

Library

Solution fonctionnelle et technique

Pour le système d'information d'une grande bibliothèque par
POUYADOU Even

Table des matières

I.	Introduction	2
	Objet du document	
	Contexte	
	Objectifs	
II.	Le domaine fonctionnel.....	3
	Description	
	Diagramme de classe	
	Description des classes	
III.	Le modèle physique de données	6
	Description	
	Modèle physique de données (MPD)	
	Description des tables	
IV.	Composants internes et externes du système	8
	Description	
	Diagramme de composant	
	Description du composant	
V.	Architecture de déploiement	9
	Description	
	Matériels des utilisateurs	
	Serveur web (Site PC)	
	Serveur Web (Web Service)	
	Batch	
	Serveur de base de données	
	Diagramme de déploiement	
VI.	Règle de gestions	11
VII.	Instructions de déploiement et configuration.....	12
	La base de données	
	Le Web Service	
	Le Site client et le batch	
VIII.	Conclusion	14

Introduction

Objet du document

Ce document définit la solution fonctionnel et technique adoptée pour la conception de l'application web, le web service et le batch du système d'information.

Ce document présente le modèle fonctionnel du système et le modèle physique de données qui serviront à la conception de la base de données de « Library » (nom du système d'information). Il sera aussi fourni avec ce document les scripts de déploiement de cette dernière.

De manière à comprendre un peu mieux cette solution technique et les liens qu'elle peut avoir avec des outils externes, ce document présente l'analyse des composants internes et externes.

Contexte

Ce document découle du cahier des charges suivant :

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. 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 le 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 À vous de réaliser ce système !

Ce document concerne uniquement le site web, le web service et le batch.

Contrainte

Le déploiement du système sera assuré par le personnel de la direction des systèmes d'information de la ville. Vous devez donc leur laisser la possibilité de modifier facilement les différents paramètres de configuration (URL et identifiant/mot de passe de la base de données, URL du webservice, envoi des mails...).

Objectifs

Dans l'optique de répondre au mieux aux attentes de notre client et de ses clients, ce document présentera une aide précieuse à la compréhension du système d'information et la manière de déployer ce dernier.

Il permettra aussi de repère pour de future modification du système d'information.

