

Documentation techniques

Charte graphique du site :

Le design étant réduit au minimum sur ce projet, j'ai réalisé une approche simple avec la police, la couleur, le logo et les boutons.

Police Lato pour les titres et Roboto pour le corps du texte.

Une couleur noire pour la Navbar et blanc pour le corps.

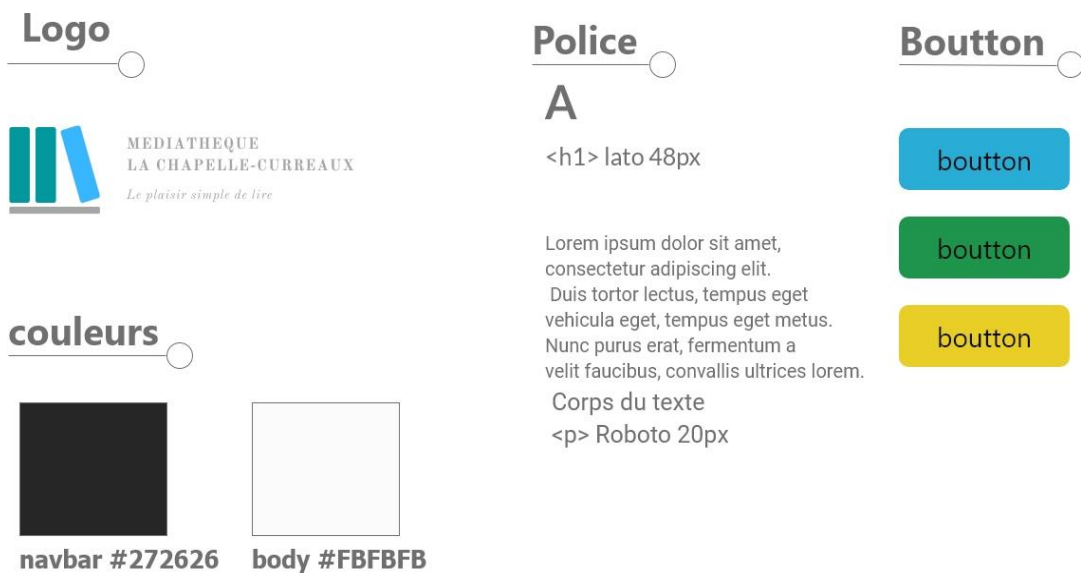
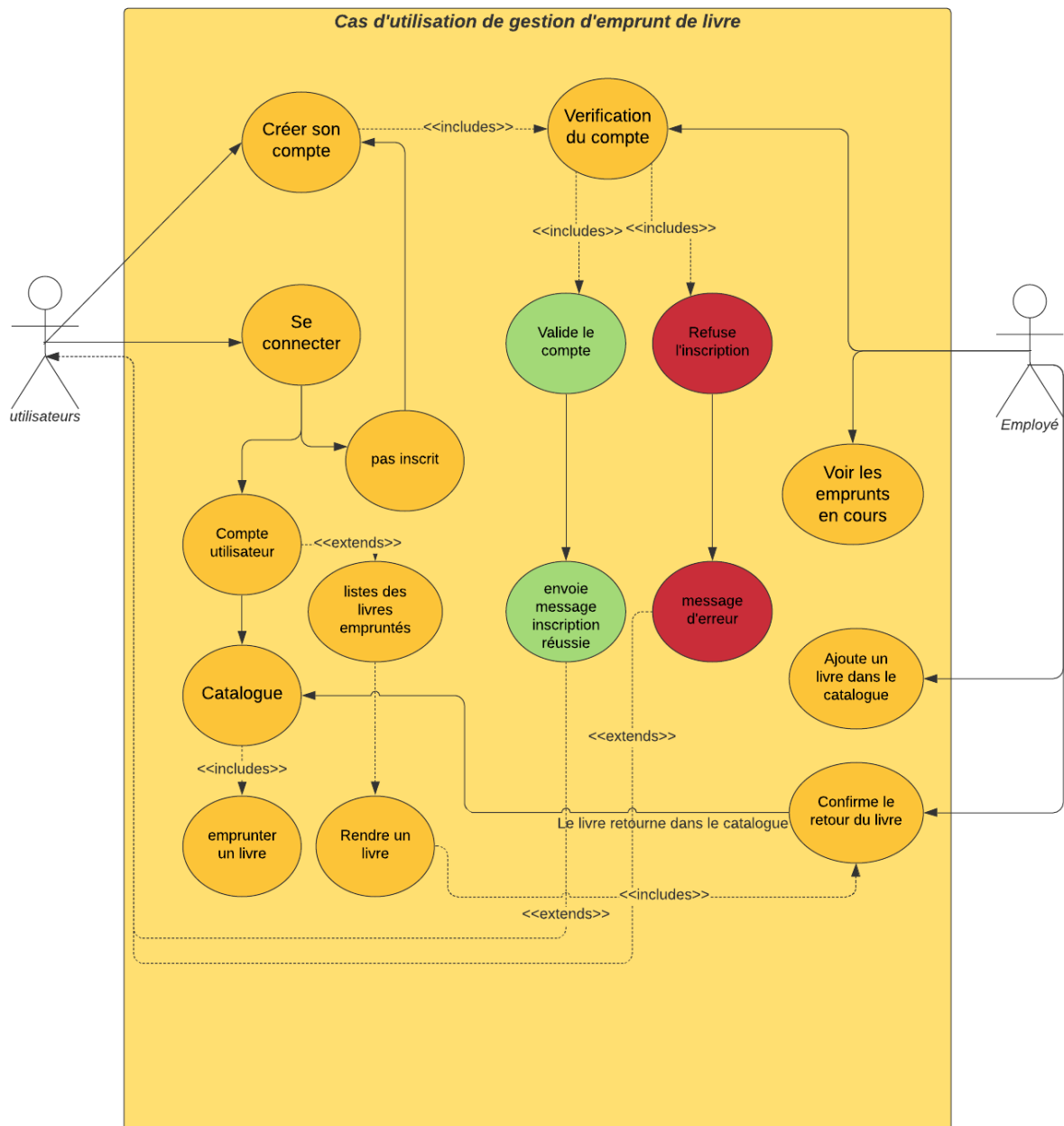


