**Cas d'Usage - Système de Gestion de Bibliothèque**

**1. Introduction**

**Ce document présente les cas d'usage du système de gestion de bibliothèque développé avec Spring Boot. Le système permet la gestion complète des adhérents, des livres, des prêts, des réservations et des pénalités.**

**2. Acteurs du Système**

**2.1 Acteur Principal**

* **Bibliothécaire : Utilisateur principal qui gère les opérations de la bibliothèque**

**2.2 Acteurs Secondaires**

* **Adhérent : Membre de la bibliothèque qui emprunte des livres**
* **Système : Processus automatisés (calcul des pénalités, vérifications)**

**3. Cas d'Usage Détaillés**

**3.1 Gestion des Adhérents**

**CU-001 : Créer un Adhérent**

**Acteur Principal : Bibliothécaire Objectif : Enregistrer un nouvel adhérent dans le système Préconditions : Le bibliothécaire est connecté au système Postconditions : L'adhérent est créé avec un type d'adhérent assigné**

**Scénario Principal :**

1. **Le bibliothécaire saisit les informations de l'adhérent (nom, prénom, date de naissance, date d'inscription)**
2. **Le système assigne automatiquement un type d'adhérent selon l'âge**
3. **Le système définit le statut d'abonnement (abonné/non abonné)**
4. **Le système sauvegarde l'adhérent**
5. **Le système confirme la création**

**Scénarios Alternatifs :**

* **2a. Les informations sont incomplètes : le système demande de compléter les champs obligatoires**

**CU-002 : Vérifier les Restrictions d'un Adhérent**

**Acteur Principal : Système Objectif : Vérifier si un adhérent peut emprunter des livres Préconditions : L'adhérent existe dans le système Postconditions : Le système retourne le type d'adhérent si autorisé**

**Scénario Principal :**

1. **Le système recherche l'adhérent par ID**
2. **Le système vérifie si l'adhérent est abonné**
3. **Le système vérifie si l'adhérent n'est pas pénalisé**
4. **Le système retourne le type d'adhérent**

**Scénarios d'Exception :**

* **1a. Adhérent non trouvé : AdherentRestrictionException("Aucun adhérent trouvé avec cet ID")**
* **2a. Adhérent non abonné : AdherentRestrictionException("L'adhérent n'est pas abonné, prêt interdit")**
* **3a. Adhérent pénalisé : AdherentRestrictionException("L'adhérent est Penalise")**

**3.2 Gestion des Livres et Exemplaires**

**CU-003 : Gérer les Exemplaires**

**Acteur Principal : Bibliothécaire Objectif : Gérer la disponibilité des exemplaires Préconditions : L'exemplaire existe dans le système Postconditions : Le statut de l'exemplaire est mis à jour**

**Scénario Principal - Rendre Disponible :**

1. **Le bibliothécaire sélectionne un exemplaire**
2. **Le système met le statut à "Disponible" (ID=1)**
3. **Le système sauvegarde les modifications**

**Scénario Principal - Rendre Indisponible :**

1. **Le bibliothécaire sélectionne un exemplaire**
2. **Le système met le statut à "Indisponible" (ID=2)**
3. **Le système sauvegarde les modifications**

**CU-004 : Rechercher des Livres**

**Acteur Principal : Bibliothécaire Objectif : Rechercher des livres par auteur Préconditions : Des livres existent dans le système Postconditions : Liste des livres correspondant aux critères**

**Scénario Principal :**

1. **Le bibliothécaire saisit le nom de l'auteur**
2. **Le système recherche tous les livres contenant ce nom (insensible à la casse)**
3. **Le système affiche la liste des résultats**

**3.3 Gestion des Prêts**

**CU-005 : Créer un Prêt**

**Acteur Principal : Bibliothécaire Objectif : Enregistrer un nouveau prêt Préconditions :**

* **L'adhérent existe et est autorisé**
* **L'exemplaire est disponible Postconditions : Le prêt est créé et l'exemplaire devient indisponible**

**Scénario Principal :**

1. **Le bibliothécaire sélectionne l'adhérent**
2. **Le système vérifie les restrictions de l'adhérent**
3. **Le bibliothécaire sélectionne l'exemplaire**
4. **Le système vérifie la disponibilité**
5. **Le système crée le prêt avec les dates appropriées**
6. **Le système rend l'exemplaire indisponible**

**CU-006 : Consulter les Prêts**

**Acteur Principal : Bibliothécaire Objectif : Consulter les prêts selon différents critères**

**Variantes :**

* **Par Adhérent : Afficher tous les prêts d'un adhérent spécifique**
* **En Cours : Afficher tous les prêts non retournés (dateRetourReelle = null)**
* **Par Statut : Afficher tous les prêts ayant un statut spécifique**

**3.4 Gestion des Prolongements**

**CU-007 : Demander un Prolongement**

**Acteur Principal : Bibliothécaire Objectif : Prolonger la durée d'un prêt Préconditions :**

* **Le prêt existe**
* **L'adhérent n'a pas dépassé son quota de prolongements (max 2)**
* **Le prêt n'a pas déjà été prolongé Postconditions : Le prêt est prolongé de 7 jours**

