

# [Livre] Présentation générale

## Contexte

L'association "Quartier Solidaire" gère une bibliothèque communautaire qui met à disposition des ouvrages pour les habitants du quartier. Actuellement, la gestion se fait manuellement avec des fiches papier, ce qui engendre des difficultés :

- Difficulté à connaître la disponibilité des livres en temps réel
- Gestion fastidieuse des emprunts et des retards
- Pas de système de réservation
- Impossibilité pour les adhérents de consulter le catalogue à distance
- Pas de statistiques sur les emprunts

Face à ces problématiques, l'association souhaite moderniser sa bibliothèque en développant une application web accessible et éco-conçue.

**Durée** : 2 semaines (10 jours ouvrés)

**Modalités** : Équipes de 2 à 4 personnes

**Méthodologie** : Gestion de projet Agile

## [Cible] Objectifs du projet

Développer **BookHub**, une plateforme web complète de gestion de bibliothèque permettant :

### Pour les lecteurs

- Consulter le catalogue de livres en ligne
- Rechercher des livres par titre, auteur ou catégorie
- Emprunter des livres disponibles
- Réserver des livres déjà empruntés
- Consulter l'historique de ses emprunts
- Noter et commenter les livres lus

### Pour les bibliothécaires

- Gérer le catalogue (ajouter, modifier, supprimer des livres)
- Valider les emprunts et les retours

- Gérer les retards
- Accéder aux statistiques d'emprunt

## Pour les administrateurs

- Gérer les utilisateurs et leurs rôles
- Accéder à une vue d'ensemble de l'activité
- Gérer les paramètres de la bibliothèque

# [Utilisateurs] Utilisateurs cibles

## Profils utilisateurs

Ci-dessous les différents types de profil possible au sein de l'application.

### Le Lecteur (adhérent)

**Exemple** : Marie, 32 ans, professeure, passionnée de lecture

- **Besoins** : Consulter le catalogue depuis chez elle, emprunter facilement, être notifiée des retards
- **Fréquence d'utilisation** : Plusieurs fois par mois

### Le Bibliothécaire

**Exemple** : Ahmed, 45 ans, responsable de la bibliothèque

- **Besoins** : Gérer le catalogue, valider les emprunts/retours, suivre les retards
- **Fréquence d'utilisation** : Quotidienne

### L'Administrateur

**Exemple** : Sophie, 50 ans, présidente de l'association

- **Besoins** : Vue d'ensemble, gestion des utilisateurs, statistiques
- **Fréquence d'utilisation** : Hebdomadaire

# [Valeur] Valeur ajoutée du projet

Le projet a pour but d'améliorer le métier de la bibliothèque communautaire.

Les objectifs sont multiples.

## Bénéfices pour l'association

- [OK] **Gain de temps** : Automatisation des processus de gestion
- [OK] **Meilleure visibilité** : Statistiques et tableaux de bord
- [OK] **Réduction du papier** : Démarche éco-responsable
- [OK] **Amélioration du service** : Disponibilité 24/7 du catalogue en ligne

## Bénéfices pour les adhérents

- [OK] **Accessibilité** : Consultation du catalogue à distance
- [OK] **Confort** : Réservations en ligne, historique
- [OK] **Découverte** : Système de notation et recommandations

# [Outils] Technologies utilisées

La stack technique conseillée est celle vu lors de la formation.

## Stack technique

### Backend

- **Langage** : Java 17+
- **Framework** : Spring Boot 3.x
- **Sécurité** : Spring Security avec JWT
- **Persistance** : Spring Data JPA / Hibernate
- **Base de données** : Microsoft SQL Server
- **Documentation API** : Swagger/OpenAPI

### Frontend

- **Framework** : Angular 17+
- **Langage** : TypeScript
- **Styling** : CSS
- **Communication** : HttpClient (REST API)

## DevOps & Outils

- **Versioning** : Git (GitHub/GitLab)
- **Gestion de projet** : Scrum (Trello/Jira/GitHub Projects)
- **Conteneurisation** : Docker (optionnel)
- **Tests** : JUnit (backend), Jasmine/Karma (frontend)

# [Liste] Fonctionnalités métiers

Voici les différentes fonctionnalités importantes pour l'application BookHub.

Elles constituent la base des cas d'utilisation.

Elles sont issues d'une analyse des besoins faites avec la bibliothèque communautaire.

## 1. Gestion des utilisateurs et authentification

### Inscription

- Formulaire avec validation (email, mot de passe, nom, prénom)
- Validation de l'adresse email
- Hachage sécurisé des mots de passe (BCrypt)

### Connexion

- Authentification par email/mot de passe
- Génération d'un token JWT
- Stockage sécurisé du token côté client

### Gestion du profil

- Consultation et modification des informations personnelles
- Changement de mot de passe
- Suppression de compte (avec confirmation)

### Rôles et permissions

- **USER** : Consultation, emprunt, réservation, notation
- **LIBRARIAN** : + Gestion du catalogue, validation des emprunts
- **ADMIN** : + Gestion des utilisateurs, accès complet

## 2. Catalogue de livres

### Consultation

- Liste paginée des livres avec aperçu (couverture, titre, auteur, disponibilité)
- Fiche détaillée d'un livre (description, catégorie, notes, commentaires)
- Indication visuelle de la disponibilité

### Recherche et filtres

- Recherche textuelle (titre, auteur, ISBN)
- Filtres par catégorie
- Filtre par disponibilité
- Tri (alphabétique, note, date d'ajout)

### Gestion (bibliothécaire)

- Ajout d'un nouveau livre
- Modification des informations
- Suppression d'un livre (avec vérification des emprunts en cours)

