

Cahier des Charges Fonctionnel et Technique - Application de Gestion de Collections

1. Introduction

Ce document présente les spécifications fonctionnelles et techniques de l'application de gestion de collections. Il sert de référence pour le développement, la validation et la maintenance du projet. L'objectif est de fournir aux utilisateurs une plateforme intuitive pour organiser, consulter et gérer leurs collections d'objets variés (Livres, Vinyles, Jeux Vidéo).

2. Objectifs du Projet

- * Centraliser la gestion : Offrir un point unique pour gérer différents types d'objets de collection.
- * Faciliter l'organisation : Permettre aux utilisateurs de classer et de retrouver facilement leurs objets grâce à des informations détaillées.
- * Améliorer l'expérience utilisateur : Fournir une interface simple, réactive et agréable à utiliser.
- * Démontrer des compétences techniques : Mettre en œuvre les bonnes pratiques de développement web avec le framework Symfony.

3. Fonctionnalités (Spécifications Fonctionnelles)

3.1. Gestion des Utilisateurs

- * Inscription : Un utilisateur peut créer un nouveau compte.
- * Connexion/Déconnexion : Un utilisateur peut se connecter et se déconnecter de l'application.
- * Authentification : Les utilisateurs sont authentifiés via un système de login/mot de passe.
- * Rôles : Distinction entre les rôles (ex: ROLE_USER, ROLE_ADMIN pour la gestion interne ou future).

3.2. Gestion des Collections

*Ajout d'objet : Un utilisateur connecté peut ajouter de nouveaux objets à sa collection, en spécifiant le type (Livre, Vinyle, Jeu Vidéo) et les détails associés (titre/nom, description, date d'ajout, statut, catégorie, tags, emplacement, et attributs spécifiques au type).

*Modification d'objet : Un utilisateur peut modifier les détails de n'importe quel objet de sa collection.

*Suppression d'objet : Un utilisateur peut supprimer un objet de sa collection.

*Détail d'objet : Affichage d'une page dédiée pour chaque objet avec toutes ses informations.

3.3. Affichage et Navigation des Collections

"Ma Collection" : Affichage de tous les objets appartenant à l'utilisateur connecté.

* Toutes les collections" : ** Affichage de tous les objets de l'application (potentiellement de tous les utilisateurs), si la visibilité publique est activée.

* Recherche : Recherche d'objets par mots-clés (ex: titre, auteur, artiste, plateforme).

* Filtrage : Filtrage des objets par type (Livre, Vinyle, Jeu Vidéo), statut, catégorie, emplacement, tags.

* Tri : Tri des objets par nom, date d'ajout.

*Compteurs / Statistiques : Affichage de chiffres clés (ex: nombre total d'objets, nombre par type).

* "Derniers ajouts" : Affichage des objets récemment ajoutés par l'utilisateur.

**"Objet aléatoire" : Permettre la découverte d'un objet au hasard dans la collection.

3.4. API RESTful (pour la sérialisation des données)

*Affichage JSON de toutes les collections : Point d'API pour récupérer toutes les collections au format JSON.

* Affichage JSON d'un objet spécifique : Point d'API pour récupérer les détails d'un objet par son ID au format JSON.

* (Future) : Points d'API pour l'ajout, la modification et la suppression via JSON.

4. Spécifications Techniques

4.1. Architecture

*Framework : Symfony 7

*Base de données : Relationnelle (MySQL, PostgreSQL, SQLite via Doctrine ORM).

*ORM : Doctrine ORM pour la gestion de la persistance des données.

4.2. Technologies Utilisées

*Backend : PHP (version compatible avec Symfony 6/7).

*Frontend :

*Templating : Twig.

*CSS : Tailwind CSS pour un design utilitaire et responsive.

* JavaScript : Vanilla JS ou Stimulus/Turbo (Symfony UX) si applicable.

*Bundler d'assets : Webpack Encore ou Vite (via PentatrickViteBundle).

4.3. Gestion de Projet

*Contrôle de version : Git.

*Dépendances PHP : Composer.

*Dépendances JS : npm ou Yarn.

5. Contraintes et Limitations

* Sécurité : Mise en place d'un système d'authentification et d'autorisation (Voters Symfony) pour la gestion des accès aux objets (seul le propriétaire peut modifier/supprimer son objet).

*Responsive Design : L'application doit être utilisable sur différentes tailles d'écran (mobile, tablette, desktop).

* Performance : Optimisation des requêtes Doctrine pour une navigation fluide.

*Délais : Le projet doit être fonctionnel pour la date de soutenance.

6. Évolutions Futures envisagées (Hors Scope de la soutenance)

*Système de Forum / Discussion : Permettre aux utilisateurs de créer des sujets de discussion, poster des messages et interagir autour de leurs collections.

*Système de Recommandation : Suggérer des objets aux utilisateurs basés sur leur collection existante.

*Import/Export de données : Fonctionnalités pour importer/exporter des collections (CSV, JSON).

* Notifications : Alertes pour les nouvelles réponses de forum, les ajouts d'amis, etc.

* Gestion d'images : Ajouter des photos aux objets de collection.

* Recherche avancée / Facettes : Amélioration des critères de recherche et des filtres.