|  |
| --- |
| **Travaux Pratiques**  **FI BDCC S4: Architecture distribuée JEE** TP : Gestion des comptes bancaires |

# Introduction :

La banque doit pouvoir atteindre plusieurs objectifs dans le cadre de la gestion quotidienne des comptes bancaires. Ces objectifs sont bien sûr complémentaires. Il lui faut :

Maîtriser les risques associés aux opérations bancaires réalisées sur le compte.

Améliorer la qualité des services rendus aux clients.

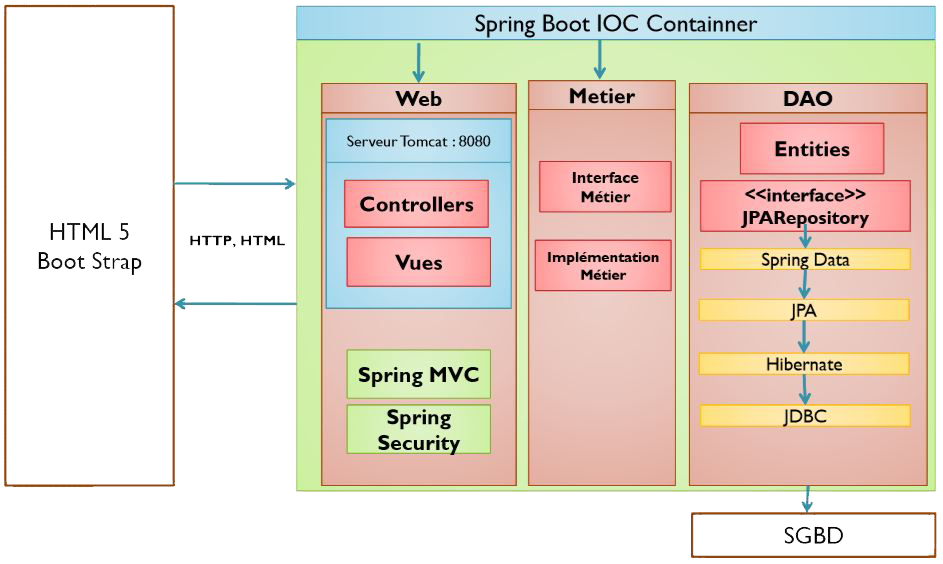
Rentabiliser ses comptes en équipant la clientèle des produits et services de la gamme.

# Enoncé :

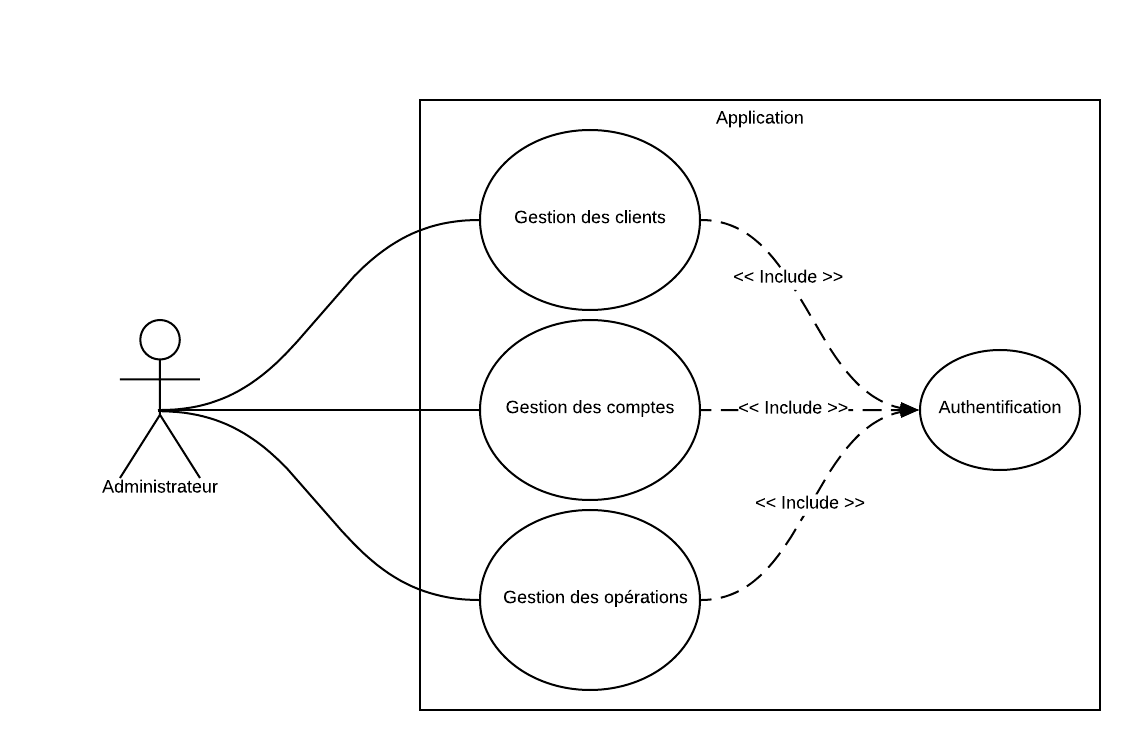
On souhaite créer une application qui permet de gérer des comptes bancaires en respectant les exigences fonctionnelles : Gérer les clients, Gérer les comptes, Gérer les opérations et les exigences techniques : les données sont stockées dans une base de données MySQL, l’application se compose de trois couches (la couche DAO, la couche Métier, la couche Web basée sur MVC coté serveur en utilisant Thymeleaf, la sécurité est basée sur Spring Security).

