



Audit de l'application existante

1. Contexte

Livrai est une société assurant un service de livraison de marchandises en grande quantité pour professionnels à l'échelle nationale.

L'application auditée est celle utilisée par la société pour gérer les clients de l'entreprise et les commandes que ces derniers effectuent, ainsi que la facturation des-dites commandes.

2. Fonctionnalités

Fonctionnalités générales :

- Possibilité de se connecter et de se déconnecter d'un compte existant ;
- Navigation entre les différentes pages accessibles via une barre de navigation (pages clients et livraisons pour les admin, pages commandes et livraisons pour les non-admin ;

Fonctionnalités relatives à la gestion des comptes utilisateurs (pour les admin) :

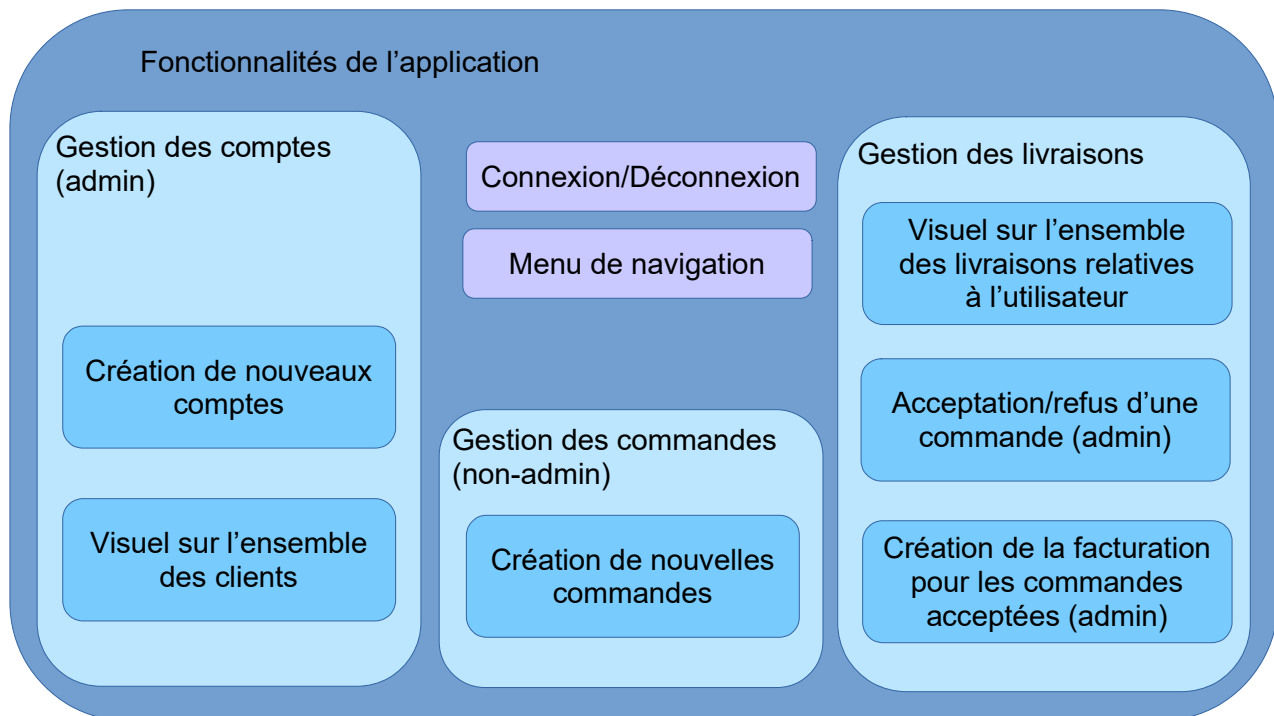
- La création de compte (réservée aux utilisateurs admin) ;
- Le visuel de l'ensemble des clients avec leur id, nom et adresse mail ;

Fonctionnalités relatives à la gestion des commandes (pour les comptes de clients) :

- Création de commande avec poids et volume associé ;

Fonctionnalités relatives à la gestion des livraisons :