Le domaine fonctionnel

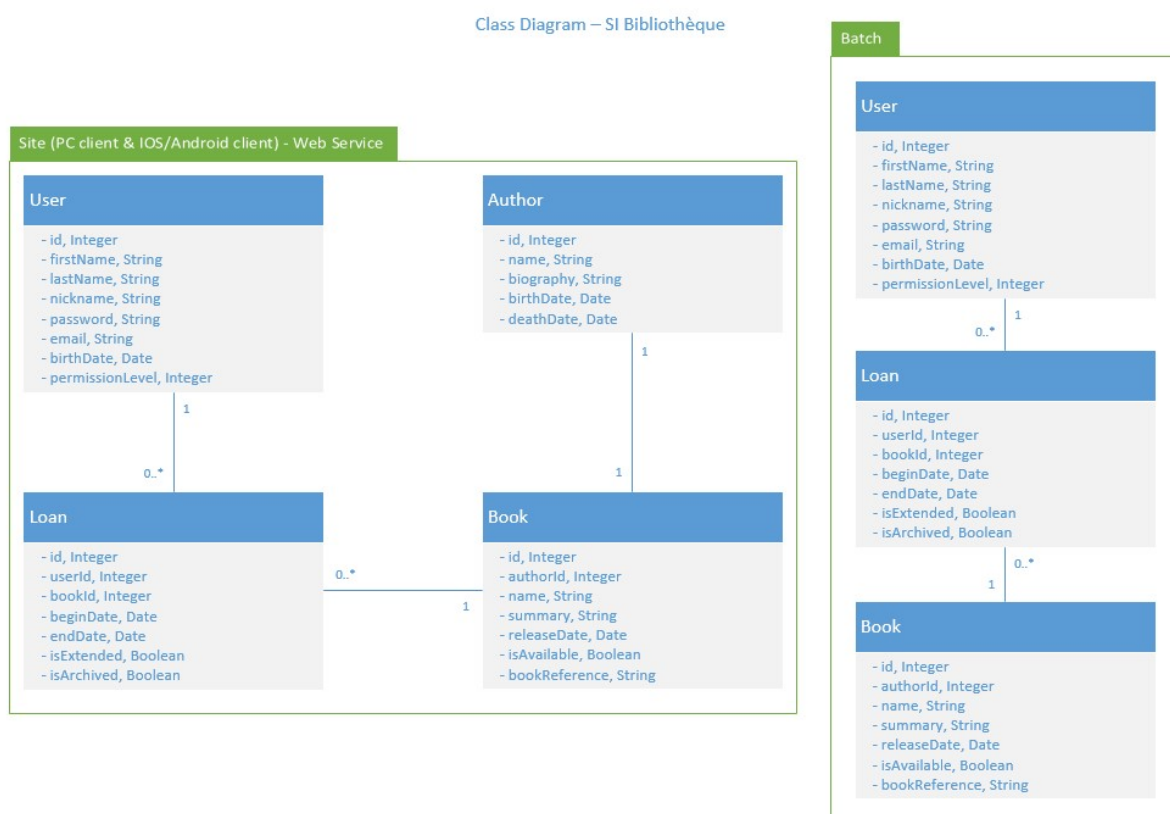
Description

Le domaine fonctionnel de l'application web, du web service et du batch définit l'ensemble des classes qui serviront de support lors de la programmation et à la modélisation du modèle physique de données (MPD). Le domaine fonctionnel est représenté par un diagramme de classes aux normes UML 2.5

Diagramme de classe

Le diagramme de classes représente l'organisation et les liens entre les classes. Il permet de mieux comprendre le fonctionnement de l'application.

Diagramme de classe UML



Description des classes

User

La classe « **User** » contient les informations principales d'une personne (Nom, prénom, pseudonyme, date de naissance, email, et mot de passe) mais aussi les permissions attribuées. Dans notre cadre, lorsqu'un utilisateur est créé, il a les permissions les plus basses.

Attribut	Description
firstName	Définit le prénom de l'utilisateur.
lastName	Définit le nom de l'utilisateur.
nickname	Définit le pseudonyme de l'utilisateur.
permissionLevel	Définit le type de permission de l'utilisateur.
birthDate	Définit la date de naissance de l'utilisateur.
email	Définit l'email de l'utilisateur.
Password	Définit le mot de passe « hashé » de l'utilisateur.

Author

La classe « **Author** » contient les informations relatives d'un écrivain.

Attribut	Description
name	Définit le nom et prénom de l'auteur.
biography	Définit la biographie de l'auteur.
birthDate	Définit la date de naissance de l'auteur.
deathDate	Définit la date de mort de l'auteur.

Book

La classe « **Book** » contient les informations relatives d'un livre

Attribut	Description
name	Définit le nom du livre.
authorID	Définit l'ID de l'auteur.
summary	Définit le résumé du livre.
releaseDate	Définit la date de parution du livre.
available	Définit si oui ou non le livre est disponible à la réservation.
Book_reference	Définit la référence du livre. La référence permet de regrouper les différents exemplaires du même ouvrage.

Loan

La classe « **Loan** » est une classe liée à un utilisateur et à un livre, car elle contient les informations liées à l'emprunt d'un livre (**Book**) par un utilisateur (**User**)

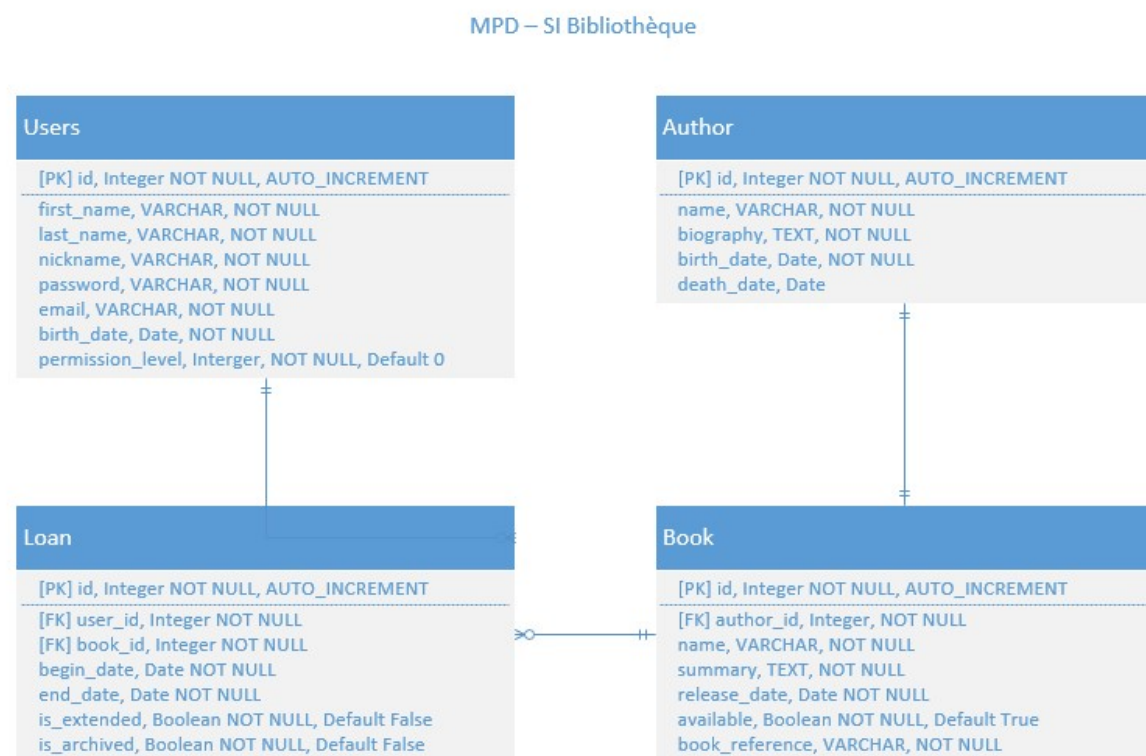
Attribut	Description
userID	Définit l'ID de l'utilisateur qui a emprunté.
bookID	Définit l'ID du livre emprunté.
beginDate	Définit la date du début de l'emprunt.
endDate	Définit la date buttoir pour rendre le livre.
isExtended	Définit si la date buttoir a été repoussé.
isArchived	Définit si le traitement de l'emprunt est terminé.