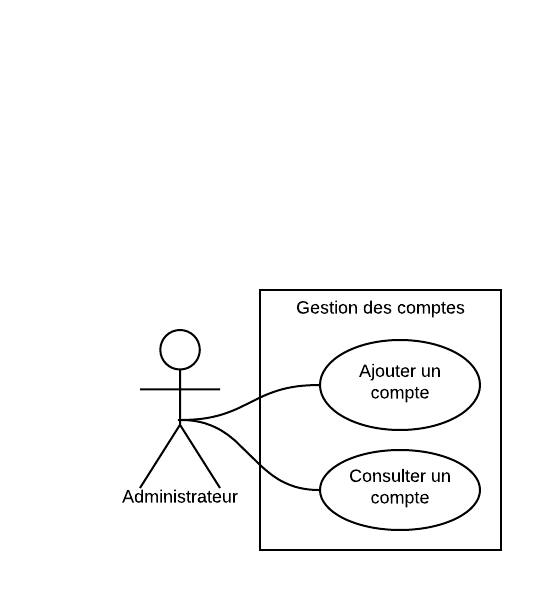
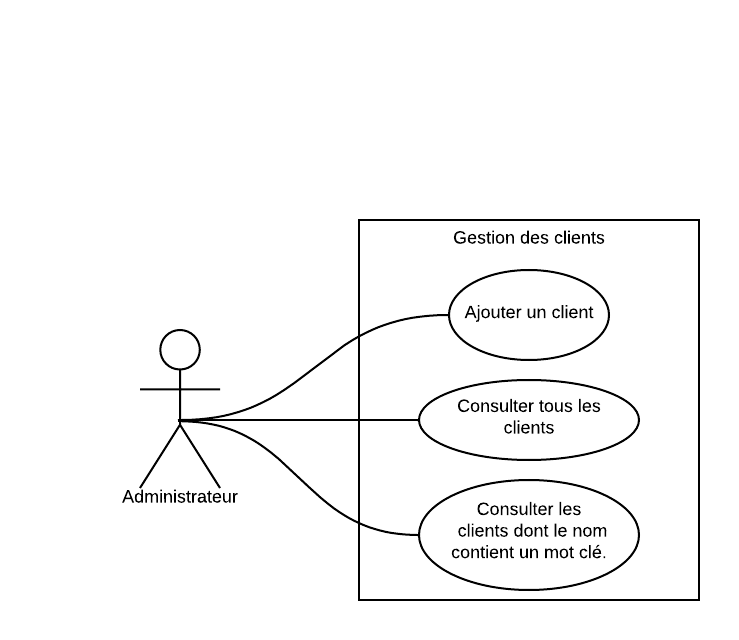
# Architecture et conception :

