

Audit de l'Application Existante

Sommaire

1. Contexte
2. Fonctionnalités
3. Expérience Utilisateur
4. Description Technique
5. Points Forts et Déficiences
6. Conclusion

Contexte

LiVrai est une entreprise spécialisée dans la livraison de marchandises en grande quantité pour les professionnels. Actuellement, l'application de gestion des clients et des livraisons ne répond plus aux besoins croissants de l'entreprise et de ses clients en raison de l'expansion de son offre

Fonctionnalités

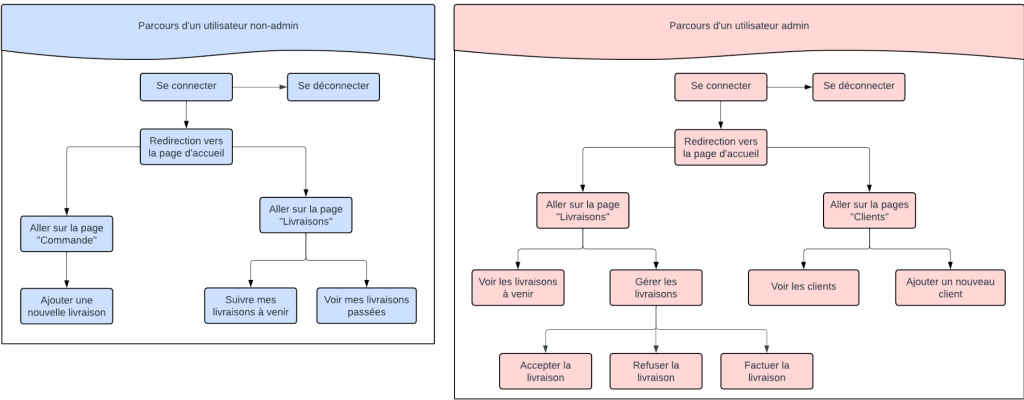
Fonctionnalités Générales :

Type d'utilisateur	Fonctionnalités
Client / Administrateur	<ul style="list-style-type: none">• Se connecter• Se déconnecter

Gestion des Comptes Clients :

Type d'utilisateur	Fonctionnalités
Client	<ul style="list-style-type: none">• Commander une nouvelle livraison• Afficher toutes les livraisons à venir de l'utilisateur• Afficher toutes les livraisons passées de l'utilisateur
Administrateur	<ul style="list-style-type: none">• Ajouter un nouveau client• Afficher les livraisons à venir de chaque client• Gérer les livraisons :<ul style="list-style-type: none">◦ Accepter la livraison◦ Rejeter la livraison◦ Facturer la livraison• Afficher les livraisons passées de chaque client

Schéma :



Expérience Utilisateur

Page(s)	Type d'utilisateur(s)	Problème	Amélioration
Toutes	Utilisateur / Administrateur	L'interface actuelle est trop simpliste et manque d'attrait visuel, ce qui peut décourager les utilisateurs	
Toutes	Utilisateur / Administrateur	Les erreurs sont présentées sous forme de rapports qui interrompent la navigation dans l'application, rendant leur compréhension difficile pour l'utilisateur.	Présenter les erreurs sous forme de messages clairs et non intrusifs, sans interrompre la navigation, afin de faciliter leur compréhension et permettre à l'utilisateur de continuer à utiliser l'application.

Accueil	Utilisateur / Administrateur	La page est quasiment vide et n'apporte aucune information utile.	Réorienter directement vers une page où il y a plus d'interactions
Clients	Administrateur	Les champs de saisie ne sont pas validés correctement, ce qui permet d'entrer des données erronées ou de laisser les champs vides.	/
Livraisons / Clients	Utilisateur / Administrateur	Lorsque un tableau est long, il faut scroller jusqu'en bas de celui-ci pour accéder à l'élément en suivant.	Ajout d'un système de pagination à un certain nombre d'éléments dans un tableau.
Livraisons / Clients	Utilisateur / Administrateur	Difficulté à trouver un élément lorsqu'il y a de nombreux éléments dans un tableau.	Ajout d'une barre de recherche et/ou de filtres pour faciliter la recherche de clients.
Livraisons	Utilisateur	Une erreur "Method Not Allowed" (405) se produit lors d'une requête POST en cas de refus de livraison.	/
Livraisons	Utilisateur / Administrateur	Les tableaux des livraisons passées présentent des problèmes de formatage : Administrateur : Certaines colonnes sont décalées, avec les noms apparaissant dans la colonne "volume" et d'autres informations mal positionnées.	/

Mes livraisons passées

Id	Volume	Poids	Prix	Statut
1	BERNADOU	12	12 123.0	Terminée
3	BERNADOU	78	67 19.0	Terminée
6	BERNADOU	57	99 1.0	Terminée

Utilisateur : Le poids et le prix sont affichés dans la même colonne, déplaçant le statut dans la colonne des prix, ce qui rend le tableau difficile à lire et confus.

