



Cégep **André-Laurendeau**

Hiver 2022

Cégep André-Laurendeau

Travail pratique #3

420-445 Programmation dans un environnement transactionnel

François Lacoursière

Hiver 2022

Consignes du travail

- Le travail est à réaliser individuellement (Sauf pour établir les stories et acteurs).
- Ce travail compte pour 15% de la note finale.
- La date de remise est le dimanche 10 Avril à 23h55 (ou lundi 11 à 9h00)
- Le travail doit être remis à l'aide de GitHub. Vous pouvez me donner un fichier 'texte' (PAS WORD) dans Léa pour me donner le lien de votre GitHub
- Tout retard dans la remise entraînera une pénalité de 10% par jour de retard jusqu'à concurrence de 5 jours. Après cette date, la note zéro sera attribuée au travail.

Contexte du travail pratique

Dans ce travail pratique vous allez mettre en pratique les notions suivantes :

- Programmation java en utilisant le framework SpringData et une base de données H2 en mémoire. Vous devrez implanter les fonctionnalités suivantes tout en suivant le modèle des couches présentées en classe ainsi que la couche présentation à l'aide du framework Thymeleaf.
- Traitement des amendes
- Ajout de tous les types de documents à la bibliothèque (Livre, Cd, DVD)
- Recherche d'un document selon les critères suivant : Titre, auteur, année et catégorie. Pour le titre, on doit faire une recherche qui permette qu'elle ne soit pas exacte. Par exemple, si le titre d'un livre est : « Les parapluies sont disparus », on devra pouvoir obtenir le document avec les critères de recherche contenant simplement le mot « parapluies » ou le mot « disparus ».
- L'emprunt d'un document en tenant compte s'il reste au moins un exemplaire de disponible.

- Le retour d'un document emprunté (Prévoyez mettre la date de retour en paramètre à votre service pour ne pas avoir à attendre 21 jours pour effectuer le retour dans le cas d'un livre)
- Le paiement des frais par le client (s'il a des frais)
- Le client doit pouvoir obtenir la liste de ses emprunts, les dates auxquelles il doit retourner chacun des documents empruntés et également la liste des frais s'il en a.
- Vous devrez fournir une interface web permettant d'enregistrer un client, enregistrer un nouveau livre, emprunter un livre, retourner un livre et payer les amendes.
- Le code doit être remis dans un repo github avec des commits ayant la bonne granularité (comme le tp2).

Livrable

Chaque étudiant devra livrer sur LÉA les éléments suivants :

.

- Le code java dans le repo github. Je veux avoir des commits sensibles, c'est-à-dire ayant la bonne granularité. Quand une fonctionnalité est terminée, aussi petite soit-elle, je veux un commit. Je ne veux pas de commit qui chevauchent 2 ajouts de fonctionnalité (Ex : Emprunt d'un document et recherche de document dans le même commit)
- Vous devez inclure dans la racine de votre projet le cahier des charges en format .docx (-10 si absent) Les annexes doivent contenir le diagramme UML. N'oubliez pas de mettre en vert tout changement pour ce TP.

Besoin client (rappel)

La librairie de la ville de Javatown a besoin de moderniser sa bibliothèque. Un système manuel est présentement en place et ne fait plus l'affaire, plusieurs livres sont perdus, les amendes des clients se perdent et plusieurs autres petits problèmes hantent les opérations de cette bibliothèque.

Votre tâche tout au cours de la session sera de produire et de livrer une nouvelle solution pour informatiser les opérations de la bibliothèque. Pour les besoins du TP, on peut assumer que le logiciel ne sera utilisé que par une seule bibliothèque

Votre première tâche sera d'identifier les acteurs du système ainsi que de produire les 'stories' décrivant les tâches et responsabilités de chacun.

La bibliothèque prête plusieurs types de documents soit évidemment des livres mais également des cd et aussi des dvd. Les clients, qui sont des résidents de Javatown, peuvent emprunter gratuitement ces documents pour une période de 3 semaines pour les livres, 2 semaines pour les CD et 1 semaine pour les DVD. Si un emprunteur enfreint les délais, une amende de 25 cents par jour lui est facturé. Celui-ci ne pourra emprunter de nouveaux documents s'il a des frais accumulés à son compte. Un préposé de la bibliothèque s'occupe d'entrer les documents dans le système. Il y consigne notamment le titre, l'auteur, l'éditeur, l'année de publication, le nombre de pages et le genre de document. Pour les livres, cela se détermine selon qu'un document est soit un roman, un manuel scolaire, une étude ou un magazine. Les utilisateurs peuvent rechercher des documents à

l'aide de la nouvelle plate-forme. Ils peuvent notamment chercher pour les documents d'un auteur, d'un éditeur, ...

Les gestionnaires de la bibliothèque peuvent interroger le système pour savoir le nombre de documents emprunter par mois et également le nombre d'amende produite.

Grille de correction

Les objectifs pédagogiques visés par ce travail :

- Conception de la structure primaire d'un système à partir d'une description d'exigences incomplètes, fournie par l'enseignant.
(Mise à jour)
- Modélisation des exigences en utilisant la norme UML (Mise à jour)
- Programmation adéquate par couche, tel que vu en classe à l'aide du framework Spring Boot et Spring Data.
- Programmation adéquate de la couche présentation à l'aide du framework 'Thymeleaf'
- Utilisation des concepts clés de la programmation orientée objet.
- Programmation de la couche persistance à l'aide de Spring Data
- Définir un API adéquat à la couche service
- Bonne utilisation des DTOs

Grille de correction

Grille de correction						
Éléments visés	Excellent	Très bien	Bien	Moyen	Insuffisant	Nul
Documents. (Cahier des charges, Stories, UML) changements en vert	10	9 8	7 6	5	4 - 2	1 0
Spring Data Repositories bien définis	10	5	4	3	2	1 0
Bonne utilisation de la DI (Dependency Injection)	5	5	4	3	2	1 0
Contraintes et validation bien définies dans la couche Model	10	9 8	7 6	5	4 2	1 0
Emprunt document	10	9 8	7 6	5	4 2	1 0
Retour document	10	9 8	7 6	5	4 2	1 0
Ajout de document	10	9 8	7 6	5	4 2	1 0
Ajout d'un client	10	9 8	7 6	5	4 2	1 0
Paieement des amendes	10	9 8	7 6	5	4 2	1 0
Listing client	10	9 8	7 6	5	4 2	1 0
Bonne granularité des commits	5	9 8	7 6	5	4 2	1 0