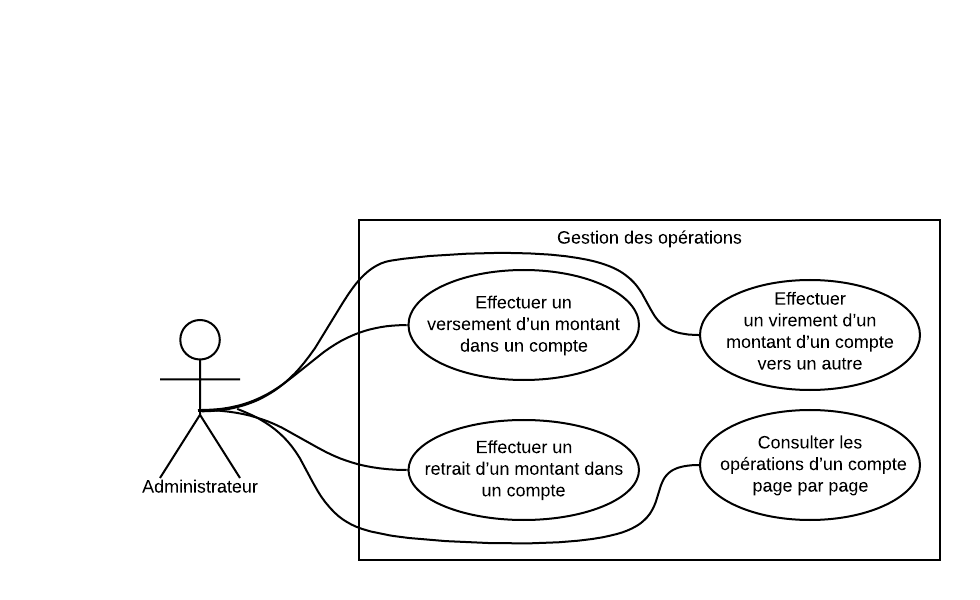
* + Architecture technique :



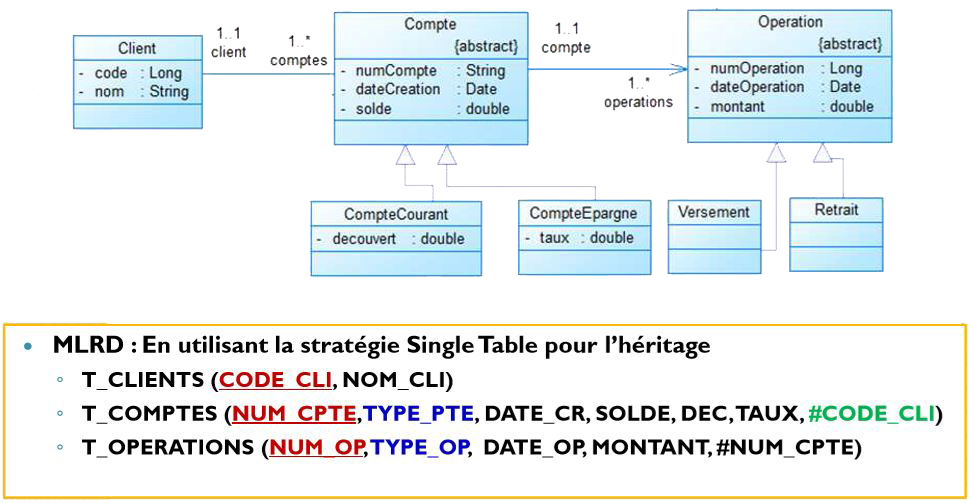
* + Diagramme de cas d’utilisation :





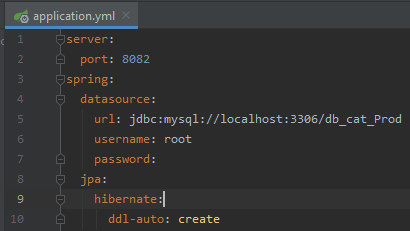
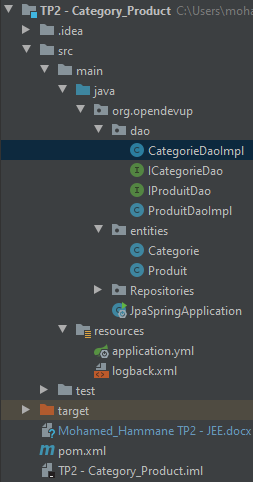


* + Diagramme de classes des entités et MLDR :

