LiVrai

Audit de l’application existante

1. Contexte

Activité de l'entreprise

LiVrai est une entreprise spécialisée dans la livraison de marchandises en grande quantité pour les professionnels sur l'ensemble du territoire national.

Objectif de l'application

L'application actuelle est un CRM (Customer Relationship Management) pour la gestion des clients et de leurs commandes de livraison. Le client final a un usage limité des fonctionnalités, qui sont gérées majoritairement au niveau du commercial (administrateur).

2. Fonctionnalités

L'application actuelle est structurée autour de trois grands axes :

Fonctionnalités générales

- Connexion et déconnexion des utilisateurs
- Gestion des sessions utilisateur

Gestion des comptes clients

- Consultation de la liste des clients (admin)
- Ajout d'un nouveau client (admin)
- Mise à jour des informations client (admin)

Gestion des livraisons

- Création d'une demande de livraison (client)
- Consultation de l'historique des livraisons (client et admin)
- Acceptation ou rejet d'une livraison (admin)
- Facturation d'une livraison terminée (admin)
- Suppression d'une livraison (admin)

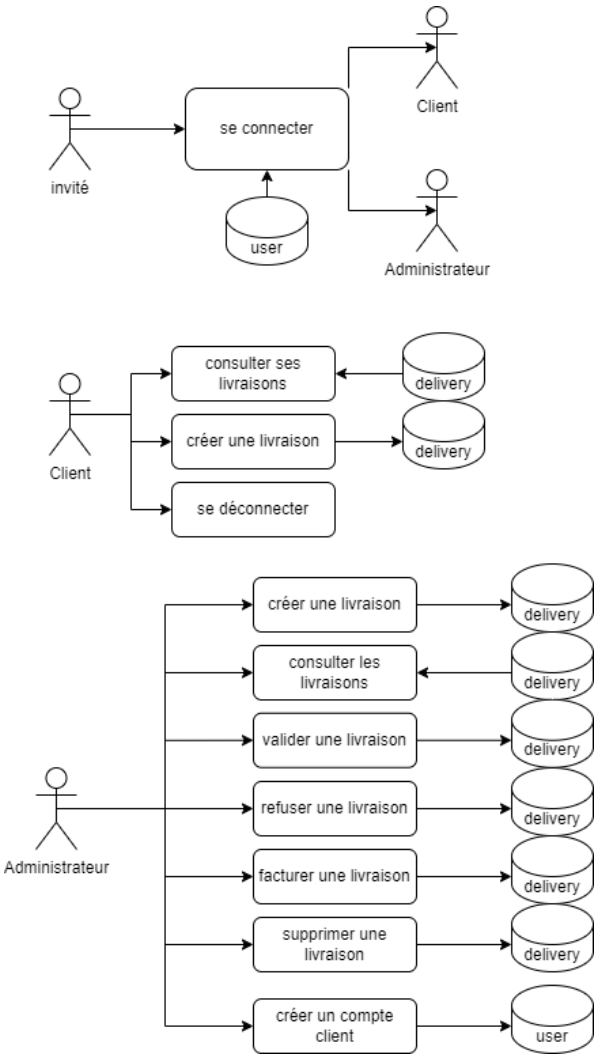


Schéma fonctionnel de l'application *Livrai*

3. Expérience utilisateur

L'expérience utilisateur présente quelques aspects positifs :

- Interface minimaliste et formulaires simples
- Navigation fluide et simple depuis la barre de navigation

Mais l'expérience générale est assez médiocre, plusieurs points sont à améliorer :

- Manque de design moderne (JSP vieillissant, css massif, pas de framework moderne)
- Absence de notifications (modales de confirmation ou d'erreur)
- Absence de mise à jour réactive de l'affichage, rechargement de l'ensemble
- Site non responsive, css fixe non flexible
- Site sans accessibilité ou ergonomie (label, navigation clavier, focus)

4. Description technique

Le projet est une application web Java EE construite avec Maven et packagée sous forme de fichier war pour être déployée sur un serveur d'application type tomcat. L'ensemble des dépendances est ancien, certains éléments fortement dépréciés ou obsolètes, induisant des problèmes de compatibilité, de maintenance et de sécurité.

