

# **Cahier des charges**

## **Conception application de gestion de bibliothèque numérique**

**Biblionest**

Ludmilla ZEPHIR

2023 - 2025

**Web***i***tech**  
nextgroup

# SOMMAIRE

<b>1. Introduction .....</b>	<b>3</b>
<b>2. Contexte .....</b>	<b>3</b>
<b>3. Objectif .....</b>	<b>3</b>
<b>4. Public ciblé.....</b>	<b>4</b>
<b>5. Besoins principaux .....</b>	<b>5</b>
<b>6. Besoins Fonctionnels.....</b>	<b>5</b>
<b>7. Références techniques.....</b>	<b>6</b>
<b>8. Hébergement et nom de domaine .....</b>	<b>6</b>
<b>9. Base de Données.....</b>	<b>7</b>
<b>10. Cahier des charges fonctionnels.....</b>	<b>7</b>
<b>11. Fonctionnalités essentielles .....</b>	<b>8</b>
<b>12. Méthodologie et planification .....</b>	<b>8</b>
<b>13. Délais .....</b>	<b>9</b>

# 1. Introduction

Ce cahier des charges va permettre de définir avec précision les objectifs, les fonctionnalités, les contraintes techniques et les attentes liées au projet. Ce document sert de référence commune entre les parties prenantes (développeurs, clients, utilisateurs finaux) pour garantir que l'application réponde aux besoins définis.

Dans ce contexte, ce cahier des charges vise à structurer le développement de l'application de gestion de bibliothèque, en décrivant ses fonctionnalités principales, son architecture technique et les exigences en matière d'interface utilisateur et de sécurité.

## 2. Contexte

Les bibliothèques, qu'elles soient publiques ou privées, nécessitent une gestion rigoureuse des ouvrages et des prêts. L'utilisation d'un système informatisé permet d'améliorer l'organisation, d'éviter les pertes d'ouvrages et de faciliter l'accès aux livres pour les utilisateurs.

Dans ce cadre, le projet consiste à développer une application web qui simplifie la gestion quotidienne des livres, des emprunts et des pénalités au sein d'une bibliothèque. Cet outil vise à offrir une **solution efficace** et **intuitive** pour les employés de la bibliothèque, leur permettant d'optimiser leurs tâches et de garantir un suivi efficace des prêts.

## 3. Objectif

L'objectif principal du projet est de fournir un tableau de bord interactif facilitant la gestion des opérations suivantes :

- Suivi des livres disponibles et empruntés.
- Gestion des catégories de livres.
- Enregistrement et suivi des emprunts et retours.
- Application automatique des pénalités en cas de retard.
- Fourniture d'un tableau de bord détaillé avec des statistiques en temps réel.
- Accès restreint aux employés grâce à un système d'authentification sécurisé

## 4. Public ciblé

L'application est exclusivement destinée aux employés de la bibliothèque. Elle leur permet de gérer efficacement les ouvrages, les emprunts et les pénalités, tout en offrant un accès structuré et sécurisé aux données de la bibliothèque. Les utilisateurs auront des niveaux d'accès différents en fonction de leur rôle, garantissant ainsi un contrôle optimal des informations et des opérations.

## 5. Charte graphique

Couleurs des boutons :

#2563EB

#10B981

Couleurs du logo

#210955

#8D39C7

Typographie :

Titre : Arial

Paragrophes : Arial, serif

Logo Desktop :



## 6. Besoins principaux

L'application permettra de :

- Ajouter, modifier, supprimer et afficher les livres de la bibliothèque.
- Classer les livres par catégories.
- Gérer les emprunts et retours des livres.
- Automatiser l'application des pénalités en cas de retard.
- Fournir un tableau de bord avec des statistiques en temps réel.

## 7. Besoins Fonctionnels

### 1.2 Gestion des Livres

- Ajouter un nouveau livre avec titre, auteur, année de publication, ISBN et catégorie.
- Modifier ou supprimer un livre.
- Lister les livres disponibles.
- Rechercher un livre par titre, auteur, catégorie ou ISBN.

### 1.2 Gestion des Catégories

- Ajouter, modifier et supprimer des catégories de livres.
- Associer un livre à une catégorie.
- Filtrer les livres par catégorie.

