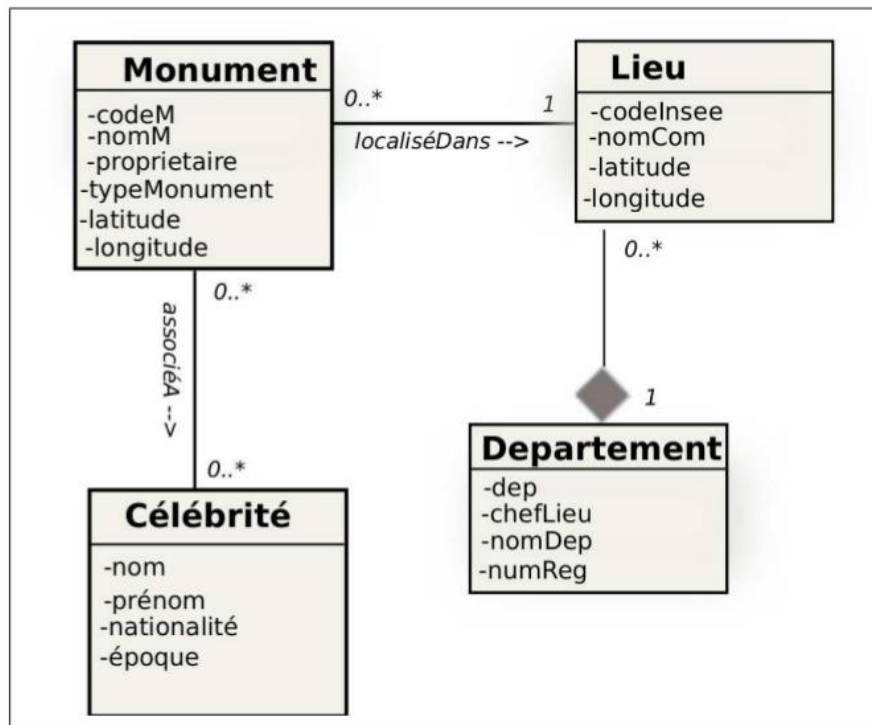
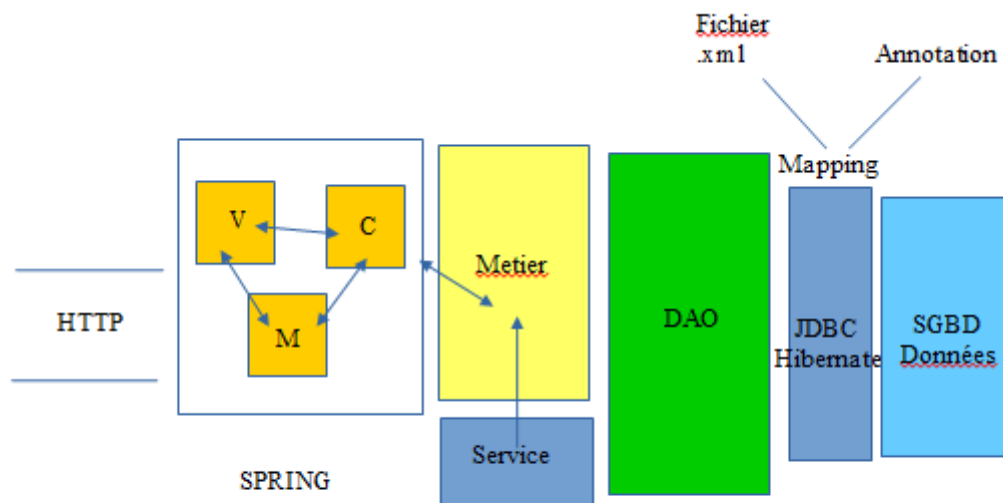


Le diagramme suivant, comme annoncé dans le cahier de charge du projet, donne le diagramme uml de la base de données:

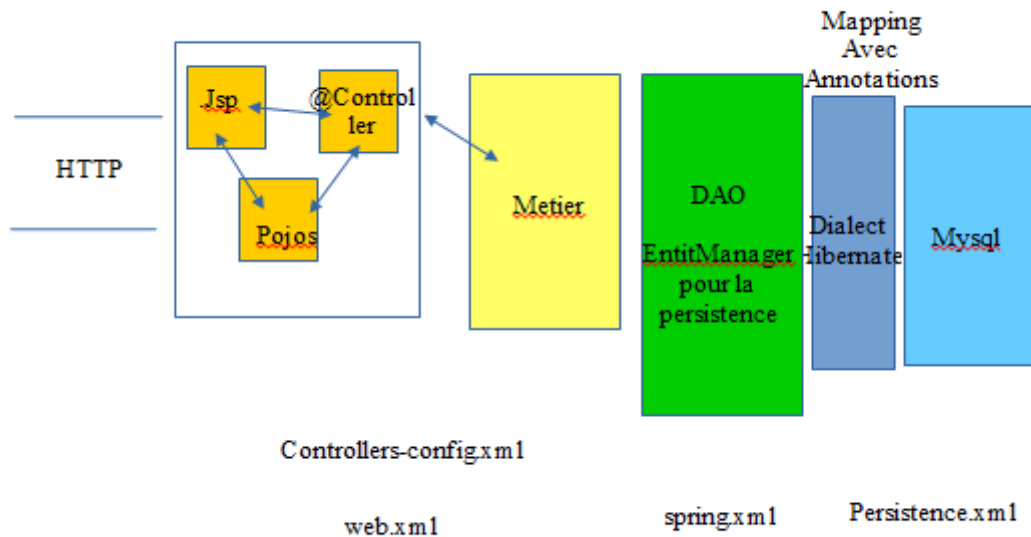


Donc, des pojos représentant les différentes classes ont été créés. Aussi pour l'association AssocieA qui est un pojo aussi.

