

# Cahier des Charges :

## C-BookManage

## 1. Introduction

### 1.1. Contexte

Ce projet s'inscrit dans le cadre de l'apprentissage de la programmation en langage C. Il vise à initier le développeur débutant aux concepts fondamentaux de la programmation, tels que les structures de données, les opérations sur les fichiers, et l'interaction utilisateur via la ligne de commande.

### 1.2. Objectifs

L'objectif principal de ce projet est de développer une application de gestion de bibliothèque permettant de :

- Ajouter des livres à la bibliothèque.
- Rechercher des livres par titre ou auteur.
- Afficher la liste des livres disponibles.
- Enregistrer et charger les données de la bibliothèque à partir d'un fichier.

## 2. Description du Projet

### 2.1. Fonctionnalités

#### **2.1.1. Ajout de Livres**

- L'utilisateur doit pouvoir ajouter un nouveau livre à la bibliothèque en fournissant les informations suivantes : titre, auteur, année de publication, et identifiant unique (ID).

#### **2.1.2. Recherche de Livres**

- L'utilisateur doit pouvoir rechercher des livres par titre ou par auteur. Les résultats de la recherche doivent être affichés à l'écran.

#### **2.1.3. Affichage des Livres**

- L'utilisateur doit pouvoir afficher la liste complète des livres disponibles dans la bibliothèque. Chaque livre doit être affiché avec ses informations complètes (titre, auteur, année de publication, ID).

#### **2.1.4. Enregistrement et Chargement des Données**

- L'application doit pouvoir enregistrer la liste des livres dans un fichier texte.
- L'application doit pouvoir charger la liste des livres à partir d'un fichier texte lors de son démarrage.

## **3. Spécifications Techniques**

### **3.1. Langage de Programmation**

- Le projet sera implémenté en langage C.

### **3.2. Structure des Données**

- Utilisation de structures pour représenter les livres.
- Utilisation d'un tableau dynamique pour stocker la collection de livres.

### **3.3. Interface Utilisateur**

- Interaction avec l'utilisateur via la ligne de commande (terminal).

### **3.4. Gestion des Fichiers**

- Utilisation des fonctions de la bibliothèque standard C pour lire et écrire dans les fichiers.

## 4. Déroulement du Projet

### 4.1. Étapes du Développement

#### **4.1.1. Jour 1 : Configuration et Planification**

- Installation de l'environnement de développement (IDE, compilateur).
- Création de la structure de base du projet.
- Définition des structures de données nécessaires (struct Livre, etc.).

#### **4.1.2. Jour 2 : Implémentation de l'Ajout de Livres**

- Implémentation de la fonction d'ajout de livres.
- Test et validation de l'ajout de livres.

#### **4.1.3. Jour 3 : Implémentation de la Recherche de Livres**

- Implémentation de la fonction de recherche par titre.
- Implémentation de la fonction de recherche par auteur.
- Test et validation des fonctions de recherche.

#### **4.1.4. Jour 4 : Implémentation de l'Affichage des Livres**

- Implémentation de la fonction d'affichage de la liste des livres.
- Test et validation de l'affichage des livres.

#### **4.1.5. Jour 5 : Implémentation de l'Enregistrement et du Chargement des Données**

- Implémentation de la fonction d'enregistrement des données dans un fichier.
- Implémentation de la fonction de chargement des données à partir d'un fichier.
- Test et validation de la gestion des fichiers.

#### **4.1.6. Jour 6 : Tests et Débogage**

- Réalisation de tests complets sur toutes les fonctionnalités.
- Correction des bugs éventuels.

#### 4.1.7. Jour 7 : Documentation et Finalisation

- Rédaction de la documentation du code.
- Rédaction d'un guide utilisateur pour l'application.
- Finalisation et préparation pour la présentation.

#### 4.1.8. Création d'un Dépôt GitHub et README

- Création d'un dépôt GitHub pour héberger le projet.
- Initialisation du dépôt avec un fichier `.gitignore` approprié.
- Rédaction d'un fichier `README.md` contenant :
  - Une description du projet.
  - Les instructions d'installation.
  - Les instructions d'utilisation.
  - Les informations sur la structure du code.
  - Les informations sur la contribution.
- Création d'un fichier `requirements.txt` (si nécessaire pour des dépendances spécifiques, par exemple pour des outils de développement ou de test).


### 4.2. Étapes de Création du Dépôt GitHub et Fichiers Associés

#### 4.2.1. Création d'un Dépôt GitHub

1. Allez sur [GitHub](https://github.com) et connectez-vous.
2. Cliquez sur le bouton "New repository" pour créer un nouveau dépôt.
3. Nommez le dépôt (par exemple, `bibliotheque-gestion`), ajoutez une description si nécessaire, et choisissez de rendre le dépôt public ou privé.
4. Cochez l'option "Initialize this repository with a README" pour créer automatiquement un fichier `README.md`.
5. Cliquez sur "Create repository".

#### 4.2.2. Clonage du Dépôt Localement

bash

 Copier le code

```
git clone https://github.com/votre_nom_utilisateur/bibliotheque-gestion.git
cd bibliotheque-gestion
```

### 4.2.3. Création du Fichier `.gitignore`

Créez un fichier `.gitignore` pour exclure les fichiers inutiles :

```
# .gitignore
*.o
*.exe
*.log
*.tmp
```

### 4.2.4. Rédaction du Fichier `README.md`

Mettez à jour le fichier `README.md` :

```
# Gestion de Bibliothèque


## Description
Une application de gestion de bibliothèque développée en langage C. Elle permet d

## Installation
Clonez le dépôt GitHub et compilez le programme :
```bash
git clone https://github.com/votre_nom_utilisateur/bibliotheque-gestion.git
cd bibliotheque-gestion
gcc -o bibliotheque main.c
```

## Utilisation

Exécutez le programme depuis la ligne de commande :

```
bash
```

 Copier le code

```
./bibliotheque
```

## Structure du Code

- `main.c` : Le fichier principal contenant le point d'entrée du programme.
- `livre.h` et `livre.c` : Les fichiers définissant et implémentant la structure et les fonctions liées aux livres.
- `bibliotheque.h` et `bibliotheque.c` : Les fichiers définissant et implémentant la gestion de la bibliothèque.

## Contribution

Les contributions sont les bienvenues. Veuillez créer une branche pour vos modifications et soumettre une pull request.

## Auteurs

- Votre Nom

## Licence

Ce projet est sous licence MIT. Voir le fichier `LICENSE` pour plus d'informations.

### 4.2.5. Création du Fichier `requirements.txt` (si nécessaire)

Si le projet nécessite des dépendances spécifiques, listez-les dans un fichier `requirements.txt`. Par exemple, si des outils Python sont utilisés pour le développement ou les tests, vous pouvez ajouter :

```
pytest  
flake8
```

### 4.2.6. Push des Modifications sur GitHub

```
```bash  
git add .  
git commit -m "Initial commit with project structure and documentation"  
git push origin main
```

## 5. Livrables

- Code source complet de l'application.
- Fichier exécutable de l'application.
- Documentation du code.
- Guide utilisateur.
- Lien vers le dépôt github du projet