### 1.3 Gestion des Emprunts

- Enregistrer un nouvel emprunt avec identité de l'emprunteur et livre emprunté.
- Suivre la date de retour prévue.
- Filtrer les emprunts en cours, en retard, ou retournés.

### 1.4 Gestion des Pénalités

- Détecter automatiquement les livres en retard.
- Appliquer une pénalité si l'emprunt dépasse la date limite.
- Marquer une pénalité comme payée.

### 1.5 Gestion des Utilisateurs

- Authentification via email et mot de passe.
- Gestion des permissions (Admin, Employé).
- Gestion des profils utilisateurs.

### 1.6 Tableau de Bord et Statistiques

- Affichage des statistiques : nombre de livres, emprunts en cours, retards.
- Graphiques sur l'activité de la bibliothèque.
- Export des données en PDF.

## 8. Références techniques

- **Backend** : Laravel 10 (PHP)
- **Frontend** : Blade + TailwindCSS
- **Base de données** : MySQL
- **Authentification** : Laravel Breeze
- **Automatisation** : Laravel Scheduler

## 9. Hébergement et nom de domaine

- **Hébergement** : O2switch
- **Nom de domaine** : <https://app.biblionest.fr>

## 10. Base de Données – MySQL

Le choix d'une base de données relationnelle est particulièrement adapté à ce projet, car elle permet une organisation structurée des données avec des relations claires entre les livres, auteurs et utilisateurs. Elle excelle dans la gestion des requêtes complexes via SQL, ce qui est essentiel pour un système de recherche de livres efficace. La garantie d'intégrité des données empêche les incohérences dans le catalogue, tandis que la gestion des accès simultanés permet à plusieurs utilisateurs d'interagir avec le système en même temps. De plus, elle offre une excellente sécurité et peut évoluer facilement selon les besoins futurs de la bibliothèque.

### 1.1 Tables principales :

- **books** (id, title, author, category\_id, isbn, published\_year, copies\_available)
- **categories** (id, name)
- **users** (id, name, email, password)
- **borrowings** (id, user\_id, book\_id, borrowed\_at, due\_date, returned\_at)
- **penalties** (id, borrowing\_id, amount, paid\_at)

### 1.2 Architecture MVC (Laravel)

- **Modèles** : Book, Category, Borrowing, Penalty, User
- **Contrôleurs** : BookController, BorrowingController, PenaltyController, UserController
- **Vues Blade** : pages pour les livres, emprunts, pénalités, dashboard

## 11. Cahier des charges fonctionnels

- **Tableau de bord** : Vue globale avec statistiques.
- **Page Livres** : Liste des livres avec options de recherche et filtres.
- **Page Emprunts** : Suivi des emprunts avec statut (en cours, en retard, retourné).
- **Page Pénalités** : Gestion des sanctions appliquées aux emprunteurs.
- **Page Catégories** : Gestion des genres de livres.
- **Page Profil** : Gestion du compte utilisateur.

## 12. Fonctionnalités essentielles

Tâche planifiée Laravel Scheduler pour :

- Détecter les emprunts en retard.
- Générer des pénalités automatiquement.

Export des données PDF

Application mobile responsive : Compatible avec tous les appareils (PC, tablette, mobile).

## 13. Méthodologie et Planification

Pour organiser et suivre le développement de l'application Biblionest, j'ai utilisé trello pour créer un tableau de tâches ainsi qu'une chronologie. Le Kanban est une méthode de gestion visuelle qui permet de suivre les progrès des tâches en temps réel et de gérer les flux de travail de manière efficace. Chaque tâche est représentée par une carte et les cartes sont déplacées à travers différentes colonnes représentant les étapes de travail. Cela m'a permis de visualiser facilement l'état d'avancement de chaque tâche et de m'adapter rapidement aux changements et aux nouvelles priorités.

Voici un aperçu de notre tableau Kanban :

À Faire : Cette colonne contient toutes les tâches qui doivent encore être commencées. Cela inclut des éléments comme la publication de posts backend, la page de la communauté frontale, les tests unitaires et fonctionnels, la correction des bogues, et l'optimisation des performances.

En Cours : Cette colonne montre les tâches en cours de développement, comme la connexion au tableau de bord.

Terminé : Une fois les tâches terminées, elles sont déplacées dans cette colonne. Par exemple, l'intégration de l'API des exercices et l'authentification backend ont été complétées avec succès.