Mes livraisons passées

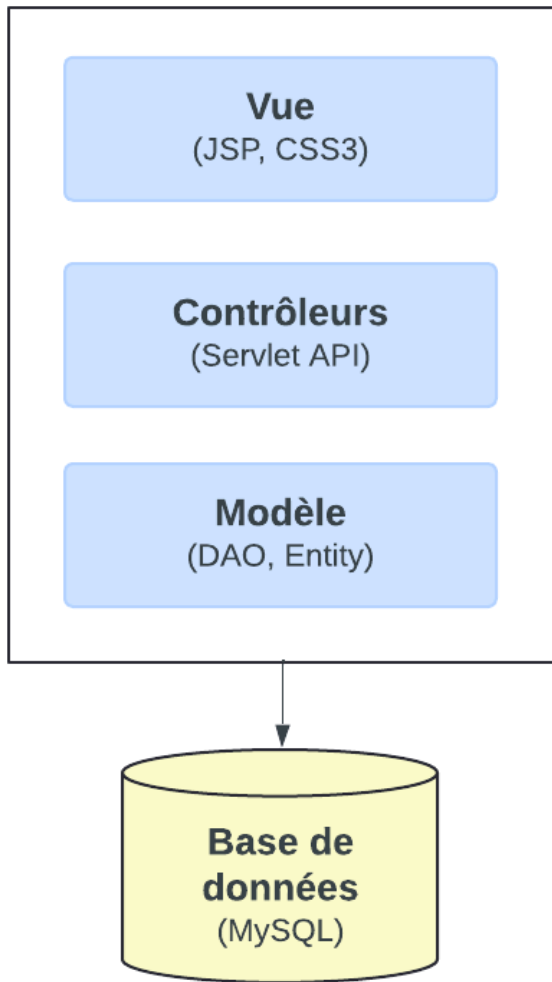
Id	Volume	Poids	Prix	Statut
1	12	12 123.0	Terminée	
3	78	67 19.0	Terminée	
6	57	99 1.0	Terminée	

Description Technique

Composant	Version
Architecture	Architecture en couches (MVC)
Langages	
Java	6

JSP (JavaServer Pages)	3.0
CSS	3
Frameworks	
Servlet API	3.0.1
MySQL	8.0.32
Plugins Maven	
maven-clean-plugin	3.1.0
maven-resources-plugin	3.0.2
maven-compiler-plugin	3.8.0
maven-surefire-plugin	2.22.1
maven-war-plugin	3.2.2
maven-install-plugin	2.5.2
maven-deploy-plugin	2.8.2

Schéma :



Points Forts et Déficiences

Points Forts :

- **Séparation des préoccupations :**

L'architecture en couches (Servlets, DAO, Beans, JSPs) est bien implémentée, ce qui permet une organisation claire du code. Chaque couche remplit une fonction spécifique, ce qui facilite la maintenance et l'évolution de l'application.

- **Sécurité des requêtes SQL :**

L'utilisation de requêtes préparées via `prepareStatement` permet d'éviter les attaques par injection SQL, augmentant ainsi la sécurité des transactions avec la base de données.

- **Gestion partielle de l'authentification :**

La présence d'un filtre pour gérer une partie de l'authentification montre une prise en compte de la sécurité. Cela peut être renforcé avec des mécanismes d'authentification plus robustes.

- **Performances optimales :**

Les temps de chargement sont très rapides, grâce à l'optimisation des requêtes et la légèreté des interactions entre le frontend et le backend.

Déficiences :

- **Manque de réactivité (Responsive) :**

L'application n'est pas responsive, ce qui la rend difficilement utilisable sur des appareils mobiles ou des écrans de tailles différentes. Un redesign avec des techniques de responsive design est nécessaire pour améliorer l'expérience utilisateur.