Architecture 3-tiers

Front-End (Vue JSP)

- JSP (JSTL 1.2) [obsolète]
- Taglibs Standard 1.1.2 [déprécié]
- CSS statique

Back-End (Servlets, DAO, JDBC)

- Java EE 1.6 [obsolète]
- Servlet API 3.0.1 [version ancienne]
- DAO pour la communication avec la base de données

Base de données (MySQL)

- MySQL 8.0+
- JDBC via MySQL Connector 8.0.32 [obsolete]

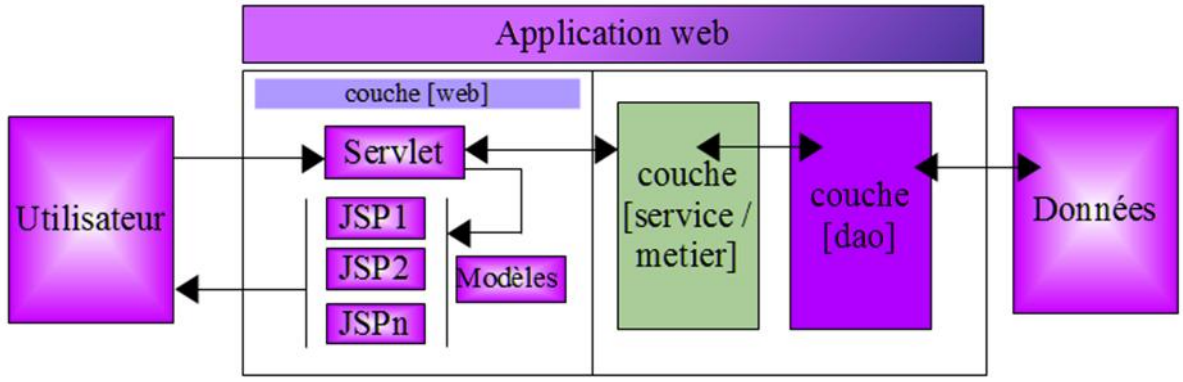
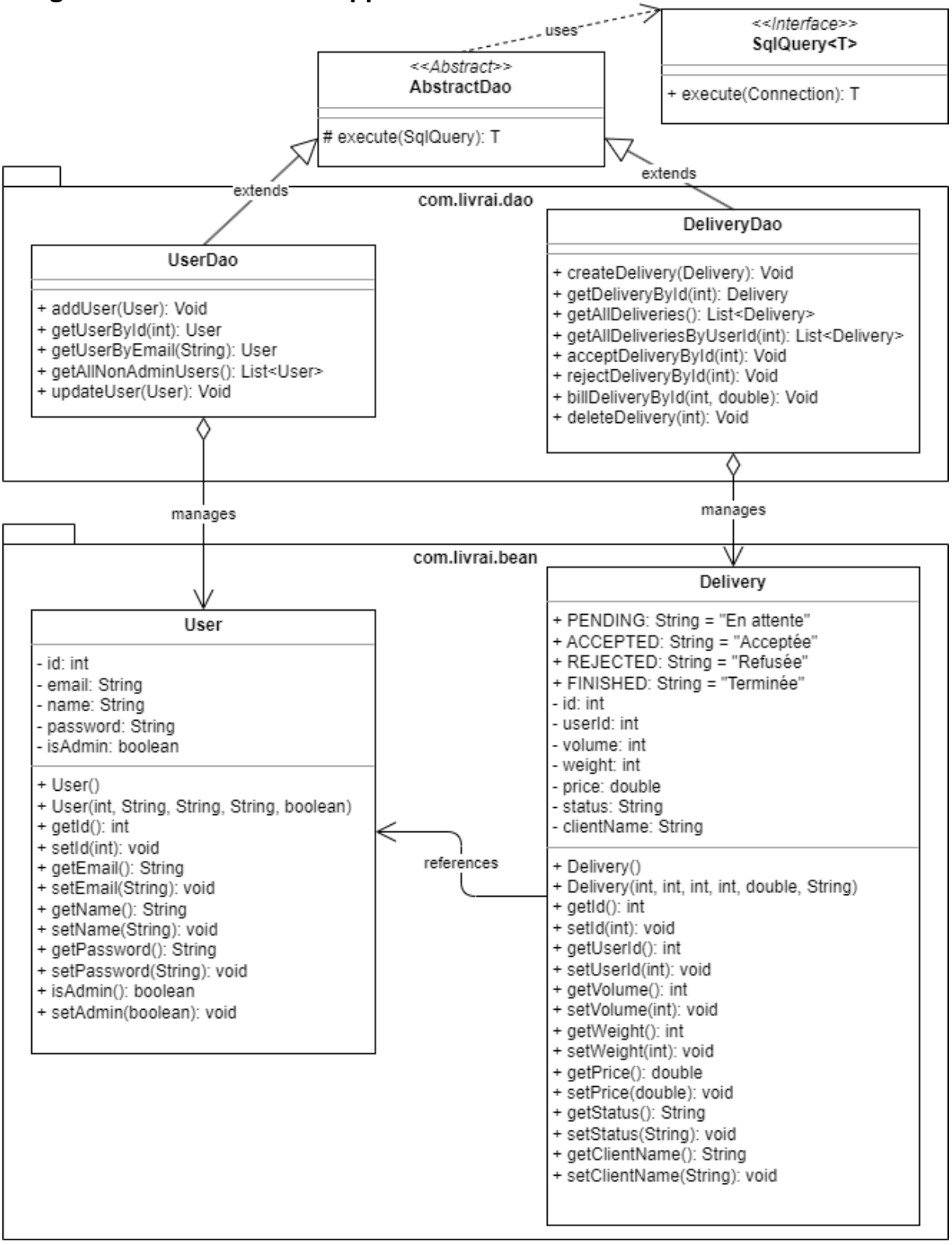