À Tester : Avant la finalisation, certaines tâches nécessitent des tests supplémentaires pour s'assurer qu'elles fonctionnent comme prévu. Par exemple, la messagerie est actuellement dans cette colonne.

Grâce à cette organisation visuelle, la méthode Kanban m'a permis de :

Visualiser le flux de travail : En voyant toutes les tâches à accomplir, en cours et terminées, on peut mieux comprendre le travail en cours et identifier les goulots d'étranglement.



Limiter le travail en cours : En définissant des limites sur le nombre de tâches pouvant être en cours simultanément, nous avons pu nous assurer que l'équipe se concentre sur l'achèvement des tâches plutôt que de commencer de nouvelles tâches sans fin.

Améliorer la livraison continue : La méthode Kanban favorise une approche de livraison continue, où les tâches sont terminées et mises en production dès qu'elles sont prêtes.

L'utilisation de cette méthode a permis de maintenir une organisation efficace et de s'assurer que toutes les tâches avancent de manière fluide. En suivant cette méthode, j'ai pu gérer les priorités et identifier les obstacles potentiels. Cela a également permis de garantir la qualité du produit final en facilitant les tests continus et les retours d'expérience.

Ajouter capture trello

## 14. Sauvegarde

Pour mon projet Bbliio nest, j'ai utilisé Bitbucket pour gérer le versionnement du code, permettant une collaboration efficace et une gestion structurée des différentes versions du projet.

**Git** est un système de contrôle de version décentralisé, ce qui signifie que chaque développeur possède une copie complète de l'historique du projet. **bitbucket**, en tant que plateforme d'hébergement de référentiels Git, permet de stocker, partager et collaborer sur le code source de manière centralisée.

Pour Biblionest, chaque modification du code est suivie et enregistrée dans des commits, permettant de revenir à n'importe quelle version précédente si nécessaire. Bitbucket facilite également la gestion des merge requests, permettant de réviser et de fusionner les modifications proposées par les différents contributeurs.

### Gestion des Branches

J'ai adopté une stratégie de gestion des branches pour maintenir la stabilité du code tout en développant de nouvelles fonctionnalités. La capture d'écran ci-dessous montre la structure des branches du projet sur bitbucket :

AJOUTER CAPTURE DE BITBUCKET (BRANCHES ET COMMIT)

**main** : Cette branche contient la version stable et prête pour la mise en production de l'application. Toutes les fonctionnalités terminées et testées sont fusionnées dans cette branche.

**dev** : Utilisée pour intégrer les nouvelles fonctionnalités et les correctifs avant de les fusionner dans main. Cette branche est la principale branche de développement.

**Branches de fonctionnalités** : Chaque nouvelle fonctionnalité est développée dans une branche dédiée, comme add\_book, connexion etc. Cela permet de travailler sur des fonctionnalités spécifiques sans affecter le code stable.

La gestion des branches permet de maintenir un code stable dans la branche principale tout en favorisant le développement parallèle de plusieurs fonctionnalités.

## 15. Outils

### Outils de développement :

Visual Studio Code (éditeur de texte)

Wampp (serveur local)

### Outil de gestion de projet :

Trello (méthode kanban)

### Outils de documentation :

Word

## 16. Conclusion

Développer l'application biblionest a été une expérience enrichissante et formatrice. Tout au long du projet, j'ai pu approfondir mes compétences en développement web, en particulier avec le framework Laravel. La création de cette application m'a permis de mieux comprendre les défis liés à la conception d'une plateforme interactive et engageante pour les utilisateurs, tout en garantissant une expérience fluide et intuitive. Le processus de développement a impliqué plusieurs aspects, allant de la conception de l'interface utilisateur à la mise en place de fonctionnalités complexes telles que la mise en place des pénalités de retard automatique. Chaque étape du projet a été une occasion d'apprentissage, me permettant de surmonter des obstacles techniques et de trouver des solutions innovantes. Cependant, il ne s'agit que de la première version de biblionest. En regardant vers l'avenir, je suis enthousiaste à l'idée de continuer à travailler sur ce projet et de développer une V2 encore plus avancée. Les améliorations envisagées comprennent le développement de l'interface visiteurs et utilisateurs avec un design attrayant et moderne.

Ces améliorations viseront à rendre l'application autonome sur les prêts de livres.