails » doit fournir toutes les informations (à l'exception des stocks et des emprunts) d'un livre particulier. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode userReservations doit fournir les réservations d'un utilisateur particulier. Renvoie une SOAP Fault s'il n'y a pas de réservation ou s'il y a un problème technique.
- La web méthode cancelReservations doit permettre d'annuler une réservation particulière pour un utilisateur particulier. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode bookReservations doit fournir une liste contenant les informations sur les réservations pour un livre particulier. Renvoie une SOAP Fault en cas de problème technique.
- La web méthode makeReservation doit permettre de faire une réservation pour un livre particulier pour un utilisateur particulier. Renvoie une SOAP Fault en cas de problème technique ou si le livre ne peut être réservé.
- La web méthode checkReservationNotPickedUp doit vérifier s'il y a des réservations qui n'ont pas été emprunté dans le temps imparti. Annule les réservations qui ont dépassé la durée et envoi un email à l'utilisateur suivant parmi la liste des réservations pour le livre.

## Partie Administration :

- Les différentes règles de gestion des web méthodes de la partie administration ne seront pas décrites ici afin de ne pas alourdir cette documentation générale. Les différentes web méthodes de la partie administration permettent d'ajouter ou de modifier des livres, des auteurs, des genres, des éditeurs et des rôles. Une SOAP Fault est renvoyé si l'ajout ou la modification ne peut être faite.

## **b. Modern Library Web Clients**

- Seul un utilisateur connecté peut accéder à l'application (en dehors de la page d'accueil et la page de connexion et login). Si l'utilisateur n'est pas connecté, il est alors renvoyé sur la page d'accueil.
- Un nouvel utilisateur doit pouvoir s'enregistrer. Les champs obligatoirement renseignés doivent être les suivants : adresse ligne 1, ville, code postal, identifiant, mot de passe, email. Si un des champs n'est pas correctement renseigné, un message doit l'indiquer à l'enregistrement. Les données saisies doivent persister en cas d'erreur.
- Un utilisateur déjà enregistré doit pouvoir se connecter en saisissant son identifiant et son mot de passe. Si le mot de passe et/ou l'identifiant sont incorrect, un message doit s'afficher.
- Un utilisateur doit pouvoir modifier les informations de son compte. Le mot de passe et l'identifiant ne sont pas modifiables par l'utilisateur.
- Un utilisateur doit pouvoir indiquer s'il souhaite avoir un email de relance avant la fin de son emprunt.
- Un utilisateur doit pouvoir afficher la liste des emprunts en cours le concernant. Cette liste doit afficher à minima le titre du livre, le/les auteurs et la date de fin de l'emprunt.
- Un utilisateur doit pouvoir prolonger un emprunt en cours. Si l'emprunt a déjà été prolongé, l'extension n'est pas possible. Si la date de retour est dépassé, l'extension n'est pas possible. L'extension est prolongé d'un nombre de jour définit par l'administrateur d'application Modern Library Web Services.
- Un utilisateur doit pouvoir effectuer des recherches de livre avec les critères de recherche suivants : titre du livre, genre du livre, éditeur du livre, ean13, nom de l'auteur, livre disponible ou non. Si aucun critère n'est renseigné, tous les livres doivent être affichés. Le résultat de la recherche doit afficher les informations suivantes : le titre, le/les auteur(s), le genre, l'éditeur et le résumé du livre.

- L'utilisateur doit pouvoir voir le détails d'un livre à partir de sa liste de livres empruntés. Le détail d'un livre doit contenir les informations suivantes : le titre, le/les auteurs, l'éditeur, la date de publication, le genre, l'ean13 et le résumé.
- L'utilisateur doit pouvoir l'historique de ses emprunts. Une liste contenant les informations sur les emprunts terminés doit donc pouvoir être affichés. Les informations à afficher sont les suivantes : le titre, la date de début de l'emprunt, la date de fin de l'emprunt.
- L'utilisateur doit pouvoir se déconnecter de l'application. Une fois déconnecté, il doit rediriger vers l'accueil de l'application.
- L'utilisateur doit pouvoir réserver des livres qui ne sont plus actuellement disponibles (s'ils sont tous en cours de prêt).
- L'utilisateur doit pouvoir voir la liste de ses réservations.
- L'utilisateur doit pouvoir annuler une réservation.

### **c. Modern Library Web Batch**

- Le batch doit envoyer de façon périodique des mail de relance aux utilisateurs ayant des emprunts non rendu.
- Le batch doit envoyer les emails de rappel de fin d'emprunt approchant aux utilisateurs ayant des emprunts proche de l'expiration.
- La périodicité d'exécution du batch doit pouvoir être configurer.



## **6. Solution technique**

### **a. Modern Library Web Services**

Les technologies utilisées sont les suivantes :

- Apache Maven 3.5.3
- Tomcat 9.0.10
- Java JDK 8 version 162
- PostgreSQL 10.4
- Spring Framework 4.3.11.RELEASE
- Apache CXF 3.2.5
- Apache Log4j 2.11.0

L'application est structurée en multi-couches :

- library-ws-services est une couche de présentation des web services et des web méthodes.
- library-ws-business contient la logique métier des web services.
- library-ws-dao contient les éléments d'interaction avec la base de données.
- library-ws-entity contient les objets métiers et les objets de transfert de données.
- library-ws-technical contient les éléments et dépendances techniques de l'application (gestion des logs...).

### **b. Modern Library Clients**

Les technologies utilisées sont les suivantes :

- Apache Maven 3.5.3

- Tomcat 9.0.10
- Java JDK 8 version 162
- Apache Struts 2.5.16
- Apache CXF 3.2.5
- Apache Log4j 2.11.0
- Bootstrap 4.1.3

### **c. Modern Library Batch**

Le batch appelle via un fichier crontab le web service à exécuter avec cURL.

## 7. Déploiement

### a. Modern Library Web Services

Le fichier war de l'application Modern Library Web Services doit être déployé sur un serveur Tomcat 9.0.10.

La configuration du niveau de log peut se faire dans le fichier : `library-ws-technical/src/main/resources/log4j2.xml`.

Les propriétés métiers peuvent se configurer dans le fichier : `fr/brucella/projects/libraryws/business/config.properties` . Le nombre de jour d'emprunt par défaut (et de prolongation) se trouve dans ce fichier.

La configuration du compte mail pour l'envoi des mails de relance (host, port, username, password) se fait également dans le fichier :

`fr/brucella/projects/libraryws/business/config.properties`

La configuration de l'accès à la base de données (url du serveur, port, username, password) se fait dans le fichier : `library-ws/library-ws-services/src/main/webapp/META-INF/context.xml`

### b. Modern Library Clients

Le fichier war de l'application Modern Library Clients doit être déployé sur un serveur Tomcat 9.0.10.

Les artefacts JAX-WS générés (en ayant pris en compte de l'url d'accès aux web services) sont déjà présent dans le fichier .war fourni.

### c. Modern Library Batch

Dans un système Unix, il convient de déposer le fichier request.xml (dans opt/scripts/ par exemple) puis d'ajouter la ligne contenue dans le fichier crontab en adaptant l'url d'accès au web service si nécessaire.