

Documentation Technique - Gestion d'une Bibliothèque

1. Introduction

Cette documentation présente l'analyse et la conception d'un système informatisé de gestion des prêts et retours de livres pour une bibliothèque.

1.1 Objectifs

- Gérer l'enregistrement des livres et des lecteurs.
- Permettre aux lecteurs d'emprunter et de rendre des livres.
- Suivre les prêts et assurer le respect des délais.

1.2 Périmètre du projet

- Gestion des lecteurs (inscription, modification, suppression).
 - Gestion des livres (ajout, mise à jour, suppression).
 - Gestion des prêts et des retours.
-

2. Modélisation des données

2.1 Modèle Conceptuel de Données (MCD)

Entités identifiées :

- **Lecteur** (id_lecteur, nom, prénom, adresse, téléphone)
- **Livre** (id_livre, titre, auteur, année, catégorie)
- **Prêt** (id_prêt, date_emprunt, date_retour, état, id_lecteur, id_livre)

Relations :

- Un lecteur peut emprunter plusieurs livres, mais un livre ne peut être emprunté que par un seul lecteur à la fois.
- Un livre peut être emprunté plusieurs fois au fil du temps.
- Un prêt est une association entre un lecteur et un livre, avec des informations sur la date d'emprunt et de retour.

2.2 Modèle Logique de Données (MLD)

Tables :

- **LECTEUR** (id_lecteur, nom, prénom, adresse, téléphone)
 - **LIVRE** (id_livre, titre, auteur, année, catégorie)
 - **PRÊT** (id_prêt, date_emprunt, date_retour, état, id_lecteur, id_livre)
-

3. Implémentation SQL

- CREATE TABLE Lecteur (
 - id_lecteur INT PRIMARY KEY,
 - nom VARCHAR(50),
 - prénom VARCHAR(50),
 - adresse VARCHAR(100),
 - téléphone VARCHAR(15)
 -);
 -
 - CREATE TABLE Livre (
 - id_livre INT PRIMARY KEY,
 - titre VARCHAR(100),
 - auteur VARCHAR(50),
 - année INT,
 - catégorie VARCHAR(50)
 -);
 -
 - CREATE TABLE Prêt (
 - id_prêt INT PRIMARY KEY,
 - date_emprunt DATE,
 - date_retour DATE,
 - état VARCHAR(20),
 - id_lecteur INT,
 - id_livre INT,
 - FOREIGN KEY (id_lecteur) REFERENCES Lecteur(id_lecteur),
 - FOREIGN KEY (id_livre) REFERENCES Livre(id_livre)
 -);
-

4. Architecture de l'application

4.1 Choix technologiques

- **Langage de programmation** : PHP / Java / Python
- **Framework** : Symfony / Spring Boot / Django
- **Base de données** : MySQL / PostgreSQL
- **Front-end** : HTML, CSS, JavaScript, Bootstrap

4.2 Architecture MVC

L'application suivra le modèle MVC (Modèle-Vue-Contrôleur) pour séparer la logique métier, l'affichage et le contrôle des actions des utilisateurs.

- **Modèle (Model)** : Gère les interactions avec la base de données (Lecteur, Livre, Prêt).
 - **Vue (View)** : Interface utilisateur pour consulter et gérer les prêts.
 - **Contrôleur (Controller)** : Gère les requêtes des utilisateurs et met à jour les modèles et les vues.
-

5. Fonctionnalités principales

5.1 Gestion des utilisateurs (lecteurs)

- Inscription d'un nouveau lecteur
- Modification / suppression des informations d'un lecteur
- Consultation de la liste des lecteurs

5.2 Gestion des livres

- Ajout d'un nouveau livre
- Modification / suppression d'un livre
- Consultation du catalogue de la bibliothèque

5.3 Gestion des prêts

- Enregistrement d'un prêt
 - Retour d'un livre
 - Consultation des prêts en cours
 - Notification en cas de retard
-

6. Prochain Projet (MVC)

- **Date limite** : 06/03/2025
- **Objectif** : Développement d'un Système d'Information (SI) pour la gestion des prêts de livres.
- **Client** : Ville de Paris
- **Prestataire** : CFA INSTA
-