Diagramme de classes de l'appli *LiVrai*



5. Points forts et déficiences

Points forts

- **Fonctionnalité**

L'application actuelle est fonctionnelle et stable et répond globalement au besoin.

- **Architecture**

Bonne séparation des responsabilités, architecture des DAO selon les principes SOLID.

- **Sécurisation**

Utilisation de preparedStatement dans les DAO, réduisant le risque d'injections SQL classiques.

Déficiences

- **Obsolescence technologique**

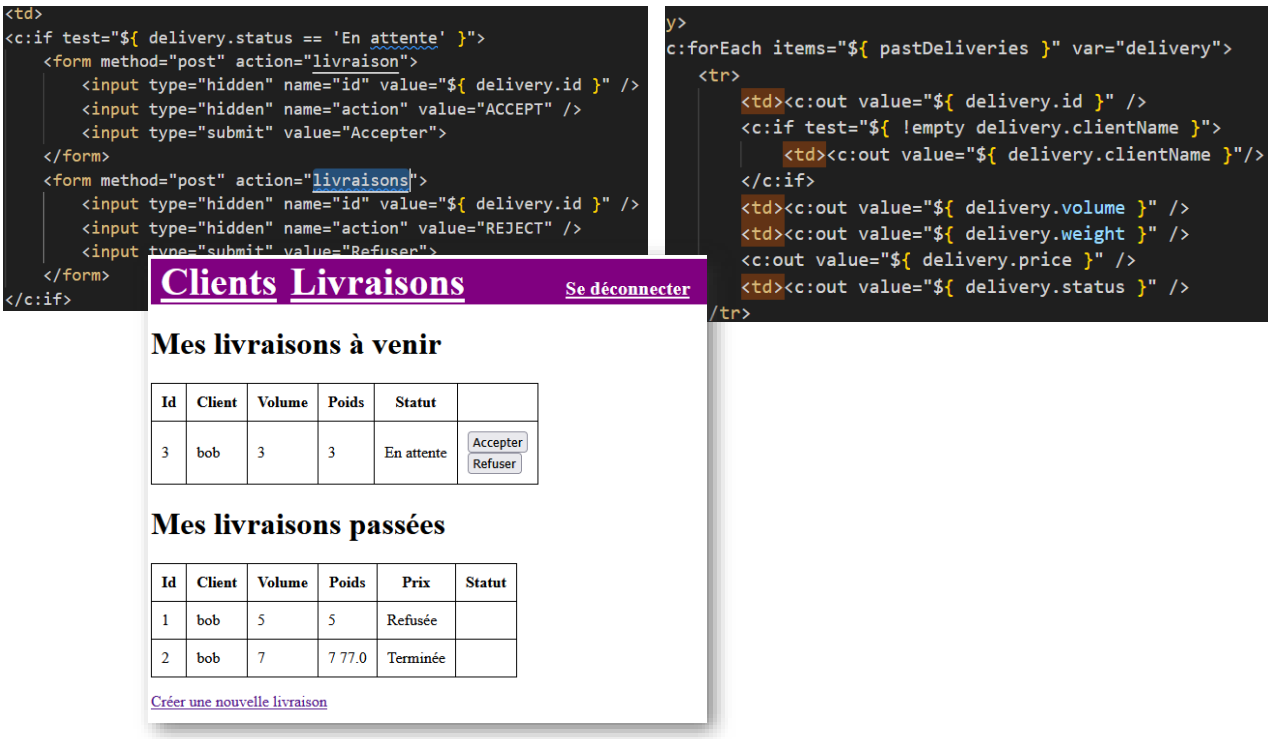
Développement en Java 6 obsolète, non maintenu.
Utilisation de Servlet API 3.0, ancienne version.