- **Exposition des données sensibles :**

Les informations sensibles comme les identifiants de la base de données, les drivers, et autres paramètres critiques sont visibles en clair, car il n'y a pas de fichier de configuration sécurisé (ex. : `application.properties` ou `.env`). Cela pose un risque majeur pour la sécurité de l'application.

- **Fermeture incorrecte des statements :**

Certains statements ne sont pas fermés correctement après leur exécution, ce qui peut entraîner des fuites de mémoire et des performances dégradées sur le long terme.

- **Connexion à la base de données non fermée :**

Il y a des oublis dans la fermeture des connexions à la base de données après utilisation. Cela peut saturer le pool de connexions et provoquer des problèmes de disponibilité à mesure que l'application prend de l'ampleur.

- **Absence de tests unitaires :**

L'application actuelle ne dispose d'aucune suite de tests unitaires pour vérifier le bon fonctionnement des différentes fonctionnalités. Sans tests pour vérifier le bon fonctionnement du code, il y a un risque accru d'introduire des régressions ou des bugs lors de l'ajout de nouvelles fonctionnalités ou de corrections.

- **Architecture monolithique :**

L'application actuelle est basée sur une architecture monolithique*, ce qui rend difficile la scalabilité et la maintenance à long terme. Toute modification ou mise à jour, même minime, nécessite de redéployer l'ensemble de l'application. Cela crée une rigidité dans les processus de développement et de déploiement, et empêche la mise en place de microservices qui pourraient offrir une plus grande flexibilité et une meilleure répartition des charges.

*Architecture monolithique : Regroupe toutes ses fonctionnalités dans un seul et même bloc de code.

Conclusion

L'audit de l'application actuelle de LiVrai met en lumière plusieurs points forts, mais également des déficiences importantes qui compromettent son évolutivité et sa sécurité. Bien que l'architecture en couches soit bien structurée et que les performances soient optimisées, plusieurs aspects nécessitent une refonte, notamment l'expérience utilisateur, la gestion des erreurs, et la sécurité des données. L'absence de réactivité et la mauvaise gestion des connexions et des statements peuvent engendrer des risques à long terme.

La modernisation de l'application, en intégrant des solutions plus robustes et adaptées à la croissance de l'entreprise, est indispensable pour maintenir la qualité du service. L'ajout de nouvelles fonctionnalités, une interface plus intuitive, et une meilleure gestion de la sécurité devraient permettre à LiVrai de répondre efficacement à ses besoins actuels et futurs.

Nouveau cahier des charges

Sommaire

1. Contexte du projet

- a. Besoins du client
- b. Objectifs du projet
- c. Mesure de la réussite du projet
- d. Contraintes réglementaires
- e. Éléments hors périmètre

2. Description fonctionnelle

- a. Liste des fonctionnalités
- b. Fonctionnement de l'application
- c. Impact Social

3. Description technique

Contexte du projet

Besoins du client :

LiVrai est une entreprise spécialisée dans la livraison de marchandises en grande quantité pour les professionnels. Actuellement, l'application de gestion des clients et des livraisons ne répond plus aux besoins croissants de l'entreprise et de ses clients en raison de l'expansion de son offre.

L'application actuelle centralise la gestion des comptes clients et des livraisons. Pour un client, le processus commence par l'intervention d'un administrateur, chargé de créer les comptes clients. Une fois le compte créé, les clients peuvent se connecter pour passer leurs commandes, consulter leurs livraisons à venir, ainsi que l'historique de leurs livraisons. Quant aux administrateurs, leurs comptes sont créés en interne dans la base de données. Une fois leurs comptes créés, les administrateurs peuvent se connecter pour ajouter des clients et consulter les livraisons à venir ainsi que l'historique des livraisons des clients.

Avec la croissance rapide de LiVrai, notamment l'augmentation du nombre de clients et des livraisons, l'application actuelle présente plusieurs limitations qui ne répondent plus aux exigences actuelles :

- **Dépendance au service commercial :**

Actuellement, chaque client doit passer par le service commercial pour la création d'un compte. Cette approche ralentit le processus et limite l'autonomie des clients, alors que nous souhaitons désormais que la majorité d'entre eux puissent gérer leurs livraisons de manière indépendante.

- **Absence de certaines fonctionnalités clés :**

La création de compte et la gestion de la facturation, par exemple, ne sont pas intégrés dans l'application actuelle.