**Scénario Principal :**

1. **Le bibliothécaire sélectionne le prêt à prolonger**
2. **Le système vérifie l'existence du prêt**
3. **Le système vérifie le quota de prolongements de l'adhérent**
4. **Le système vérifie que le prêt n'a pas déjà été prolongé**
5. **Le système crée un enregistrement de prolongement**
6. **Le système met à jour la date de retour prévue (+7 jours)**
7. **Le système marque le prêt comme prolongé**
8. **Le système incrémente le quota de prolongements utilisés**

**Scénarios d'Exception :**

* **2a. Prêt inexistant : retourne false**
* **3a. Quota dépassé : retourne false**
* **4a. Prêt déjà prolongé : retourne false**

**3.5 Gestion des Réservations**

**CU-008 : Créer une Réservation**

**Acteur Principal : Bibliothécaire Objectif : Créer une réservation pour un adhérent Préconditions : L'adhérent et le livre existent Postconditions : La réservation est créée avec le statut "en\_attente"**

**Scénario Principal :**

1. **Le bibliothécaire sélectionne l'adhérent et le livre**
2. **Le système crée la réservation**
3. **Le système assigne automatiquement le statut "en\_attente"**
4. **Le système sauvegarde la réservation**

**3.6 Gestion des Pénalités**

**CU-009 : Gérer les Pénalités**

**Acteur Principal : Bibliothécaire Objectif : Créer, modifier ou supprimer des pénalités Préconditions : L'adhérent existe Postconditions : La pénalité est gérée selon l'opération**

**Scénario Principal - Création :**

1. **Le bibliothécaire sélectionne l'adhérent**
2. **Le bibliothécaire choisit le type de pénalité**
3. **Le système définit les dates de début et fin**
4. **Le système sauvegarde la pénalité**
5. **Le système met à jour le statut de pénalisation de l'adhérent**

**CU-010 : Consulter les Pénalités**

**Acteur Principal : Bibliothécaire Objectif : Consulter les pénalités d'un adhérent Préconditions : L'adhérent existe Postconditions : Liste des pénalités affichée**

**Scénario Principal :**

1. **Le bibliothécaire sélectionne l'adhérent**
2. **Le système recherche toutes les pénalités de cet adhérent**
3. **Le système affiche la liste des pénalités**

**3.7 Gestion des Droits de Prêt**

**CU-011 : Vérifier les Droits de Prêt**

**Acteur Principal : Système Objectif : Vérifier si un type d'adhérent peut emprunter un type de livre Préconditions : Les types d'adhérent et de livre existent Postconditions : Autorisation ou refus du prêt**

**Scénario Principal :**

1. **Le système identifie le type d'adhérent**
2. **Le système identifie le type d'utilisation du livre**
3. **Le système consulte la table des droits de prêt**
4. **Le système retourne l'autorisation (true/false)**

**4. Règles Métier**

**4.1 Règles d'Adhésion**

* **Un adhérent doit être abonné pour emprunter**
* **Un adhérent pénalisé ne peut pas emprunter**
* **Le type d'adhérent détermine les droits de prêt**

**4.2 Règles de Prêt**

* **Un prêt ne peut être prolongé qu'une seule fois**
* **Maximum 2 prolongements par adhérent**
* **Prolongement de 7 jours maximum**
* **Un exemplaire ne peut être prêté que s'il est disponible**

**4.3 Règles de Pénalité**

* **Les pénalités ont une date de début et de fin**
* **Un adhérent pénalisé ne peut pas emprunter**
* **Différents types de pénalités peuvent être appliqués**

**4.4 Règles de Réservation**

* **Les réservations sont créées avec le statut "en\_attente"**
* **Une réservation peut être annulée ou confirmée**

**5. Exceptions et Gestion d'Erreurs**

**5.1 AdherentRestrictionException**

**Cette exception personnalisée est levée dans les cas suivants :**

* **Adhérent non trouvé**
* **Adhérent non abonné**
* **Adhérent pénalisé**

**5.2 Gestion des Erreurs**

**Le système gère les erreurs de manière robuste :**

* **Vérification de l'existence des entités avant manipulation**
* **Validation des données avant sauvegarde**
* **Retour de valeurs nulles ou booléennes pour les opérations facultatives**

**6. Points d'Extension**

**6.1 Quota Management**

**Le système inclut une interface QuotaService pour gérer les quotas des adhérents :**

* **Suivi des prêts en cours**
* **Gestion des prolongements**
* **Réinitialisation des quotas**

**6.2 Historique des Statuts**

**Le système maintient un historique des changements de statut des prêts pour traçabilité.**

**7. Conclusion**

**Ce système de gestion de bibliothèque offre une solution complète pour :**

* **La gestion des adhérents et leurs restrictions**
* **Le suivi des prêts et prolongements**
* **La gestion des réservations**
* **Le système de pénalités**
* **La vérification des droits de prêt**

**Le système est conçu avec une architecture modulaire utilisant Spring Boot, garantissant la maintenabilité et l'extensibilité.**