- Le visuel sur les livraisons effectuées par son compte (pour les non-admin) ou l'ensemble des comptes (pour les admins), qu'elles soient en cours ou passées ;
- La possibilité pour les admin d'accepter (ce qui la fait passer de « en attente » à « passée ») ou refuser une commande ;
- En cas d'acceptation d'une livraison par un admin, la réalisation de la facturation associée à cette livraison ;



3. Expérience utilisateur

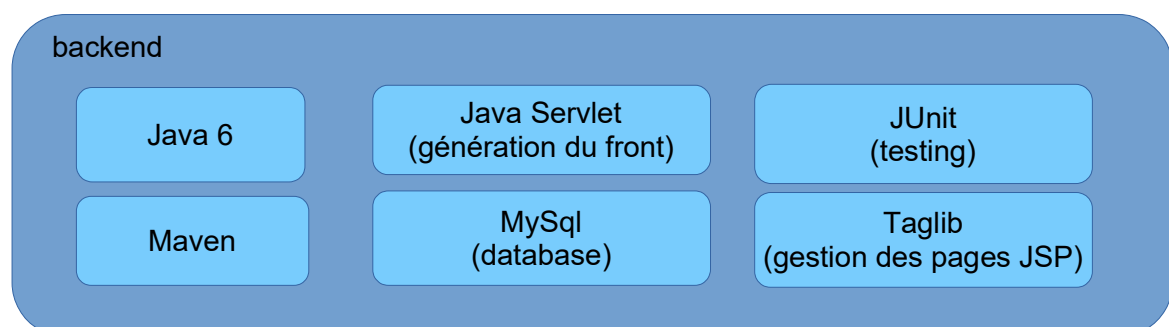
La prise en main de l'application est facile et la navigation entre les pages se réalise de manière fluide.

Cependant, l'application présente des bugs visuels (données générées dans de mauvaises colonnes dans les tableaux de livraisons passées et à venir) et certaines fonctionnalités n'ont pas été implémentées ou ne fonctionnent plus (refuser une livraison pour les admin en cliquant sur le bouton « Refuser », qui renvoie une erreur 405).

L'envoi de données n'est également pas vérifié par l'application, ce qui peut provoquer de nouveaux bugs (erreur 500 si des champs restent vides en cas de création de commandes) ou des comportements indésirables (création d'utilisateurs avec des champs vides, quantités négatives considérées comme possibles par l'application). La page d'erreur affichée en cas de problème n'est pas ailleurs pas explicite pour les clients (affichage du seul message d'erreur automatique du serveur).

4. Description technique

L'application a été réalisée avec Java 6 (1.6) avec Maven, et utilise l'API Java Servlet couplé à des pages JSP pour générer un rendu HTML de l'application pour le frontend à partir des requêtes serveur. Il n'y a donc pas de séparation stricte entre frontend et backend dans l'application. Le stockage des données se fait via une base de donnée MySQL. L'application intègre également JUnit comme librairie de test.



5. Points forts et déficiences

Points forts de l'application :

- Rapide ;
- Intuitive à prendre en main.

Déficiences de l'application :

- Technologie Java 6 ancienne et pas compatible avec l'évolution du langage ;
- Pas de séparation entre backend et frontend dans l'application, le frontend est généré par servlet Java ;
- Pas de vérification des données envoyées via les formulaires (quantités négatives, données vides pour la création de nouveaux users, ...) ;
- Certains endpoints sont manquants (erreur 405 si refus d'une commande faute de méthode POST correspondante dans l'API) ;
- Présence de bugs dans les tableaux de commandes ou de users ;
- Problème de sécurisation des routes (accès aux pages admin pour les non-admin et non-admin pour les admin en modifiant l'url) ;
- Problème de sécurité des données : des informations sensibles sont stockées sans cryptage dans la base de donnée (comme les mots de passe).

6. Conclusion

En son état actuel l'application Livrai permet de réaliser efficacement la majorité des fonctionnalités attendues d'elle, mais elle souffre de nombreux bugs et d'une technologie ancienne et dépassée. Une refonte de l'application est recommandée pour assurer une mise à jour du langage employé, une correction des nombreux bugs et le respect de bonnes pratiques (telle que la séparation du frontend et du backend sur la base de frameworks dédiés).