## 3. Système d'emprunt et de réservation

### Emprunter un livre

- Bouton "Emprunter" sur la fiche livre
- **Vérifications :**
  - Livre disponible
  - Maximum 3 emprunts simultanés par utilisateur
  - Pas de retard en cours
- **Durée d'emprunt :** 14 jours

### Mes emprunts

- Liste des emprunts en cours
- Date de retour prévue
- Alerte si retard
- Historique complet des emprunts passés

## **Retour d'un livre (bibliothécaire)**

- Enregistrement du retour
- Calcul automatique du retard (si applicable)
- Libération d'un exemplaire

## **Réserver un livre**

- Si un livre est emprunté, possibilité de le réserver
- File d'attente

# **4. Système de notation et commentaires**

## **Noter un livre**

- Note de 1 à 5 étoiles
- Possibilité de modifier sa note
- Calcul de la note moyenne

## **Commenter un livre**

- Avis textuel (max 1000 caractères)
- Date de publication
- Nom de l'auteur du commentaire (pseudonyme ou nom réel selon préférences)

## **Modération (bibliothécaire)**

- Consultation des commentaires
- Suppression des commentaires inappropriés

# **5. Tableaux de bord**

## **Dashboard Lecteur**

- Mes emprunts en cours
- Mes réservations
- Livres lus récemment

## **Dashboard Bibliothécaire**

- Nombre total de livres
- Nombre d'emprunts actifs
- Liste des retards
- Livres les plus empruntés
- Statistiques mensuelles

## Dashboard Administrateur

- Nombre d'utilisateurs actifs
- Vue d'ensemble de l'activité
- Statistiques complètes (graphiques)
- Évolution dans le temps

# Fonctionnalités techniques

Ces fonctionnalités sont essentielles pour la sécurité et l'accessibilité des utilisateurs ainsi que pour l'écologie.

## [Securite] Sécurité (ANSSI / OWASP Top 10)

### Authentification et autorisation

- Mots de passe hachés (BCrypt, coût  $\geq 12$ )
- Token JWT signé avec secret fort
- Expiration des tokens (24h recommandé)
- Contrôle d'accès basé sur les rôles (RBAC)

### Protection contre les attaques

- **Injection SQL** : Requêtes paramétrées (JPA)
- **XSS** : Sanitization des entrées (Angular DomSanitizer)
- **CSRF** : Tokens CSRF (Spring Security)
- **CORS** : Configuration restrictive
- **Validation des entrées** : Côté client ET serveur

### Données sensibles

- Pas de données sensibles dans les URLs
- HTTPS en production

- Pas de mots de passe en clair dans les logs
- Respect du RGPD (gestion du consentement, droit à l'oubli)

## [Eco] Éco-conception

### Frontend

- Images optimisées (compression, formats modernes WebP)
- Lazy loading des images
- Minification du code en production
- Limitation des requêtes HTTP
- Pas de bibliothèques inutiles

### Backend

- Requêtes SQL optimisées (pas de N+1, index)
- Pagination systématique
- Cache des données fréquentes (optionnel)
- Limitation de la charge serveur

## [Performance] Performance

### Temps de réponse

- **Page d'accueil** : < 2 secondes
- **Recherche** : < 1 seconde
- **Appels API** : < 500ms (hors traitement lourd)

### Optimisations

- Index sur les colonnes de recherche fréquentes
- Eager/Lazy loading approprié (JPA)

## [Ordinateur] Compatibilité

### Navigateurs



- Chrome (2 dernières versions)
- Firefox (2 dernières versions)
- Edge (2 dernières versions)

## Appareils

- Desktop (1920x1080, 1366x768)
- Tablette (iPad, Android tablets)
- Mobile (iPhone, Android smartphones)
- **Responsive design obligatoire**

# [Calendrier] Planning et organisation

Voici un planning proposé.

L'équipe est libre d'adapter en fonction du contexte.

## Durée

2 semaines (10 jours ouvrés)

## Méthodologie

### Scrum (Agile)

- **Sprint 0** : Conception (J1-J2)
- **Sprint 1** : MVP (J3-J5)
- **Sprint 2** : Complétion et finalisations (J6-J9)
- **Clôture** : Documentation et présentation (J10)

## Rituels Agile

- **Sprint Planning** : Début de chaque sprint
- **Daily Stand-up** : Quotidien (15 min)
- **Sprint Review** : Fin de chaque sprint
- **Sprint Retrospective** : Après chaque review

# [Education] Livrables attendus

Les livrables attendus sont multiples et listés ci-dessous.

## 1. Documentation de conception

- User stories avec critères d'acceptation
- Diagrammes UML (cas d'utilisation, classes, séquence)
- Maquettes (wireframes)
- Modèle de données (MCD, MLD, MPD)
- Schéma d'architecture logicielle

## 2. Code source

- Backend Spring Boot (structuré, commenté, testé)
- Frontend Angular (structuré, commenté, testé)
- Scripts SQL (création tables + données de test)
- Versionning Git (commits réguliers, branches)

## 3. Documentation technique

- README complet (installation, configuration, lancement)
- Documentation API (Swagger)

## 4. Tests

- Tests unitaires backend (JUnit, couverture  $\geq 20\%$ )
- Tests unitaires frontend (Jasmine/Karma, couverture  $\geq 20\%$ )

## 5. Présentation finale

- Diaporama (20 min + 10 min questions)
- Démonstration en direct
- Tous les membres de l'équipe interviennent

**[Fusee] Bon développement et bon courage !**

Rappelez-vous : l'objectif est d'apprendre en pratiquant. N'hésitez pas à expérimenter, à faire des erreurs et à solliciter de l'aide quand nécessaire. Le travail en équipe et la communication sont essentiels à la réussite de ce projet !