



# Audit de l'application existante

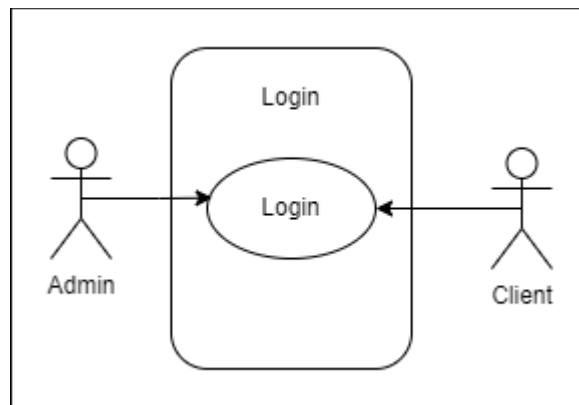
## 1. Contexte

- Livrai est une société de 150 employés proposant un service de livraison de marchandises en grande quantité pour les professionnels. Pour gérer nos clients et les livraisons, nous utilisons une application qui a été développée il y a plusieurs années.
- L'application permet à des clients de réserver des livraisons par l'intermédiaire du service commercial, ce dernier se charge de créer un compte et l'attribue au client.

## 2. Fonctionnalités

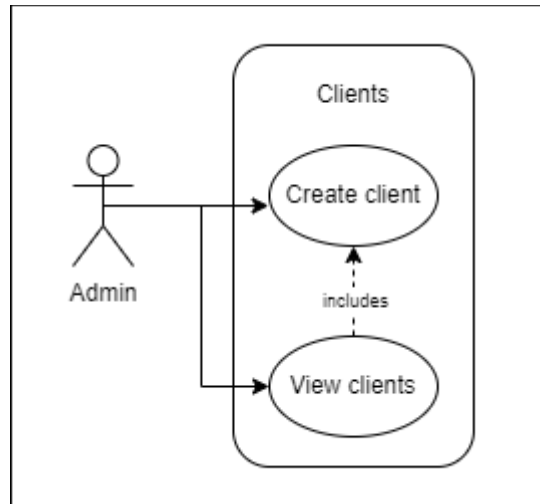
Voici les fonctionnalités illustrées sous forme de diagrammes de cas d'usages (le service commercial sera nommé Admin)

- Login



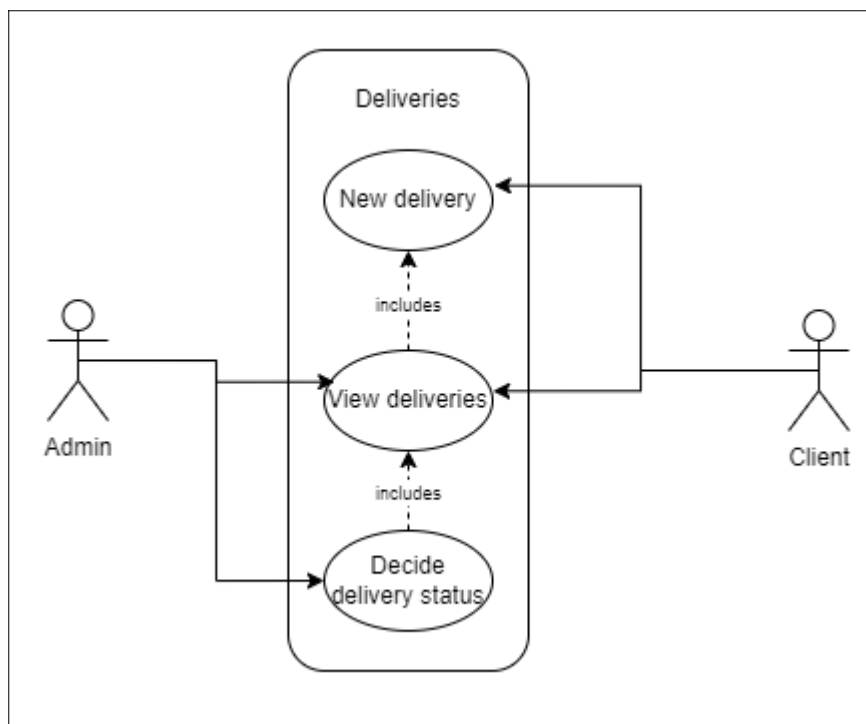
L'Admin et le Client peuvent se connecter grâce à la fonctionnalité Login.