- **Failles de sécurité critiques**

Stockage du rôle Admin en local storage, peut être modifié par injection (escalade de privilèges).
Endpoints non sécurisés (absence d'authentification en en-tête, vulnérable aux attaques CSRF).
Base permissive (un poids ou un volume négatif sont acceptés, risque d'incohérence)

- **Bugs fonctionnels et UX obsolète**

Bug d'affichage prix/statut dans le tableau des livraisons passées, corriger <***td***>
Bug fonctionnel (erreur 405-Method not allowed) pour le refus de commande, corriger *livraisons***s**
Utilisation du nom réservé `user` dans la base de données, préférer ***users*** pour éviter les conflits.
Mauvaise gestion des erreurs (absence de page 404, information utilisateur)
Interface utilisateur non responsive, peu accessible et vieillissante.



6. Conclusion

L'application **LiVrai CRM** est une solution fonctionnelle et stable qui répond aux besoins métier de gestion des livraisons pour les professionnels. Cependant, elle repose actuellement sur une architecture **obsolète** et souffre de **vulnérabilités critiques en matière de sécurité et d'expérience utilisateur**.
Avant de se pencher sur l'évolution des fonctionnalités existantes et l'ajout de nouveaux axes de développement, des modifications majeures sont à mettre en place sur le socle de l'application.

Axes d'amélioration prioritaires

1. **Mise à jour technologique et compatibilité**
   - Migration vers **Java 17** (LTS)et **Jakarta EE** pour garantir un support et une maintenance à long terme, ou **Java 21** pour une modernisation totale.
   - Mise à niveau vers une **Servlet API moderne (4.0 ou 5.0)** pour bénéficier des nouvelles fonctionnalités et améliorations de sécurité.
2. **Renforcement de la sécurité**
   - Utilisation d'une **gestion sécurisée des sessions et des rôles** ou mise en place de JWT.
   - Ajout d'un **mécanisme d'authentification sécurisé** (ex. Bearer Token en en-tête des requêtes).
   - Validation stricte des entrées utilisateur pour empêcher **les incohérences.**
3. **Modernisation de l'interface utilisateur**
   - Adoption d'un **framework frontend moderne** (Angular, React, Vue.js) pour améliorer la réactivité et l'accessibilité.
   - Ajout de **notifications et modales interactives** pour un meilleur retour utilisateur.
   - Rendre l'interface **responsive et accessible** (navigation clavier, labels, focus).
4. **Correction des bugs et amélioration fonctionnelle**
   - Correction du **bug 405 lors du refus de commande**.
   - Affichage dynamique et correct des **statuts et prix** dans le tableau d'historique des livraisons.
   - Ajout d'une **page d'erreur** pour mieux gérer les erreurs et améliorer l'UX.