Cas médiathèque – proposition de solution

# Diagramme de cas d'utilisation :

## Gestion des documents :

* + Consulter la liste des documents
  + Rechercher un document par titre, auteur, côte, etc.
  + Emprunter un livre ou un CD-ROM
  + Consulter un microfilm sur place

## Gestion des lecteurs :

* + Obtenir une carte de lecteur
  + Mettre à jour les informations du lecteur (adresse, etc.)
  + Consulter les documents empruntés
  + Payer la cotisation

## Gestion des bénévoles :

* + Enregistrer un emprunt (sur une période déterminée)
  + Consulter la disponibilité des documents

## Gestion du personnel :

* + Enregistrer un nouvel achat de document
  + Enregistrer un nouveau lecteur
  + Consulter la situation de chaque lecteur
  + Envoyer des lettres de relance

# Diagramme de séquence (exemples) :

## Emprunter un livre :

* + Acteur : Lecteur
  + Actions :
    1. Sélectionner le livre à emprunter
    2. Présenter la carte de lecteur
    3. Enregistrement de l'emprunt dans le système

## Enregistrer un nouvel achat de document :

* + Acteur : Personnel
  + Actions :
    1. Ajouter les détails du nouveau document
    2. Générer un numéro de côte unique
    3. Enregistrer le document dans le système

# Diagramme d'activité :

## Processus d'emprunt d'un CD-ROM :

* + Commence par la sélection du CD-ROM
  + Vérification de la caution et de la carte de lecteur
  + Si les conditions sont remplies, enregistrement de l'emprunt
  + Fin du processus

## Processus d'envoi de lettre de relance :

* + Commence par la vérification des emprunts en retard
  + Génération de lettres pour les lecteurs concernés
  + Envoi des lettres
  + Fin du processus

# Diagramme de classes :

## Classe Document :

* + Propriétés : côte, titre, auteur, type (livre, journal, microfilm, CD-ROM), statut (disponible, emprunté, perdu), etc.
  + Méthodes :   
    **getDetails(): string**: Retourne une chaîne de caractères décrivant les détails du document (côte, titre, auteur, etc.).  
    **isAvailable(): boolean**: Vérifie si le document est disponible.  
    **checkOut(lecteur: Lecteur): void**: Enregistre l'emprunt du document par un lecteur.  
    **checkIn(): void**: Enregistre le retour du document.

## Classe Lecteur :

* + Propriétés : numéro de lecteur, nom, adresse, cotisation payée, nombre d'emprunts en cours, etc.
  + Méthodes :  
    **getDetails(): string**: Retourne une chaîne de caractères décrivant les détails du lecteur (numéro de lecteur, nom, adresse, etc.).  
    **paySubscription(): void**: Enregistre le paiement de la cotisation du lecteur.  
    **getActiveLoans(): Document[]**: Retourne la liste des documents actuellement empruntés par le lecteur.

## Classe Emprunt :

* + Propriétés : numéro d'emprunt, date d'emprunt, date de retour prévue, document emprunté, lecteur concerné, etc.
  + Méthodes :  
    **calculateDueDate(): Date**: Calcule la date de retour prévue pour l'emprunt.  
    **isOverdue(): boolean**: Vérifie si l'emprunt est en retard.

## Classe Bénévole :

* + Propriétés : nom, période d'autorisation, etc.
  + Méthodes :  
    **authorizeLoan(): boolean**: Autorise un bénévole à effectuer des enregistrements d'emprunts pendant sa période d'autorisation.