- Clients



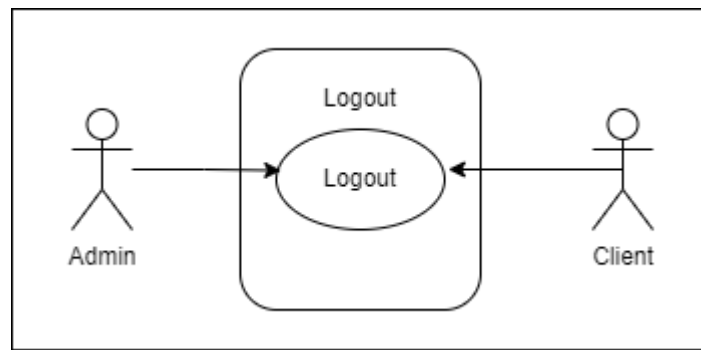
L'Admin a la possibilité de créer un client (Create client) depuis l'onglet client, uniquement réservé aux admins. Il peut également visualiser la liste des clients inscrits sur la plateforme (View clients), cette fonctionnalité implique la création de clients, d'où le "includes".

- Deliveries



Pour les livraisons, le Client peut commander une nouvelle livraison (New delivery), ce qui n'est pas le cas de l'Admin. Les deux acteurs peuvent en revanche consulter la liste des livraisons (View deliveries), ce qui implique la commande de livraisons de la part du Client. Pour l'Admin, il peut décider du statut d'une livraison (Decide delivery status), plusieurs choix s'offrent à lui: il peut la laisser en attente, l'accepter ou la refuser, une fois la décision prise, le statut de la commande est "Terminée". Cette fonctionnalité (Decide delivery status) implique la consultation des livraisons (View deliveries).

- Logout



Pour la fonctionnalité de déconnexion (Logout), il n'y a aucune différence entre les acteurs, le fonctionnement est le même.

### 3. Expérience utilisateur

Au lancement du projet, l'utilisateur se trouve sur la page de connexion, il entre son email et mot de passe. Lorsque l'utilisateur connecté est un Admin, il est dirigé vers la page principale du site et voit trois choix s'offrir à lui sur la barre de navigation: le premier choix, cliquer sur l'onglet "Clients", ajouter un nouveau client en saisissant l'email, nom et mot de passe, il peut aussi voir la liste de clients sous forme de tableau, les attributs des clients sont rangés en colonne (id, nom, email). Le deuxième choix est de cliquer sur l'onglet "Livraisons", l'Admin peut voir les livraisons à venir et passées, rangées aussi sous forme de tableau, il peut voir les attributs d'une livraison (id, volume, poids, prix et statut). Il a également la possibilité d'accepter ou de refuser une livraison à venir. Le troisième choix est la déconnexion en cliquant sur "Se déconnecter". Quant au client, il peut se connecter comme l'Admin pour se diriger vers la page principale du site et voir trois choix s'offrir à lui: le premier choix, cliquer sur l'onglet "Commande", commander une nouvelle livraison en saisissant le volume et le poids. Le deuxième choix est de cliquer sur l'onglet "Livraisons", le client peut voir ses livraisons à venir dont le statut est en attente, il peut aussi voir les livraisons passées. Le troisième choix est la déconnexion en cliquant sur "Se déconnecter" comme l'Admin.

### 4. Description technique

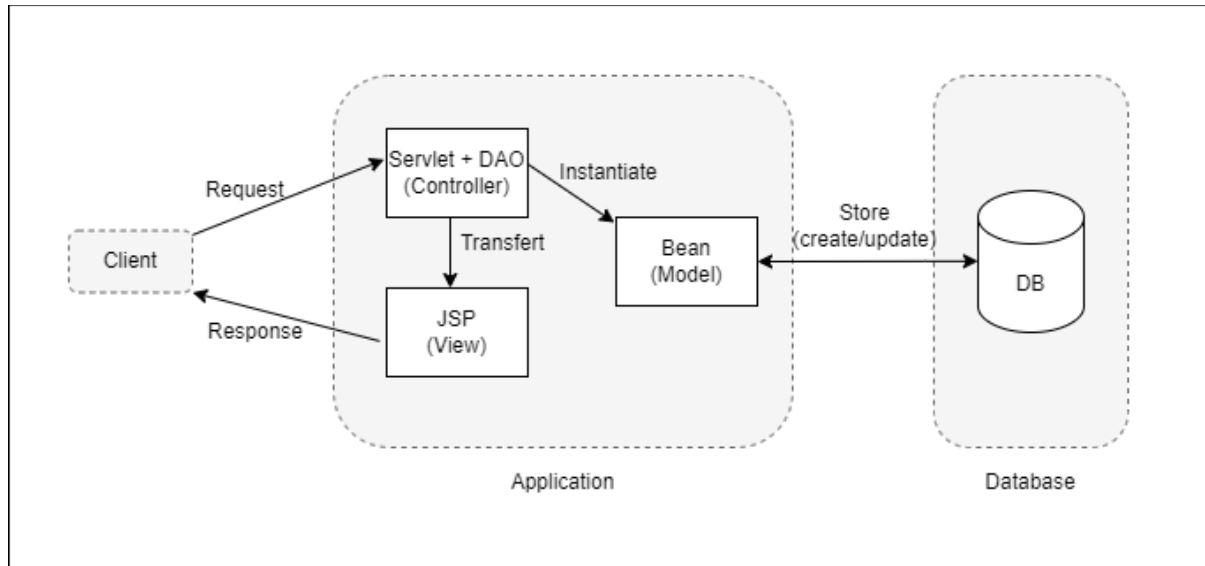
Architecture du projet divisée en deux parties. La première partie concerne le serveur en Java lui-même divisé en plusieurs couches: dossier avec un filtre d'authentification (classe AuthenticationFilter), dossier de beans (classes Delivery et User), dossier de dao (classes AbstractDao, DeliveryDao et UserDao), dossier de servlet (classes ClientsServlet, CommandServlet, DeliveriesServlet, DeliveryServlet, HomeServlet, LoginServlet, LogoutServlet). La deuxième partie contient les pages du site, dont le langage est le jsp (JavaServer Pages) qui est une technologie de développement côté serveur permettant la création de pages web dynamiques.

