



Bookswap

Application de Gestion de Bibliothèque & Échanges de Livres

Documentation Technique Complète

Auteur	Youma DRAME
Formation	BTS SIO SLAM — École IRIS Paris
Période	Février 2026
Technologies	PHP, Firebase (NoSQL), HTML5/CSS3, JavaScript, UML
Type	Application web — Échange de livres entre particuliers
Portfolio	youmadrame.com

1. Contexte & Objectifs

Bookswap est une plateforme communautaire permettant l'échange de livres entre particuliers. Les utilisateurs peuvent partager leurs livres, rechercher des ouvrages disponibles près de chez eux, et organiser des échanges avec d'autres membres.

1.1 Problématique

Favoriser le partage et la réutilisation des livres physiques en créant une plateforme simple, intuitive et accessible, avec un système de notation pour instaurer la confiance entre membres.

1.2 Objectifs

Fonctionnalités cibles	
▶	Catalogue de livres disponibles avec recherche multicritère
▶	Profils utilisateurs avec statistiques d'échanges
▶	Système de notation et avis (1 à 5 étoiles)
▶	Gestion des disponibilités : Disponible / Indisponible
▶	Messagerie entre membres pour organiser les échanges
▶	Localisation des livres par ville / arrondissement
▶	Firebase pour synchronisation temps réel

2. Architecture Technique

Front-end	HTML5, CSS3, JavaScript — Design responsive
Back-end	PHP — Logique métier et routage

Base de données	Firestore Realtime Database (NoSQL)
Modélisation	UML — Diagrammes de classes et cas d'utilisation
Versioning	Git / GitHub

2.1 Choix de Firebase

Firebase a été choisi pour sa capacité de synchronisation en temps réel, idéale pour afficher instantanément les nouvelles disponibilités et messages sans rechargement de page. C'est également une introduction aux bases de données NoSQL.

3. Modélisation UML

3.1 Diagramme des cas d'utilisation

Acteurs & cas d'utilisation	
▶	Utilisateur non connecté : consulter le catalogue, rechercher un livre
▶	Utilisateur connecté : ajouter un livre, proposer un échange, envoyer un message, noter un échange
▶	Administrateur : modérer les annonces, gérer les utilisateurs

3.2 Structure des données Firebase

Collections Firebase	
▶	users/{uid} : nom, email, localisation, note_moyenne, nb_echanges
▶	livres/{id} : titre, auteur, genre, description, id_proprietaire, disponible, localisation
▶	echanges/{id} : id_livre, id_demandeur, id_proprietaire, statut, date
▶	messages/{id} : id_expediteur, id_destinataire, contenu, timestamp
▶	notations/{id} : id_echange, note, commentaire, date

4. Fonctionnalités Développées

4.1 Catalogue des livres

Interface de recherche avec filtres :

- Recherche par titre, auteur, genre
- Filtre par localisation (ville, arrondissement)
- Badges de disponibilité (Disponible / Indisponible)
- Affichage de la note du propriétaire et du délai de réponse