- **Gestion limitée des utilisateurs :**

L'application ne différencie pas assez clairement les différents types d'utilisateurs (clients, service commercial, service Livraisons), ce qui complique la gestion des permissions et des fonctionnalités spécifiques à chaque rôle.

Objectifs du projet :

Certaines fonctionnalités sont incontournables pour cette refonte :

- **Création de compte**
- **Gestion des informations clients**
- **Commande et suivi des livraisons**
- **Gestion de la facturation**
- **Historique des livraisons**

L'application devra également gérer trois types d'utilisateurs :

- **Les clients**
- **Le service commercial**
- **Le service livraisons**

Sur le plan technique, l'application devra répondre aux exigences suivantes :

- **Frontend**
sera une application monopage (SPA) développée en **Angular**
- **Backend**
sera développé en **Java**, dans sa version LTS (Long Term Support)
- **Déploiement en production**
se fera dans un environnement **Docker**

Mesure de la réussite du projet :

- **Satisfaction des utilisateurs :**
Évaluer la satisfaction des utilisateurs via des enquêtes et des tests d'utilisabilité après le déploiement.
- **Performance de l'application :**
Mesurer les temps de réponse et la fluidité de l'application avant et après la refonte.
- **Réduction des bugs :**
Diminuer le nombre de rapports de bugs et d'erreurs post-déploiement.
- **Taux de réussite de livraisons :**
Mesurer l'efficacité de l'application en suivant le pourcentage de livraisons réalisées sans incidents ou retards. Un taux de réussite élevé démontrera la fiabilité du nouveau système.
- **Coûts opérationnels :**
Une réduction significative des coûts opérationnels reflètera l'efficacité de la refonte et la durabilité à long terme de l'application.

Contraintes réglementaires :

- **Protection des données personnelles :**

Garantir que les données des utilisateurs sont stockées de manière sécurisée et traitées conformément aux exigences du Règlement Général sur la Protection des Données (RGPD).

- **Droit d'accès et de rectification :**

Permettre aux utilisateurs de consulter, corriger, ou supprimer leurs données personnelles via une interface dédiée dans l'application.

- **Gestion des consentements :**

Intégrer des fonctionnalités pour obtenir, gérer et enregistrer le consentement des utilisateurs concernant le traitement de leurs données personnelles.

- **Notification de violation de données :**

Mettre en place des procédures pour détecter, signaler, et gérer toute violation de données dans les délais requis par le RGPD (72 heures). En cas de fuite ou de compromission d'informations personnelles, les utilisateurs doivent être immédiatement informés et des mesures correctives doivent être mises en œuvre.

- **Conditions générales et politiques de vente (CGV) :**

Rédiger et intégrer les Conditions Générales de Vente dans l'application. Ces CGV doivent encadrer les relations commerciales entre LiVrai et ses clients, précisant les obligations de chaque partie.

- **Conditions générales d'Utilisation (CGU) :**

Rédiger et afficher clairement les Conditions Générales d'Utilisation dans l'application. Celles-ci doivent préciser les droits et obligations des utilisateurs lors de l'utilisation de l'application, notamment en ce qui concerne la propriété intellectuelle, les responsabilités, et l'usage des services proposés.

- **Mentions légales :**

Inclure les mentions légales obligatoires dans l'application, telles que l'identification de l'éditeur, l'hébergeur, les coordonnées de l'entreprise, ainsi que les informations liées à la protection des données. Ces informations doivent être facilement accessibles et conformes aux exigences légales en vigueur.

Éléments hors périmètre :

- **Intégration d'un système de paiement (exemple : Stripe)**
Permet aux clients de régler directement leurs factures en ligne via l'application.
 - **Notifications en temps réel**
Envoi de notifications (SMS, email, push) aux clients et aux administrateurs pour les mises à jour des livraisons (ex : confirmation, retards, modifications).
 - **Suivi des livraisons en temps réel via GPS**
Permet aux clients de voir l'emplacement en temps réel de leurs marchandises en cours de livraison.
 - **Gestion des retours ou des réclamations**
Implémenter une fonctionnalité pour gérer les retours de marchandise ou soumettre des réclamations directement depuis l'application.
 - **Support client intégré (exemple : chat en ligne)**
Ajouter une option de chat en temps réel pour les clients afin de poser des questions ou obtenir de l'assistance technique.
-

