

# Lab 1

## Kanban Board

### Description du projet

Vous devez concevoir une application Web avec les frameworks Spring, Spring Boot et Hibernate mettant en œuvre un tableau Kanban.

### Besoins Fonctionnels

L'application se compose de trois pages: la page permettant d'ajouter une tâche au tableau Kanban, la page présentant la liste des développeurs et la page affichant le tableau Kanban :

#### Tableau Kanban : Projet 1

[Liste des développeurs](#)

À faire +	En cours +	À tester +	Terminé +
Corriger le fichier CSS Fx COTE ≥			
	Écrire la classe App Fx COTE, James GOSLING ≤ ≥		
		Créer spring-servlet.xml Fx COTE ≤ ≥	
Rédiger la documentation James GOSLING ≥			

A noter, il existe différents types de tâches :

- les tâches en bleu représentent les fonctionnalités à réaliser
- les tâches en orange correspondent à des bugs

- les tâches en vert sont des améliorations à apporter au logiciel
- les tâches en magenta sont des “spikes”, des tâches d'exploration.

Le lien « + » de chaque colonne permet d'ajouter une tâche dans la colonne concernée. Ce lien redirige vers la page d'ajout de tâche.

Toutes les tâches peuvent être déplacées à l'aide des liens “<” et “>”. Si la tâche se trouve dans la colonne “À faire” le lien “<” n'est pas disponible. De la même manière, si la tâche se trouve dans la colonne “Terminé” le lien “>” n'est pas disponible. Les déplacements de tâches sont stockés dans la table HistoriqueDeplacement.

Un lien “Liste des développeurs” redirige vers la page listant les développeurs.

NB: Il n'est pas nécessaire d'écrire un fichier CSS ni d'utiliser Bootstrap.

Voici une capture de la page “Ajout d'une tâche”. La tâche est ajoutée au projet dont le nom est “Projet 1”:

## Ajout d'une Tâche

Projet	Projet 1
Colonne	À faire
Type de tâche	Amélioration ▾
Intitulé	<input type="text"/>
Développeur(s)	<div>Fx COTE</div> <div>James GOSLING</div>
Nombre d'heures prévues	<input type="text"/>
	<input type="button" value="Ajouter"/>

Si l'intitulé est vide, un message d'erreur s'affiche à droite de la zone de texte. Si aucun développeur n'est sélectionné, un message d'erreur s'affiche à droite de la liste déroulante. Si le nombre d'heures prévues n'a pas été renseigné, un message d'erreur s'affiche à droite du champ de saisie.

Voici une capture d'écran de la page présentant la liste des développeurs:

## Les Développeurs

Nom ▼	Prénom ▼	Email ▼	Date de début de contrat ▼
GOSLING	James	james.gosling@amazon.com	24/05/2018
COTE	Fx	fxcote@fxcote.com	22/09/2018

Page 1 ► ►►

Développeurs de 1 à 2 sur 2

[Retour au tableau Kanban](#)

Il est possible de trier sur n'importe quelle colonne du tableau.

Quatre boutons de navigation (Premier, Précédent, Suivant, Dernier) servent à parcourir l'ensemble des développeurs. En dessous de ces boutons de navigation, la position actuelle sur l'ensemble des développeurs apparaît, exemple: Développeurs de 1 à 4 sur 10.

Un lien "Retour au tableau Kanban" redirige vers le tableau Kanban.

## Règles de gestion

RG1 (Validation): la date de début de contrat doit être dans le passé et stockée en base

RG2 (Validation): l'email du développeur doit être unique en base

RG3 (Validation): le mot de passe du développeur doit contenir au moins 5 caractères

RG4 (Validation): la liste des développeurs à laquelle une tâche est confiée ne peut être vide

RG5 (Validation): une tâche doit être dans une et une seule colonne

- 3 -

RG6 (Validation): une tâche doit avoir un et un seul type de tâche

RG7 (Validation): un projet doit être associé à un client

RG8 (Validation): la couleur des types de tâches doit contenir exactement six caractères

