# Documentation technique

## A. Spécifications techniques

Serveur :

- Apache
- PHP
- Extension PHP : PDO
- PostgreSQL
- XAMPP

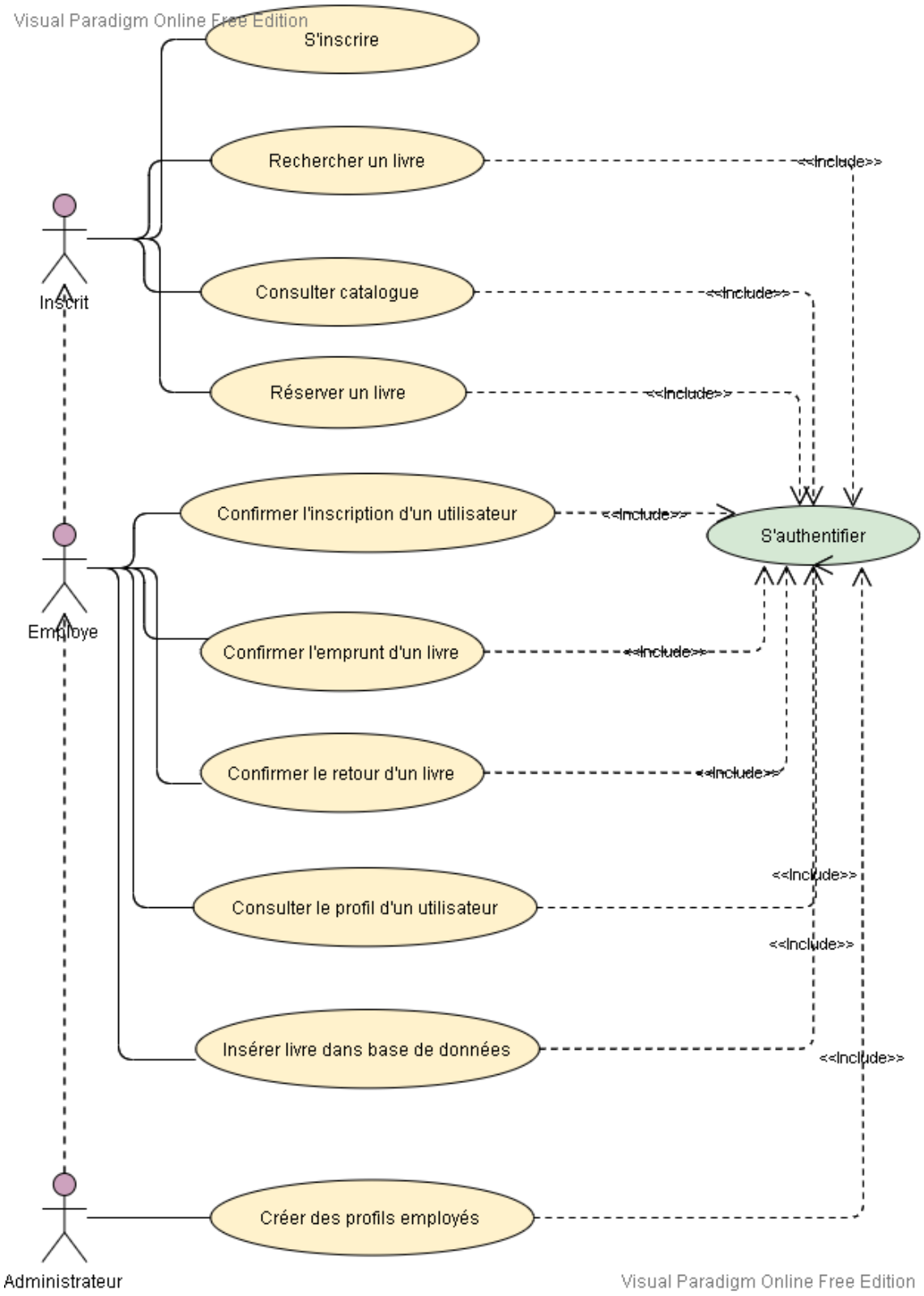
Front :

- HTML 5
- CSS 3
- Bootstrap
- JavaScript

Back :