Le modèle physique de données

Description

Créé à partir du précédent diagramme, le modèle physique de données reprend globalement le diagramme de classe et spécifie les liaisons, les contraintes, les clés primaires et étrangères de chaque classe pour permettre la juste création de la base de données.

Modèle physique de données (MPD)



Description des tables

User

Attribut	Description
id	Clé primaire de la table « User », de type INTEGER, non nulle, avec auto-incrémentations.
first_name	Attribut de type VARCHAR, non nul.
last_name	Attribut de type VARCHAR, non nul.
nickname	Attribut de type VARCHAR, non nul.
password	Attribut de type VARCHAR, non nul.
birth_date	Attribut de type DATE, non nul.
email	Attribut de type VARCHAR, non nul.
permission_level	Attribut de type INTEGER, non nul.
Salt	Attribut de type VARCHAR, non nul.

Author

Attribut	Description
id	Clé primaire de la table « Author », de type INTEGER, non nulle, avec auto-incrémentations.
name	Attribut de type VARCHAR, non nul.
biography	Attribut de type TEXT, non nul.
birth_date	Attribut de type DATE, non nul.
death_date	Attribut de type DATE, non nul.

Book

Attribut	Description
id	Clé primaire de la table « Book », de type INTEGER, non nulle, avec auto-incrémentations.
name	Attribut de type VARCHAR, non nul.
summary	Attribut de type TEXT, non nul.
author_id	Attribut de type INTEGER, non nul.
release_date	Attribut de type DATE, non nul.
available	Attribut de type BOOLEAN, non nul.
book_reference	Attribut de type VARCHAR, non nul.

Loan

Attribut	Description
id	Clé primaire de la table « Loan », de type INTEGER, non nulle, avec auto-incrémentations.
user_id	Attribut de type INTEGER, non nul.
book_id	Attribut de type INTEGER, non nul.
begin_date	Attribut de type DATE, non nul.
end_date	Attribut de type DATE, non nul.
is_extended	Attribut de type BOOLEAN, non nul.
is_archived	Attribut de type BOOLEAN, non nul.

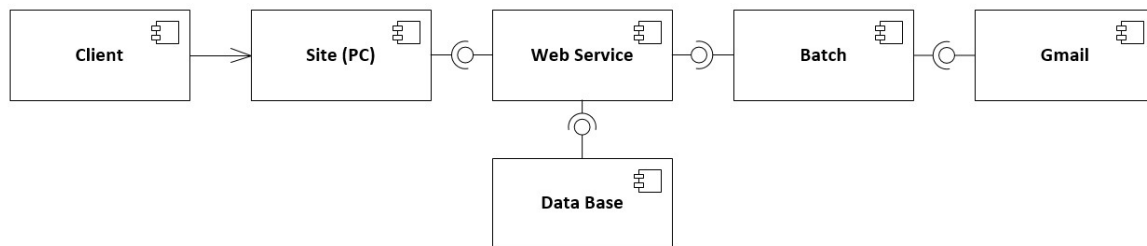
Composants internes et externes du système

Description

Le diagramme de composants décrit l'organisation de l'application web. Il met en évidence les dépendances entre les différents composants, et décrit les interfaces entre les composants internes et externes du système d'information.

