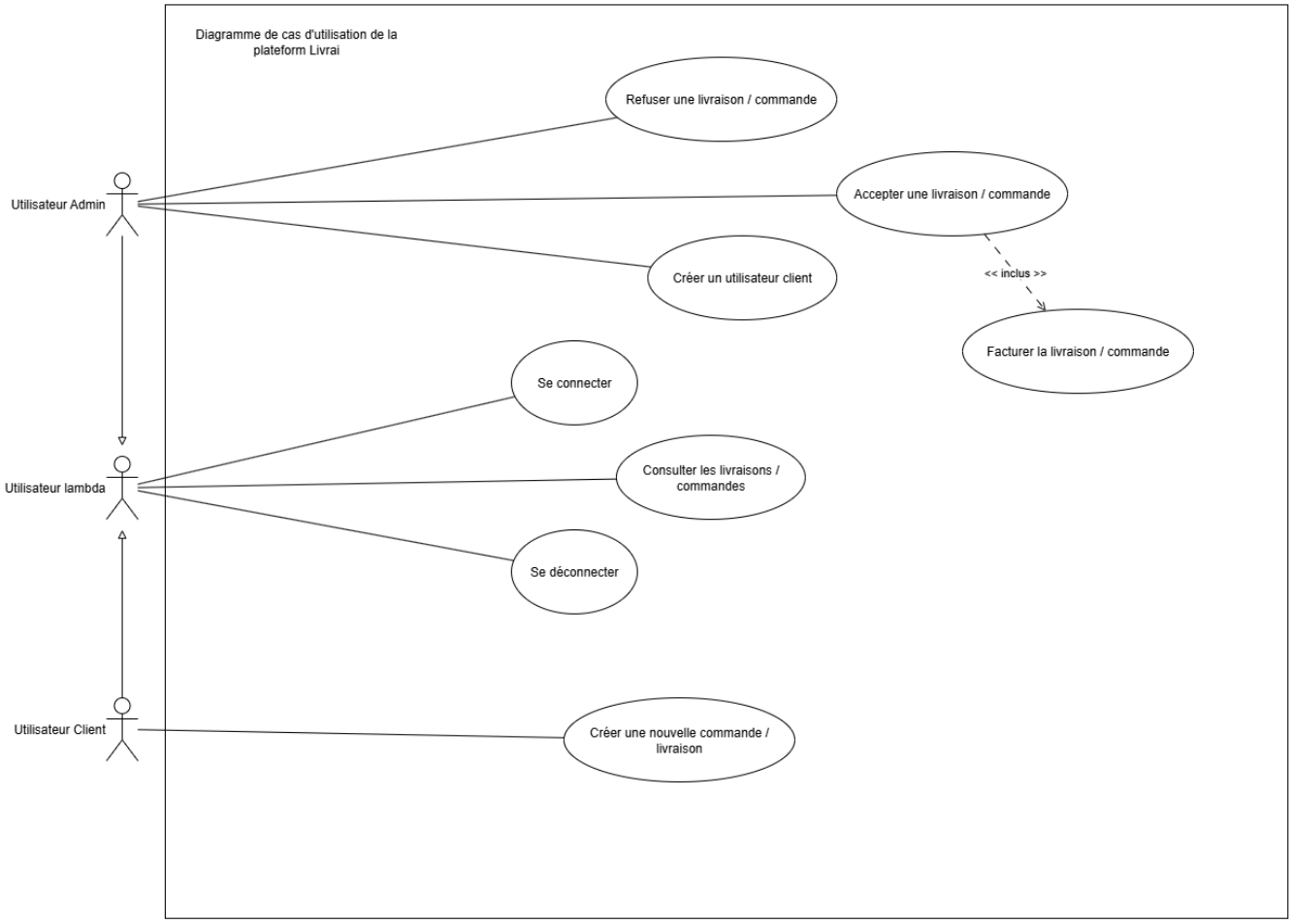
Audit de l’application existante

1. Contexte

Livrai est une entreprise de livraison, originairement spécialisée dans les marchandises en grosse quantité pour les professionnels. Désormais, l’entreprise entend s’étendre à tous types de colis et ce dans toute la France, ce qui l’amène à reconsidérer son application de sorte à accommoder l’ensemble des utilisateurs français