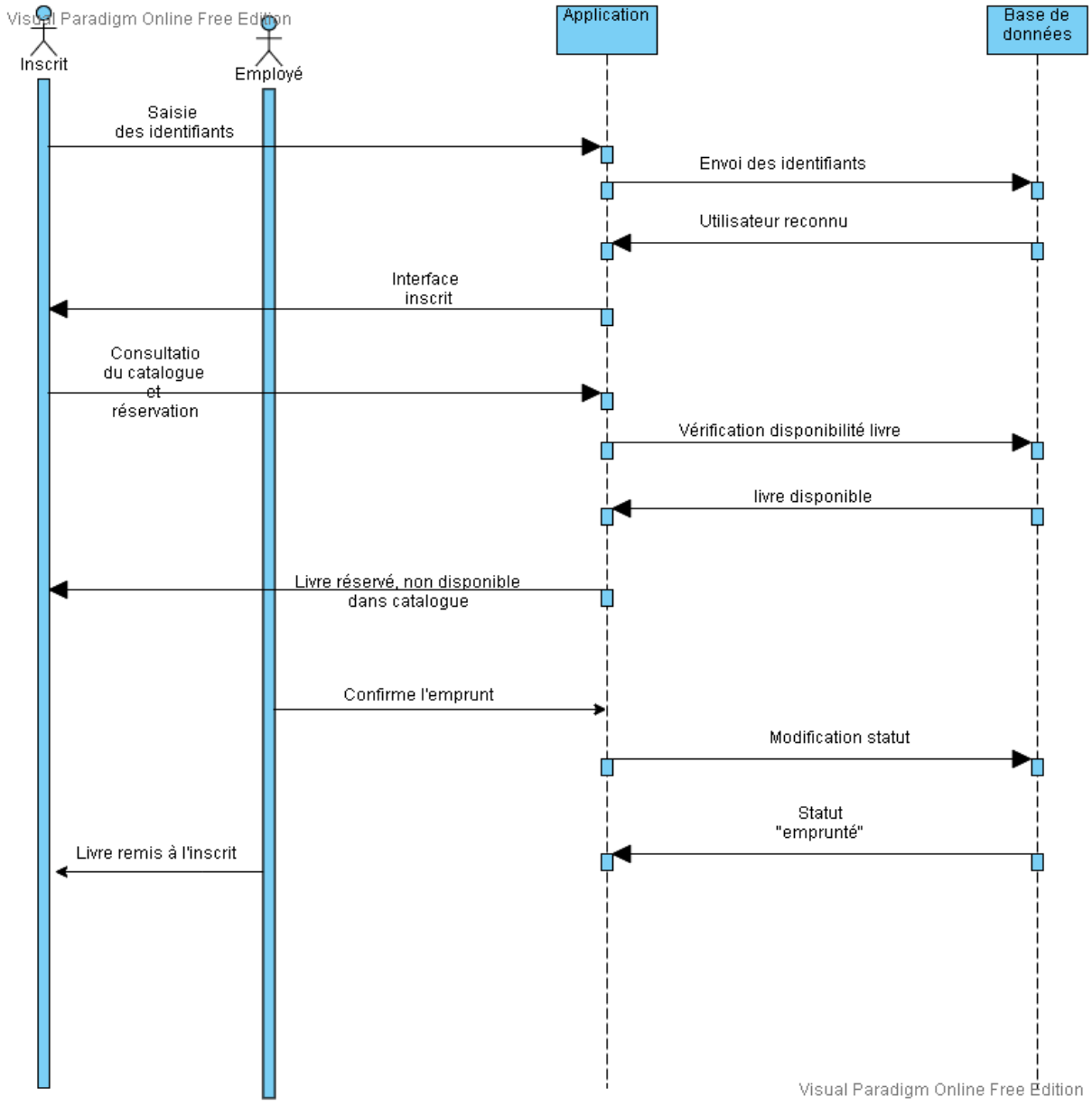
- PHP sous PDO
- PostgreSQL

## B. Diagramme de Cas d'utilisation

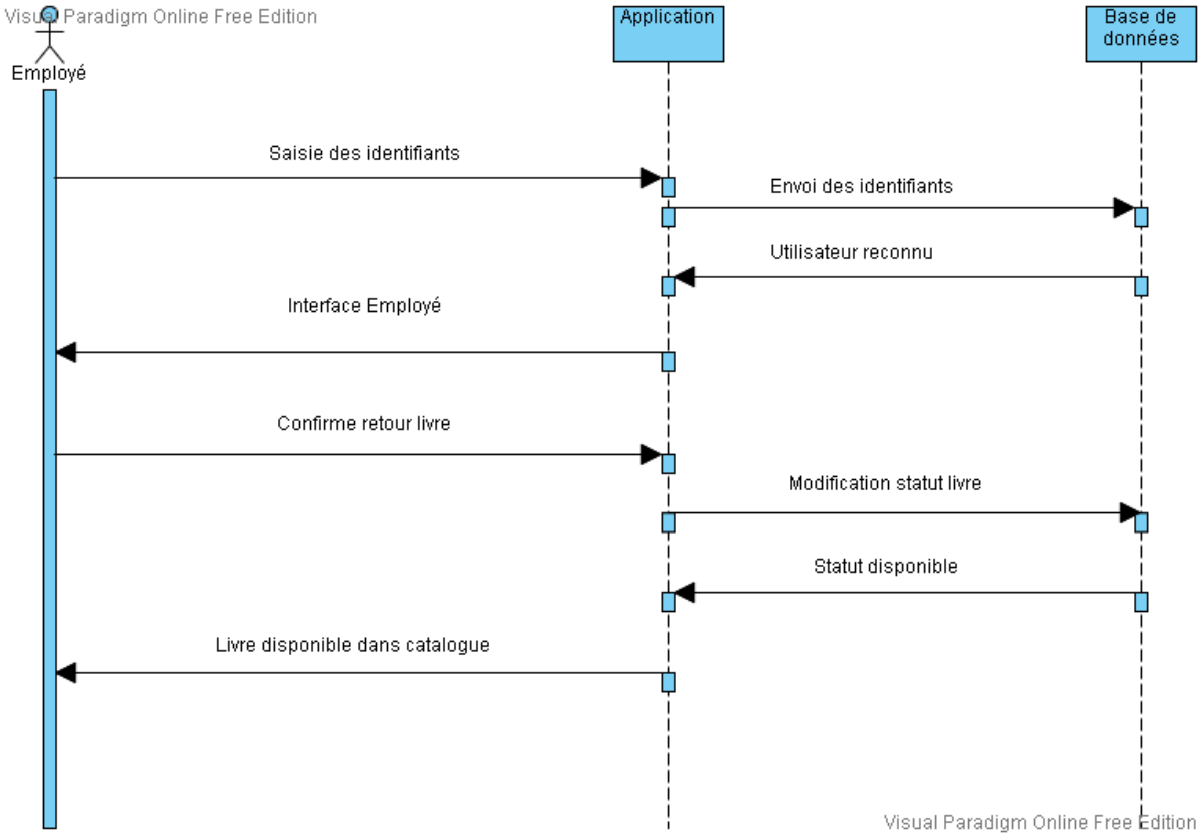


## C. Diagrammes de Séquence

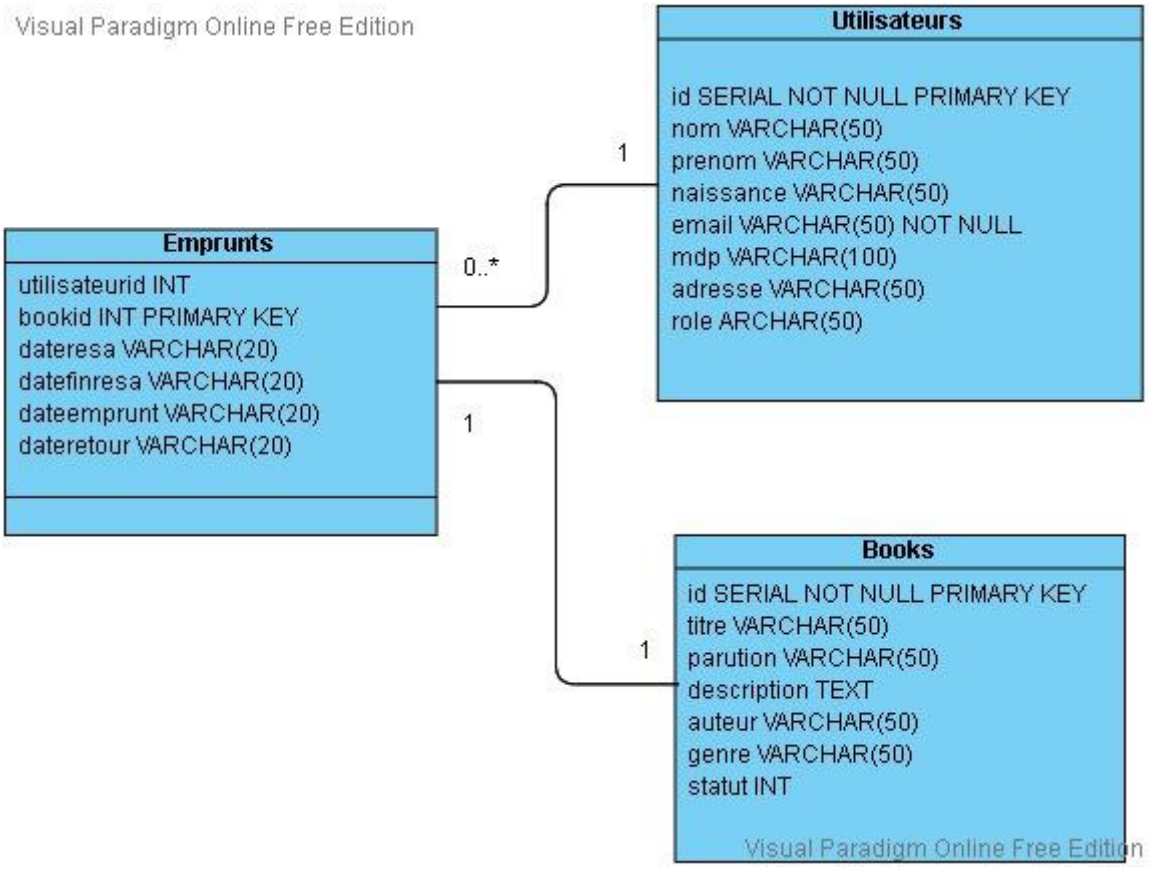
### 01. Emprunter un livre



02. Rendre un livre



## D. Diagramme de Classe




## E. Mesures de sécurité

Concernant la sécurité j'ai opté pour plusieurs méthodes :

- les mots de passe sont 'hashés' avec la méthode PASSWORD_BCRYPT afin qu'ils n'apparaissent pas en clair même dans la base de données.

- Pour l'accès à la base de données, afin d'éviter des injections sql j'ai utilisé des requêtes préparées.

- Concernant les failles XSS, lorsqu'une personne s'inscrit, j'ai utilisé la méthode :

(htmlspecialchars($_POST['prenom'], ENT_QUOTES))

Ceci permet d'éviter des caractères spéciaux qui seraient considérés comme du langage html ou JavaScript par exemple.

- Lors de la connexion d'un utilisateur un token est créé de cette manière :

$_SESSION['token'] = md5(time() * rand(142, 628));

Afin d'être sûr qu'un token est unique je multiplie l'heure timestamp de connexion par un chiffre aléatoire et le tout est passé dans la fonction md5().

Cette fonction concatène les infos données puis utilise un algorithme et renvoie une chaine de 32 caractères.

A chaque chargement de page PHP vérifie que le token est toujours le même que sur la page précédente.

- De plus chaque type d'utilisateur se voit attribué un rôle différent : 'En attente' tant que le compte n'est pas confirmé, INSCR pour un habitant dont le compte est confirmé, EMPL pour un employé et ADMIN pour un administrateur.

A chaque accès d'une page différente, PHP vérifie, en plus du token, que l'utilisateur a bien le rôle nécessaire. Quant au rôle 'En attente', il ne peut même pas se connecter.