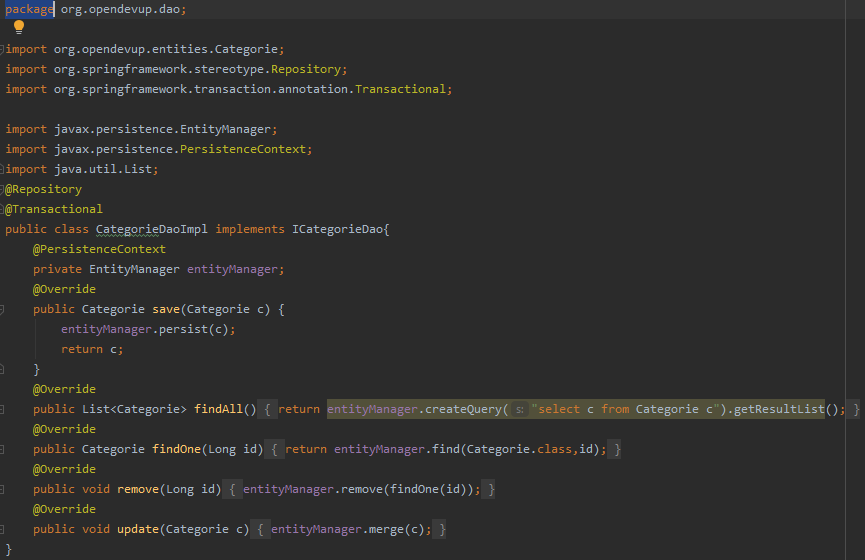


# Code source :

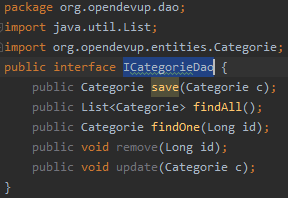
* + Structure du projet :



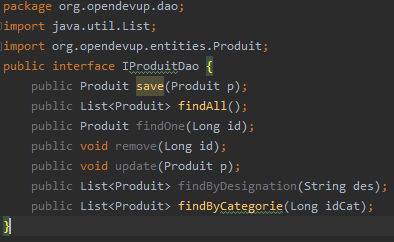
* + CategorieDaoImpl :



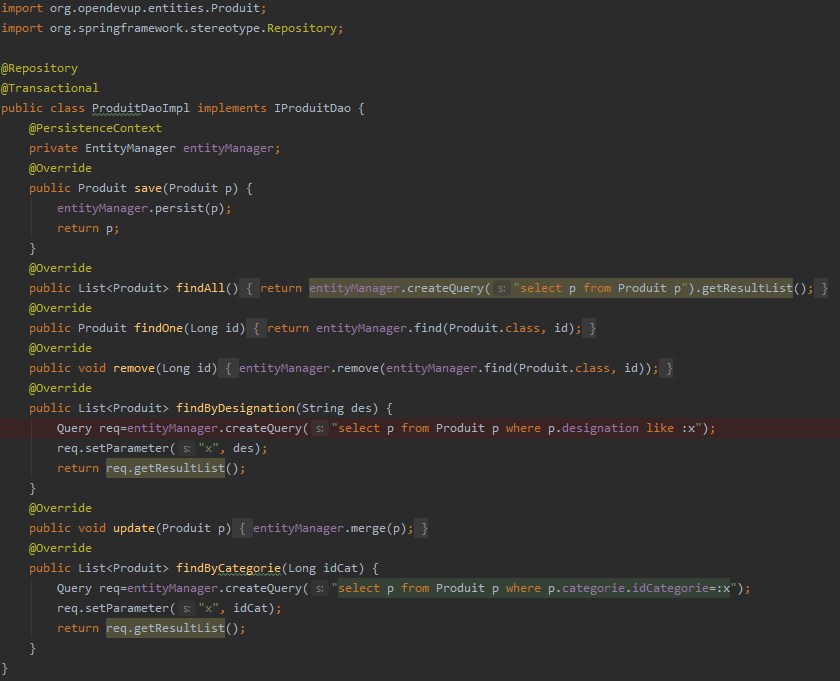
* + ICategorieDao :



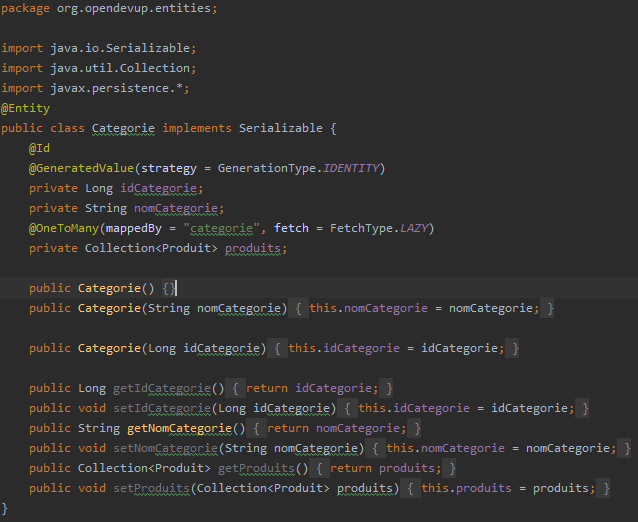
* + IProduitDao :