Description fonctionnelle

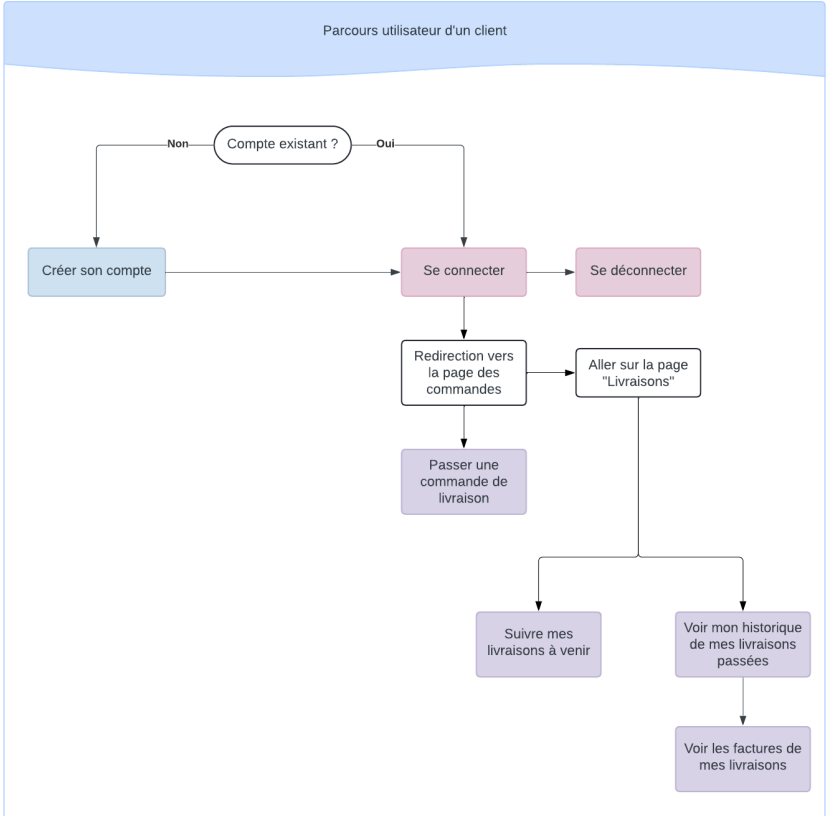
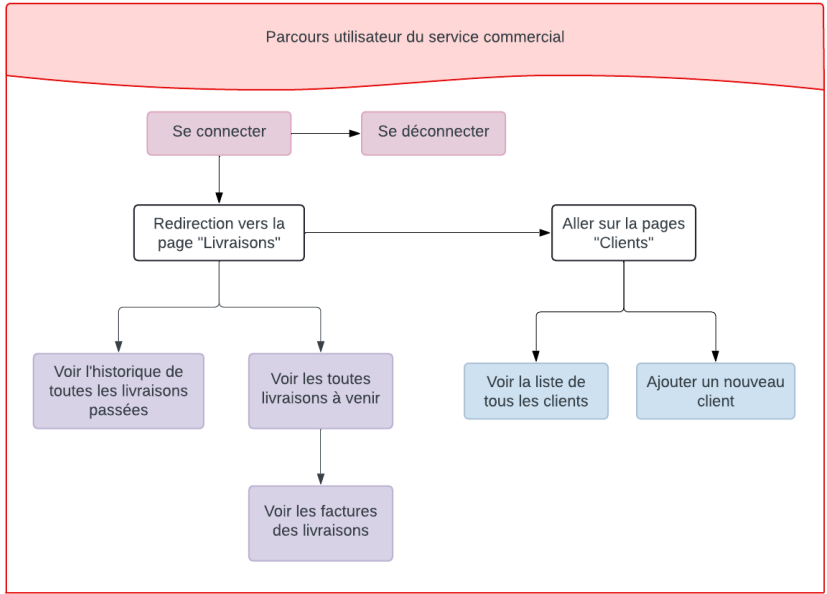
Liste des fonctionnalités :