Diagramme de composant

Diagramme de composant



Description du composant

Le composant « **Client** » représente des ordinateurs avec lesquels les clients se connectent.

Le composant « **Site (PC)** » communique avec le client lui envoie les pages demandées via des requêtes **HTTP**.

Le composant « **Web Service** » est le cœur du système. Ce composant est le seul à communiquer avec la base de données. Il donne via son interface, les données sur les utilisateurs, les livres, les auteurs et les emprunts de la bibliothèque.

Les composants « **Site** » et « **Batch** » envoient et reçoivent les données par fichier **Json** tout en utilisant le protocole **HTTP**.

Quant au composant « **Batch** » qui s'occupe d'envoyer un message tous les jours à 12H pour chaque livre emprunter et dont la date buttoir est dépasser pour le leur rappeler, via une adresse Gmail avec un protocole **POP/SMTP**.

Architecture de déploiement

Description

Le diagramme de déploiement vient apporter des informations sur la conception architecturale de l'application web. Elle permet de connaître comment les composants se connectent et quels sont leurs dépendances.

Matériels des utilisateurs

Les clients peuvent se connecter à partir de leur PC. Tous ces utilisateurs doivent cependant passer par un navigateur web, qui enverra des requêtes HTTP ou HTTPS au serveur Apache.

Serveur web (Site PC)

Le serveur web est un serveur Apache qui va gérer la connexion des clients et leur renvoyer les pages à afficher. Ce serveur utilisera le Framework Spring Boot en backend avec JavaEE pour tout le reste.

Ce site sera donc programmé en JavaEE, et interagira avec le Web Service, pour laisser au Web Service gérer toute interaction avec la base de données.

Serveur Web (Web Service)

Le Web Service est un serveur Apache qui va répondre aux connexions du « Site PC » et du « Batch » pour leur envoyer, suivant la requête reçue, les données attendues par ces derniers, à condition que la requête soit valide ou qu'il n'y ait pas d'erreur.

Le Web Service doit pouvoir communiquer avec la base de données.

Batch

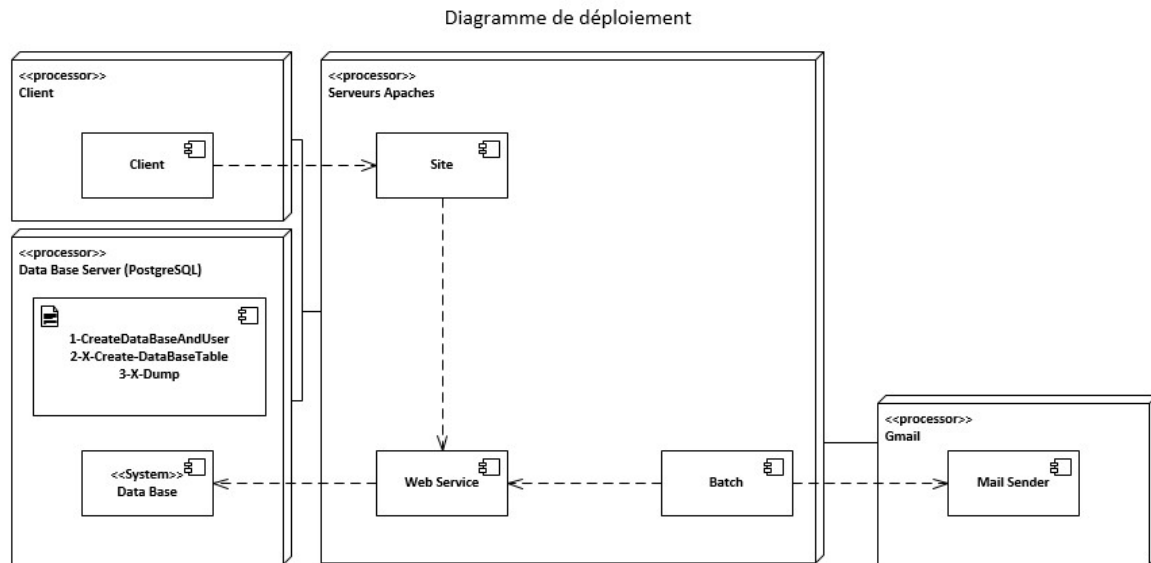
Le Batch est un serveur Apache qui à son lancement fait tourner une première fois la vérification des emprunts puis refait cette vérification tous les jours à 12H.