Diagramme de cas d'utilisation :

J'ai réalisé le diagramme de cas d'utilisation en suivant l'ordre des fonctionnalités désirées afin de clarifier le rôle de chacun des intervenants.

LE PROJET Médiathèque

Bruno Richard - octobre 2021



Le diagramme de classes :

J'ai réalisé le diagramme de classe en définissant les différentes entités qui composent le projet :

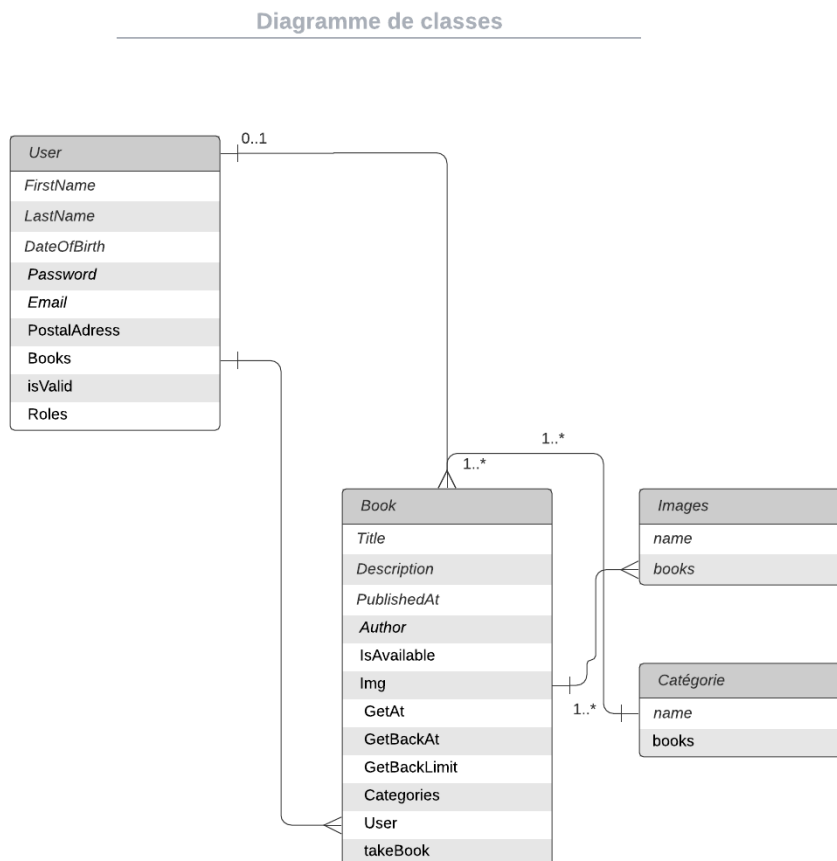
- Les utilisateurs
- Les livres
- Les catégories

Puis, au fur et à mesure de la réalisation d'autres contraintes se sont imposées.