L'architecture Maven-JEE-Spring à respecter est la suivante :



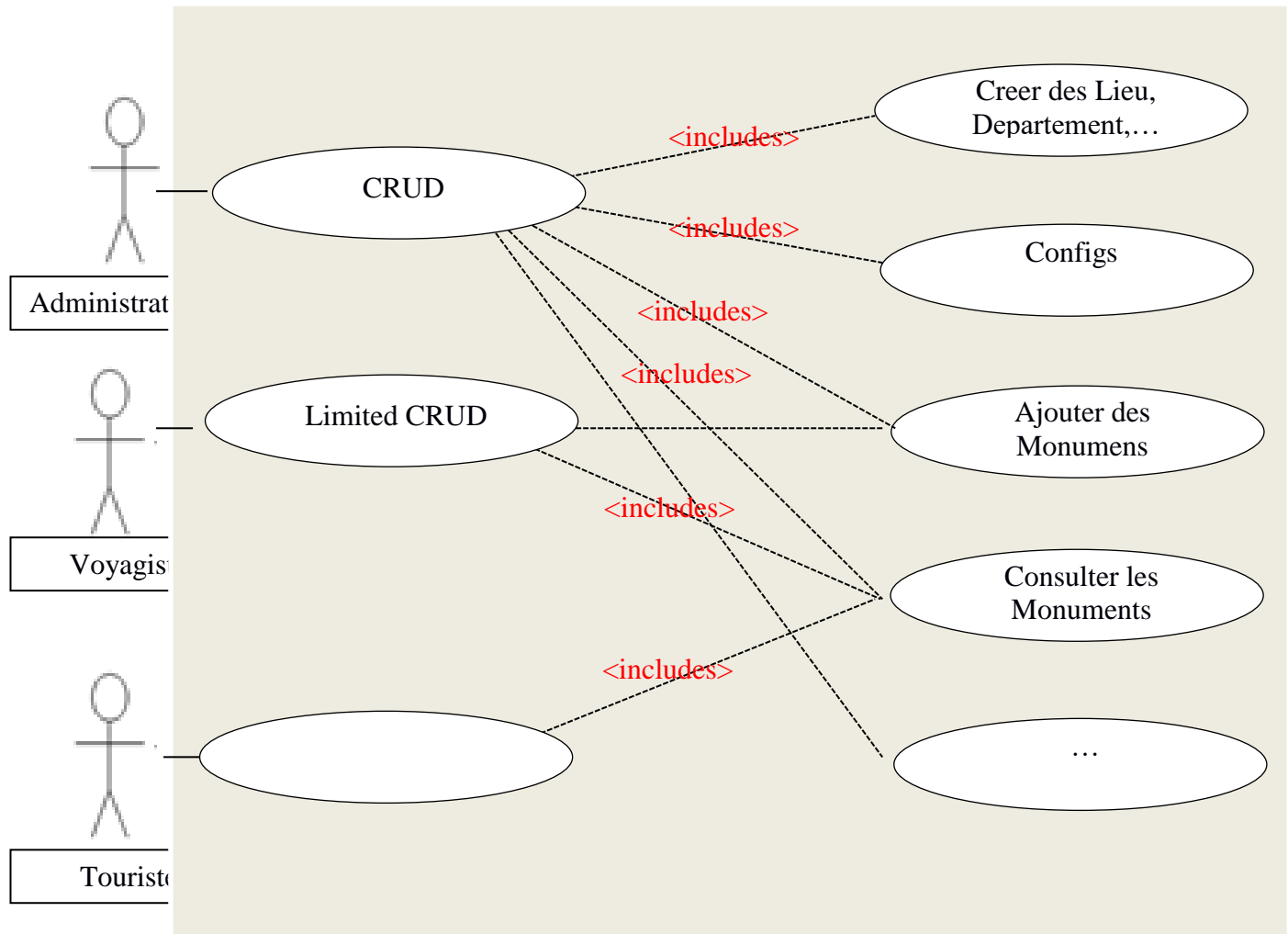
Je reprends le diagramme précédant pour décrire les options prise pour le projet.



En partant de la droite vers la gauche en passant les couches:

- Dans la base de données, j'ai les tables décrite précédemment dans le diagramme uml. Les tables ne sont pas encore peuplées mais elles le sont au fur et à mesure de création de nouvelles pages JSP qui ajoutent des tuples dans la bdd.
- Dans le mapping j'ai utilisé Hybernate et les annotations
- Dans le dao j'utilise l'EntityManager pour consulter la base de données
- La couche Metier est comme une de passage ou de securite
- L'architecture mvc est gerée par Spring pour mettre en relation tous les composants et faciliter l'echange de données
- les Pojos sont mappés
- J'ai pas encore beaucoup de vues mais j'ai déjà une qui marche
- dans le CRUD j'ai réussi à faire un exemple de chacun sauf le Update.

Voici un exemple de diagramme de cas d'utilisation de l'application en vue :



La vue sur le navigateur va changer suivant la personne qui se connecte.  
Aussi on peut représenter le diagramme de sequences suivant :