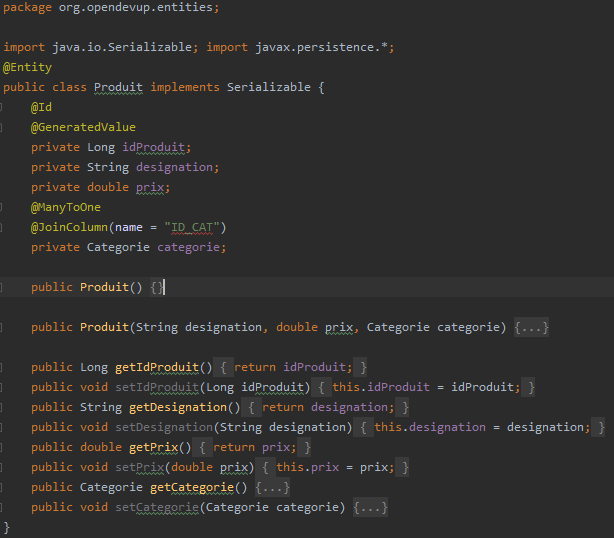
* + ProduitDaoImpl :



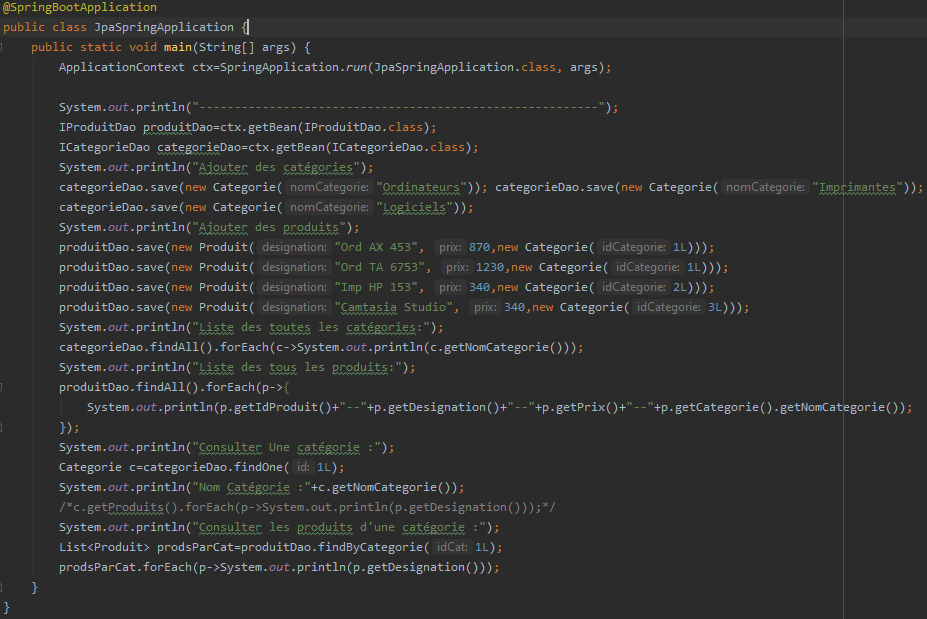
* + Catégorie :



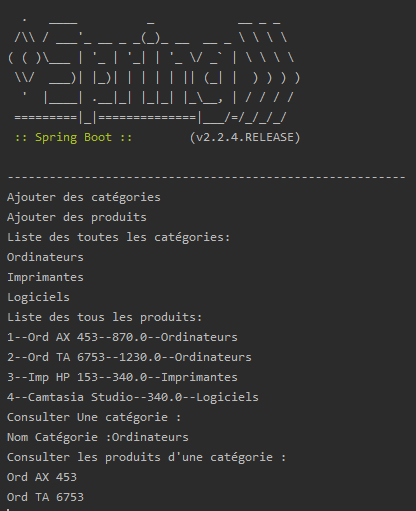
* + Produit :



* + JpaSpringApplication :



# Exécution :



# Conclusion :

Spring Boot vous offre la possibilité de démarrer un projet Spring et de le déployer facilement, tout en étant opérationnel pour la Production. ([Lien GitHub](https://github.com/Medomane/JEE/tree/master/TP2%20-%20Category_Product))