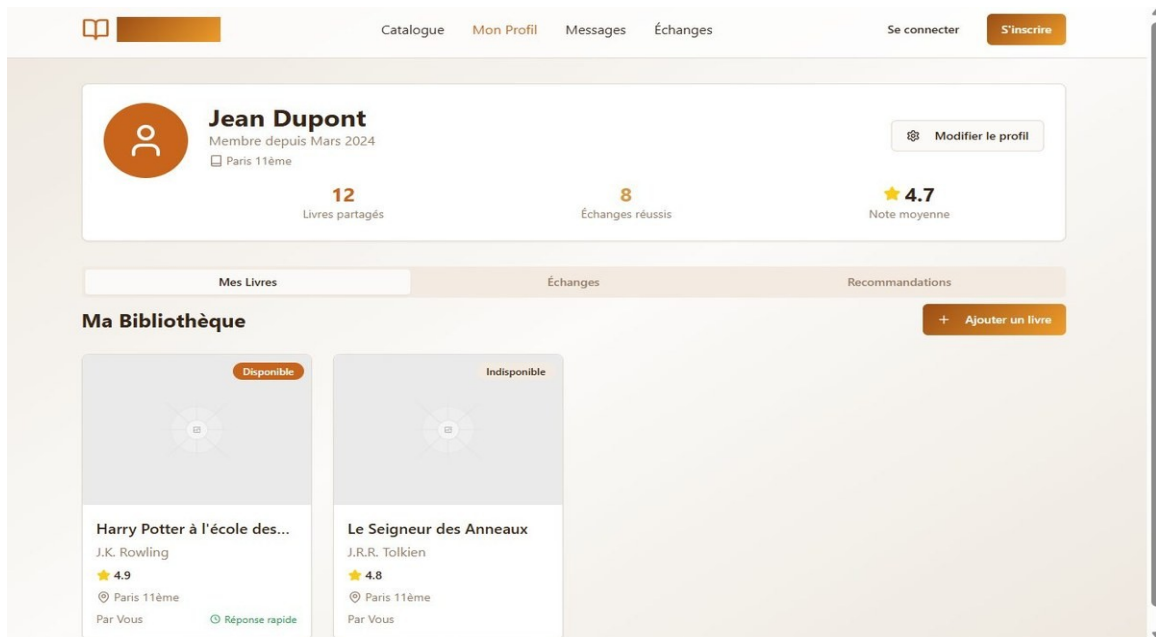


Fig. 1 — Catalogue des livres disponibles à l'échange

4.2 Profil utilisateur

Chaque membre dispose d'un profil complet :

- Photo de profil et localisation
- Statistiques : livres partagés, échanges réussis, note moyenne
- Bibliothèque personnelle avec statuts de disponibilité
- Historique des échanges et recommandations reçues

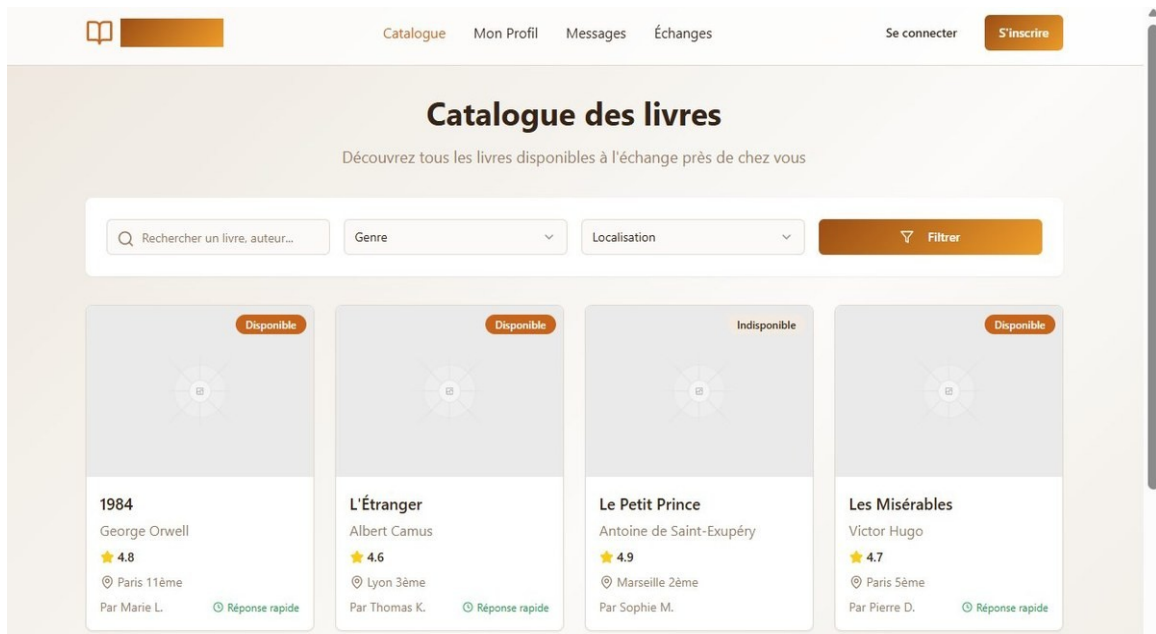


Fig. 2 — Profil utilisateur avec bibliothèque et statistiques

5. Tests Réalisés

Test 1	Inscription & connexion utilisateur → Session créée ✓
Test 2	Ajout d'un livre → Apparition dans le catalogue ✓
Test 3	Recherche par titre → Résultats filtrés correctement ✓
Test 4	Changement de disponibilité → Mise à jour temps réel Firebase ✓
Test 5	Notation après échange → Moyenne recalculée ✓
Test 6	Affichage mobile → Interface responsive validée ✓

6. Bilan & Compétences Mobilisées

Compétences BTS SIO mobilisées	
▶	Travailler en mode projet — conception UML, découpage fonctionnel
▶	Mettre à disposition un service informatique — plateforme d'échange fonctionnelle
▶	Développer la présence en ligne — interface web accessible et responsive

Ce projet m'a permis de découvrir les bases de données NoSQL avec Firebase, la modélisation UML appliquée à un projet réel, et la conception d'une interface communautaire avec système de notation.