2. Fonctionnalités

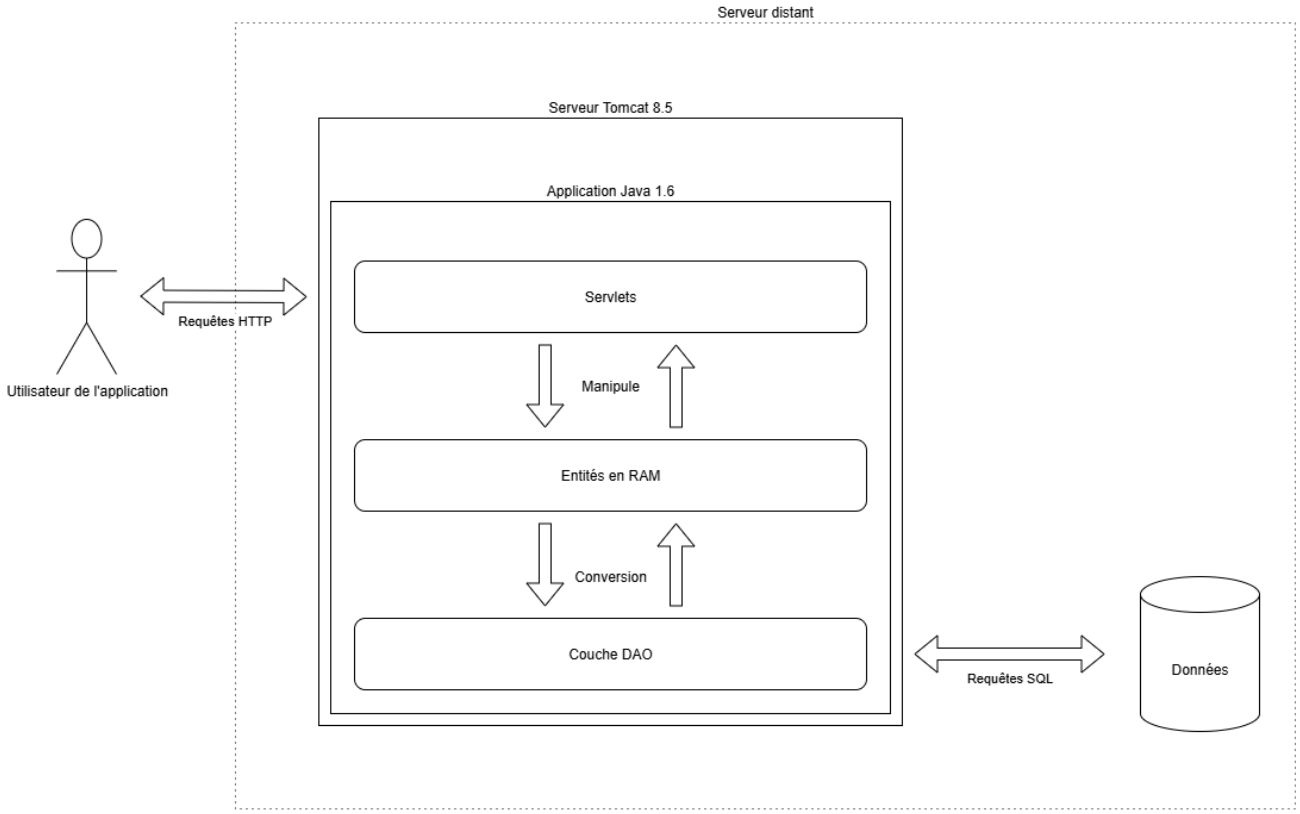
A l’heure actuelle, l’application dispose déjà de multiples fonctionnalités, que l’on peut schématiser via le diagramme de cas d’utilisation suivant :



3. Expérience utilisateur

En l'état, l'expérience utilisateur est simple et efficace, mais risque d'être peu engageante pour une application destinée à l'ensemble des Français. En effet, au-delà du design qui est assez sommaire et vieillissant, l'absence de responsiveness risque de poser soucis dans le cas où les utilisateurs souhaiteraient se servir de Livrai depuis leur téléphone mobile. Quelques soucis au niveau des entêtes des colonnes pour les tableaux d'affichage des livraisons / commandes dans la partie administrateur peuvent également créer de la confusion. Une fonctionnalité, permettant le refus des livraisons côté administrateur, semble également poser soucis et renvoie un code 405.

4. Description technique



5. Points forts et déficiences

- Points forts
    - L'application utilise exclusivement un code écrit en Java, ce qui peut aider à la maintenance dans le cas où l'équipe est déjà connaisseuse de ce langage.
    - Son design et son interface sommaire permet de rendre clair les fonctions et l'interaction avec l'application n'est pas entravée par de nombreux éléments visuels distrayants.
- Déficiences
    - L'utilisation de Java / Tomcat et l'absence de CSS moderne crée un design peu engageant face aux standards actuels. L'absence de responsiveness risque de porter préjudice dans le cas d'une sortie tout public
    - L'utilisation d'une si ancienne version de Java amène à des problèmes potentiels de performances / vulnérabilité du code

6. Conclusion

Il serait de bon ton de moderniser l'application, via entre autre l'utilisation de technologies plus modernes et reconnues pour leur fiabilité / performances telles que Spring / Angular et de revoir l'affichage de sorte à le rendre plus engageant pour les futurs utilisateurs qui pourront alors retrouver une interface plus en lien avec leurs habitudes et leur mode de navigation. L'ajout de Spring et la réalisation d'une API liée à une application front-end pourrait également permettre plusieurs façon d'interagir avec l'application, que cela soit via un back-office, des appels REST ou une application mobile réalisée, par exemple, avec Ionic (fortement compatible avec Angular).