Cahier des charges

Thème : Application de gestion de bibliothèque (Emprunt et Retours)

1. Description du projet

Domaine

Gestion d'une bibliothèque permettant d'administrer les livres, les utilisateurs, et les emprunts.

Objectif général

Développer une application desktop en Java permettant :

- La gestion des livres d'une bibliothèque
- La gestion des membres
- La gestion des emprunts et retours
- L'accès sécurisé selon le rôle (ADMIN / USER)
- Une interface JavaFX fluide et intuitive

Utilisateurs

- **ADMIN** : gère tout (livres, membres, emprunts, utilisateurs, statistiques)
- **USER** : simple employé/bibliothécaire → gère uniquement les emprunts/retours et la consultation des livres.

---

2. Exigences fonctionnelles

2.1 Authentification

- Connexion via login + mot de passe hashé (BCrypt recommandé)
- Déconnexion
- Tableau de bord après connexion

2.2 Gestion des rôles

- **ADMIN:**
    - Gérer les livres (CRUD)
    - Gérer les membres (CRUD)
    - Gérer les utilisateurs du système (CRUD)

    - Gérer les emprunts / retours
- **USER:**
    - Consulter les livres disponibles
    - Gérer les emprunts / retours uniquement

Les boutons, pages et actions doivent s'activer/se désactiver en fonction du rôle.

2.3 Modules CRUD prévus

1. Livre

- Ajouter un livre (titre, auteur, année, catégorie, stock)
- Modifier / Supprimer
- Rechercher un livre
- Lister les livres

2. Membre

- Ajouter un membre (nom, email, téléphone…)
- Modifier / Supprimer
- Lister / Rechercher

3. Emprunts

- Enregistrer un emprunt : (date_emprunt, date_retour_prévue)
- Retourner un livre
- Voir l'historique des emprunts

4. Utilisateurs (ADMIN uniquement)

- Créer des comptes utilisateurs
- Définir rôle ADMIN ou USER

Ce module permet de respecter le minimum de "2 entités liées". Ici : Livre ↔ Emprunt ↔ Membre.

---

3. Exigences non fonctionnelles

Sécurité

- Hashage des mots de passe
- Règles de visibilité selon rôle
- Validation des champs (email valid, texte non vide, etc.)

Ergonomie

- Interface JavaFX claire
- Navigation simple (menus, tableaux, formulaires)
- Messages d'erreur et de confirmation

Contraintes techniques

- Java 11+ ou 17
- JavaFX
- MySQL
- JDBC
- Architecture MVC ou MVVM
- GitHub public

---

4. Conception

Entités principales

- **Utilisateur** (id, username, passwordHash, role)
- **Livre** (idLivre, titre, auteur, année, catégorie, stock)
- **Membre** (idMembre, nom, email, téléphone)
- **Emprunt** (idEmprunt, idLivre, idMembre, dateEmprunt, dateRetourPrévue, dateRetourReelle)

Relations

- Un **membre** peut faire plusieurs **emprunts**
- Un **livre** peut être emprunté plusieurs fois
- Relation principale :
  → **Livre (1) -- (N) Emprunts (N) -- (1) Membre**

Architecture proposée

src/

controller/

model/

dao/

view/

utils/

---

5. Livrables attendus

Code source GitHub

- JavaFX + JDBC + MySQL
- README clair

Rapport final

1. Introduction : problématique + objectifs
2. Diagrammes UML :
    - Use Case (ADMIN et USER)
    - Classe
3. Implémentation (captures + explications)
4. Tests (scénarios de connexion, CRUD, emprunts…)
5. Conclusion + améliorations possibles (statistiques, QR Code, notifications, etc.)