RG9 (Validation): le nombre d'heures prévues doit être compris entre 1 et 144

## Architecture Logicielle

---

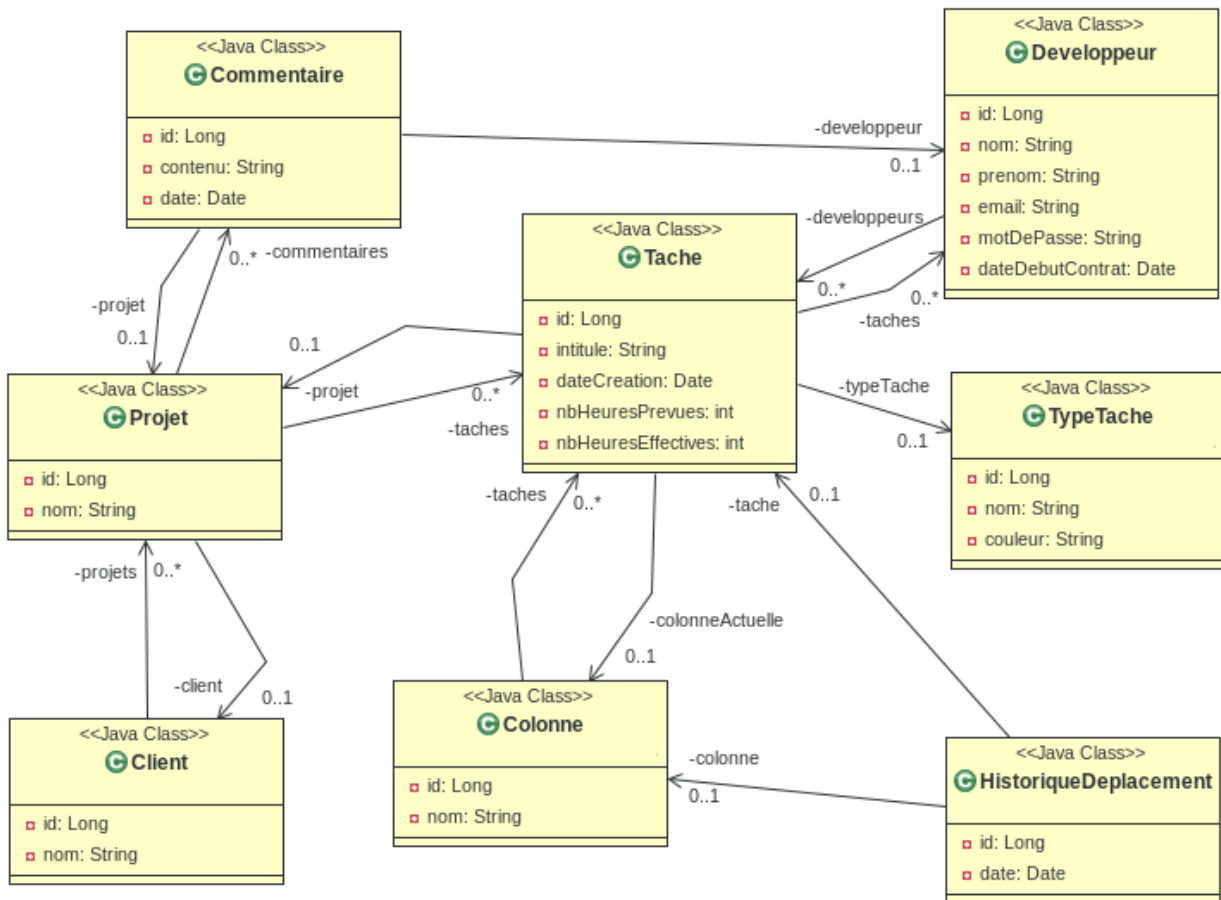
Cette application Spring Boot est construite en cinq couches:

- la couche présentation utilise des JSPs et les taglib de Spring et JSTL.
- la couche coordination est assurée par un contrôleur Spring.
- la couche service contient des services Spring.
- la couche DAO contient uniquement des interfaces.
- la couche persistance est assurée par une base de données MySQL nommée kanban.

## Diagramme de classes métier

---

Voici le diagramme de classes métier. Les classes de votre projet Maven doivent être identiques à celles présentées ci-après.



## Travail demandé

1. Importer le projet Maven disponible à l'adresse suivante:

<https://www.clelia.fr/Telecom/>

2. Créer une base MySQL ou H2 nommée kanban.

3. Ajouter les classes métier présentées sur le diagramme ci-dessus et annoter ces classes avec les annotations JPA, validation et Hibernate. Votre projet Maven contiendra une version png de votre diagramme de classes dans le répertoire doc.

4. Écrire les interfaces de la couche DAO.

5. Écrire les interfaces et les classes de la couche service.

6. Compléter le contrôleur Spring KanbanController pour votre application Web. Le contrôleur doit créer (s'ils n'existent pas) 1 client, 1 projet nommé « Projet 1 », 4 types de tâches (« Fonctionnalité », « Bug », « Spike », « Amélioration »), 4 colonnes (« À faire », « En cours », « À tester », « Terminé ») et 10 développeurs. Compléter la méthode accueil(), écrire les méthodes tacheGet(), tachePost(), déplacementGet(), developpeursGet() et initBinder.

7. Écrire les JSPs

8. Développer un service Web REST en complétant la classe KanbanRestController