Versions des langages et frameworks:

- javax.servlet-api: 3.0.1
- javax.servlet:1.2

- taglibs: 1.1.2
- mysql-connector-java: 8.0.32
- junit: 4.11

Pour plus de précisions, on peut constater que ce projet reprend le modèle d'architecture MVC (Modèle-Vue-Contrôleur) comme l'illustre le schéma suivant:



Le client effectue une requête auprès du contrôleur, celui-ci instancie le/les modèle(s) impliqués dans la requête. Le modèle est ajouté/mis à jour dans la base de données selon la nature de la requête. Ensuite le contrôleur transfère les résultats dans la vue, elle renvoie la réponse de la requête vers le client.

## 5. Points forts et déficiences

Points forts:

- Site rapide
- Authentification (login) fonctionne correctement
- Ajout et visualisation des clients (Admin)
- Acceptation de la livraison fonctionne (Admin)
- Commande de livraison fonctionne (Client)
- Visualisation de livraison à venir correcte (Client)
- Déconnexion fonctionne correctement
- Navigation au clavier (tabulation)

## Déficiences:

- Formulaire de connexion non centré, situé en haut à gauche de la fenêtre du navigateur.
- Barre de navigation peu esthétique, onglets de navigation blancs sur fond violet.
- Pour les livraisons passées (connecté en tant que Client), les valeurs sont placées dans les mauvaises cases, les prix sont dans les cases "Poids", les statuts sont dans les cases prix. Les cases "Statut" sont vides.
- Pour les livraisons passées (connecté en tant qu'Admin), plusieurs erreurs sont aussi à noter: à la place du "Volume", on voit la valeur du nom du client, à la place du "Poids", c'est la valeur du "Volume" qui apparaît. Dans la case prix, on voit en effet le prix mais aussi la valeur du poids à côté. Il y a donc un décalage dans l'affichage des valeurs entre le volume, le poids et le prix.
- Pour les livraisons à venir (connecté en tant qu'Admin), on remarque les mêmes erreurs: le nom du client apparaît dans la case "Volume" alors ça devrait être le titre "Nom" (nom d'utilisateur), dans la case "Poids", c'est la valeur du volume qui y est affiché. Pour la valeur "Statut", c'est la valeur du poids qui apparaît. On voit également des colonnes sans titre mais avec du contenu, la première avec les valeurs "En attente", on suppose que cette colonne doit avoir pour titre statut, la deuxième colonne sans titre, il y a les boutons "Accepter" et "Refuser".
- Le refus d'une livraison à venir (cliquer sur le bouton "Refuser") ne fonctionne pas, voici le message d'erreur: "Etat HTTP 405 – Méthode non autorisée, Type: Rapport d'état, message: La méthode HTTP POST n'est pas supportée par cette URL, description: La méthode HTTP spécifiée n'est pas autorisée pour la ressource demandée."

## 6. Conclusion

Cet audit présente les fonctionnalités de l'application avec des diagrammes illustrant les acteurs et leurs possibilités. Suite à l'analyse de l'architecture du projet, on déduit que celui-ci reprend le modèle MVC (Modèle-Vue-Contrôleur) avec les langages JSP pour l'affichage et Java pour la logique. L'application répond rapidement aux demandes de l'utilisateur, cependant, elle présente plusieurs défauts d'affichage et une erreur de fonctionnalité. Il est préférable de refaire ce projet avec des technologies modernes comme Angular-Spring.