Les livres comportant des images pour les couvertures nécessitaient une table :

- Images

En établissant les cardinalités, j'ai défini les relations entre les tables. Dans Symfony cela se traduit par les types `ManyToOne`, `OneToMany`...



Un livre appartient à une catégorie => une catégorie comporte plusieurs livres

Un livre peut avoir une ou plusieurs images => une image appartient à un livre

Un livre est emprunté par un utilisateur => un utilisateur peut emprunter un ou plusieurs livres.

Le diagramme de séquences :

Pour bien comprendre le déroulement dans le temps d'une fonctionnalité, j'ai utilisé un diagramme de séquences.

Pour :

- L'emprunt d'un livre
- Le retour du livre

Diagramme de sequence emprunter un livre

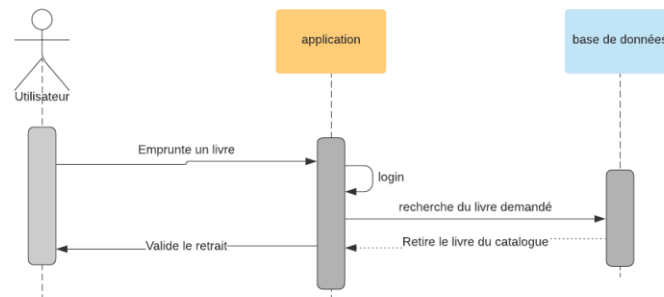
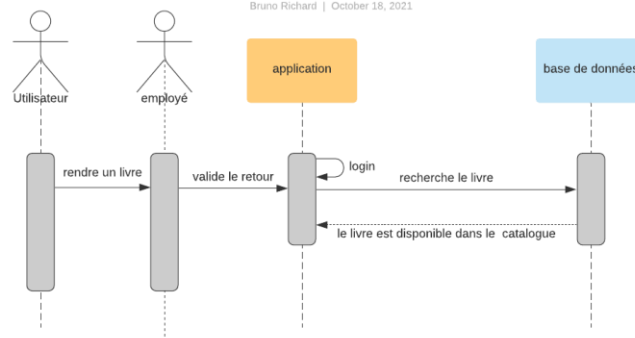


Diagramme de sequence rendre un livre

Bruno Richard | October 18, 2021



La sécurité dans le projet :

La première menace et la plus connue est celle des injections.

- Pertes de données
- Vol de données
- Injection de données

L'injection «SQL » est liée à la base de données. La première règle est « Never trust user input ». Dans Symfony, la gestion des inputs passe par les formTypes. La sécurité étant déjà induite dans Symfony, comme d'autres Framework, en définissant les inputs avec des types de champs spécifiques (Email, password(nombre de caractères minimum, required...)) j'ai choisi de sécuriser mes formulaires de cette façon. J'aurais pu aussi utiliser le validator constraint de Symfony au niveau des propriétés des entités.

J'ai sécurisé les routes en conditionnant les boutons et les menus d'accès pour les restreindre à un type d'utilisateur (Employé, inscrit).

Pour le stockage des mots de passe, j'ai utilisé le système de hash de Symfony qui encode le mot de passe avant de l'envoyer dans la base de données.

Maquettage :

Comme conseillé dans l'énoncé du sujet, j'ai préconisé une approche minimaliste du design :

Voici quelques maquettes réalisées avec AdobeXD.

Maquette du formulaire de connexion. Le formulaire est intitulé "Connexion". Il contient deux champs de saisie : "Email" et "Mot de passe". En dessous du champ "Mot de passe", il y a un lien hypertexte "créer un compte" en bleu. En bas du formulaire, il y a un bouton vert avec le texte "Connexion".



Créer un compte

Email

Mot de passe

Adresse postale

Date naissance

Nom

Prénom

Sauvegarder