**Rapport - Gestion de prêt de livres pour une bibliothèque**

**1. Introduction**

Le présent rapport décrit la réalisation d'une Preuve de Concept (PoC) d'un programme de gestion de prêt de livres pour une bibliothèque. L'objectif de ce travail est de se familiariser avec les Patterns Tactiques et l'Event Sourcing de Domain-Driven Design (DDD). Le programme est conçu en mettant en pratique les principes de DDD et en utilisant une Architecture Hexagonale pour une meilleure séparation des préoccupations.

**Architecture Hexagonale :** Nous avons organisé le programme en plusieurs couches distinctes, isolant le domaine métier de l'infrastructure technique. Cela facilite la maintenance, les tests et l'évolutivité du système.

**Modélisation du domaine :** Nous avons identifié les concepts clés du domaine tels que Livre, Membre, Emprunt, etc., et les avons modélisés comme des Entities. Les champs spécifiques des Entities ont été définis comme des Value Objects pour garantir leur immutabilité.

**Agrégats :** Nous avons utilisé des Agrégats pour gérer les collections d'Entities liées, par exemple, l'agrégat Emprunt gère les opérations liées aux emprunts de livres.

**Event Sourcing :** Nous avons mis en place le concept d'Event Sourcing en utilisant des Domain Events pour capturer les événements importants du domaine. Lorsqu'une action est effectuée, un événement est émis, et les Agrégats réagissent à ces événements pour effectuer d'autres actions (logs, envoi de mails, etc.).

**Type d'application :** Nous avons choisi d'implémenter une API REST pour fournir une interface conviviale permettant aux utilisateurs d'interagir avec le système.

**Type de persistance :** Nous avons utilisé une base de données pour stocker les informations relatives aux livres, aux membres, aux emprunts, etc.

**Tests unitaires :** Nous avons mis en place des tests unitaires pour assurer la qualité et la fiabilité du code. Ces tests couvrent les différentes fonctionnalités du programme pour détecter les erreurs éventuelles dès leur apparition.

**4. Conclusion**

La réalisation de cette Preuve de Concept nous a permis de mieux comprendre l'application des Patterns Tactiques de DDD et l'utilisation de l'Event Sourcing dans la conception d'un système logiciel. En suivant une architecture Hexagonale, nous avons pu obtenir un code modulaire et évolutif. Le programme de gestion de prêt de livres pour une bibliothèque offre une interface pratique pour les utilisateurs et assure une gestion robuste des emprunts. Les tests unitaires ont joué un rôle essentiel dans la validation de la fonctionnalité du système.

En poursuivant ce travail, il serait intéressant de continuer à améliorer le programme en ajoutant davantage de fonctionnalités, en optimisant les performances et en explorant d'autres types de persistance. Cette PoC fournit une base solide pour le développement d'un système complet de gestion de bibliothèque basé sur les principes de DDD et d'Event Sourcing.

Diagramme de Classes

Une image contenant texte, capture d’écran, diagramme, Police

Description générée automatiquement

Diagramme de séquence (Cas d'emprunt) :

Une image contenant texte, capture d’écran, astronomie

Description générée automatiquement

Event story

Une image contenant texte, capture d’écran, Rectangle, Police

Description générée automatiquement

1. Agrégat "Bibliothèque" :
   * Responsable de gérer les livres disponibles à la bibliothèque et les emprunts en cours.
   * Gère les événements LivreAjouté, LivreEmprunté et LivreRendu.
   * Possède une liste de livres et une liste d'emprunts en cours.
2. Entité "Livre" :
   * Représente un livre dans la bibliothèque.
   * Contient des informations telles que le titre du livre, l'auteur, etc.
   * Peut être identifié par un ID unique.
3. Entité "Membre" :
   * Représente un membre de la bibliothèque.
   * Contient des informations telles que le nom du membre, prénom, etc.
   * Peut être identifié par un ID unique.
4. Entité "Emprunt" :
   * Représente un emprunt de livre effectué par un membre.
   * Contient des informations telles que le livre emprunté, le membre qui a emprunté le livre, la date d'emprunt et la date de rendu prévue.
   * Peut être identifié par un ID unique.

Les User Story prévisionnel :

1. User Stories : En nous basant sur les événements identifiés, nous pouvons formuler des User Stories pour décrire les fonctionnalités du système.

User Story 1 : En tant qu'utilisateur, je veux connaître la liste des livres disponibles dans la bibliothèque.

User Story 2 : En tant qu'utilisateur, je veux connaître la liste des membres inscrits à la bibliothèque. User Story 3 : En tant qu'utilisateur, je veux connaître la liste des livres empruntés, avec les informations sur le membre qui a emprunté le livre, la date d'emprunt et la date de rendu.