Il doit communiquer avec le Web Service pour obtenir les données des emprunts, des utilisateurs et des livres.

Serveur de base de données

Le serveur de base de données qui communique avec le serveur Apache tournera sur, au minimum la version 11 de PostgreSQL. Il pourra être hébergé sur la même plateforme que le serveur Apache ou ailleurs. L'application nécessitera donc une configuration lors du déploiement.

Diagramme de déploiement



Règle de gestions

Les règles de gestions concernant ces projets sont les suivantes :

- Un utilisateur ne peut avoir qu'un seul compte avec une même adresse ou un même pseudonyme
- Un utilisateur ne peut pas emprunter plus d'un exemplaire du même ouvrage
- A partir du site client, il est seulement possible d'emprunter des livres. Rendre un livre se fait à partir du site de l'administration.
- Un utilisateur doit être connecter pour emprunter un livre ou prolongé le prêt.
- Un utilisateur doit être connecter pour voir son profil et ses prêts.

Instructions de déploiement et configuration

Il y a 3 projets packager WAR :

- Le site client
- Le Web Service
- Le batch

Chacun de ces trois projets peuvent se situer sur des serveurs différents. Ces projets nécessitent d'être déployer sur des serveurs web. Il est cependant possible en changeant dans le fichier « pom.xml » de chaque projet de le packager en Jar pour pouvoir le lancer directement. Mais le fichier jar ne peut pas être déployer sur les serveurs web.

La base de données

Avant de mettre les mettre en place il faut installer et préparer la base de données. Pour ceci, il faut que sur le serveur qui gèrera la base de données, il y ait PostgreSQL v11 au minimum d'installé.

Ensuite utilisé les scripts fournit avec à exécuter dans l'ordre (suivre le chiffre).

Attention : Les scripts commençant par 3 sont les scripts du dump. Ce ne sont que des données de test !

Le Web Service

La mise en place du web service se fait avant le batch et le site client. Pour configurer le web service, il suffit de modifier le fichier « application.properties » dans le fichier « ressources » du projet.

Propriétés

server.port : correspond au port de connexion pour accéder au Web Service.
debug : correspond à l'affichage des message de debug dans la console.
spring.datasource.driverClassName : correspond au type de base de données à laquelle on va s'adresser.
spring.datasource.url : correspond à l'url de la base de données.
spring.datasource.username : correspond au nom d'utilisateur pour se connecter à la base de données.
spring.datasource.password : correspond au mot de passe pour se connecter à la base de données.
spring.jpa.database-platform : correspond au dialecte utilisé pour communiquer avec la base de données.

Il est conseiller de ne pas toucher aux autres propriétés.

Le Site client et le batch

La mise en place du site client et du batch se fait comme le Web Service. Pour configurer le site client et le batch, il suffit de modifier le fichier « application.properties » dans le fichier « ressources » du projet.

Propriétés du Site client

debug : correspond à l'affichage des message de debug dans la console.
spring.application.name : correspond au nom de l'application
library.serviceURL : correspond à l'url du Web Service
library.defaultReservationWeeks : correspond au nombre de semaine avant que l'utilisateur se doit de ramener le livre.
library.extentReservationWeeks : correspond au nombre de semaine ajouter lors de la prolongation d'un prêt.
library.numberElementByPage : correspond au nombre d'élément afficher par page dans l'affichage de la liste des livre et des auteurs.

Propriétés du Batch

debug : correspond à l'affichage des message de debug dans la console.

library.serviceURL : correspond à l'url du Web Service

spring.mail.host : correspond à l'hôte du service mail

spring.mail.port : correspond au port SMTP (en générale 587)

spring.mail.username : correspond à l'identifiant du compte qui va envoyer les mail (en générale une adresse email)

spring.mail.password : correspond au mot de passe du compte

spring.mail.properties.mail.smtp.auth : correspond à « si oui ou non » il y a besoin d'une authentification

spring.mail.properties.mail.smtp.starttls.enable : correspond à « si oui ou non » on active la connexion chiffré ssl/tls.

Conclusion

Suite à l'analyse du domaine fonctionnel, nous en avons déduit l'architecture de la base de données. Et nous avons émis nos avis sur le moyen de déployer l'application. La prochaine étape à ce document est le développement de l'application.

Lors de l'analyse de ce projet et des demandes du client, nous avons constaté que vos clients ont pour l'instant la possibilité de faire des emprunts mais n'ont pas la possibilité de laisser des commentaires sur chaque livre ou de le noter. Peut-être serait-il important de faire cet ajout !