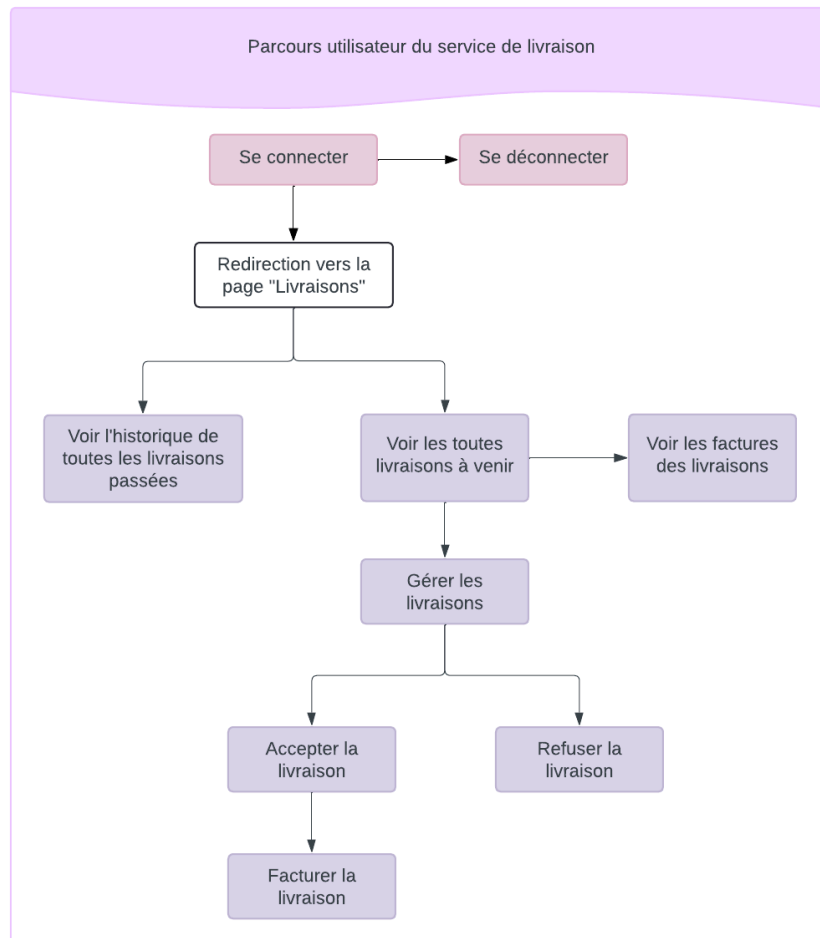
/	Type d'utilisateur	Fonctionnalités	Besoin client
Général	Tous	Se connecter	Accéder à l'application et aux fonctionnalités propres à chaque rôle
Général	Tous	Se déconnecter	Sécuriser l'accès en fermant la session

Gestion des comptes clients	Service commercial / Client	Création de compte client	Permettre l'accès au service en autonomie pour les clients ou via le service commercial
Gestion des comptes clients	Service commercial	Voir la liste de tous les clients	Permet au service commercial de gérer et de suivre l'ensemble des comptes clients dans l'application.
Gestion des comptes clients	Client	Modification du compte client	Permettre aux clients de mettre à jour leurs informations personnelles (RGPD)
Gestion des comptes clients	Client	Suppression du compte client	Offrir aux clients la possibilité de supprimer leur compte si besoin (RGPD)
Gestion des livraisons	Service commercial / Service de livraison	Suivre toutes les livraisons à venir	Facilite la gestion et le suivi des livraisons pour le service commercial et de livraison.
Gestion des livraisons	Service de livraison	Accepter ou refuser des livraisons	Donne au service de livraison le contrôle sur les commandes.
Gestion des livraisons	Service de livraison	Facturer les livraisons	Permet au service de livraison de facturer les clients après chaque livraison effectuée.
Gestion des livraisons	Service commercial	Voir tout l'historique des livraisons passées des clients	Offrir un historique pour le suivi client
Gestion des livraisons	Service commercial / Service de livraison	Voir les factures de toutes les livraisons passées	Gérer la facturation et les archives comptables

Gestion des livraisons	Client	Passer une commande de livraison	Offrir un service autonome pour commander des livraisons
Gestion des livraisons	Client	Suivre ses livraisons à venir	Offrir un suivi en temps réel des commandes en cours
Gestion des livraisons	Client	Voir son historique de ses livraisons passées	Fournir un accès à l'historique des livraisons
Gestion des livraisons	Client	Voir les factures de ses livraisons passées	Permettre aux clients de suivre leurs factures de leurs livraisons

Fonctionnement de l'application :





Impact Social :

Impact environnemental :

- **Optimisation des ressources :**

La refonte vise à améliorer l'efficacité de l'application, réduisant ainsi les besoins en ressources serveur et, par conséquent, l'empreinte carbone associée.

Accessibilité :

- **Utilisabilité pour les personnes handicapées :**

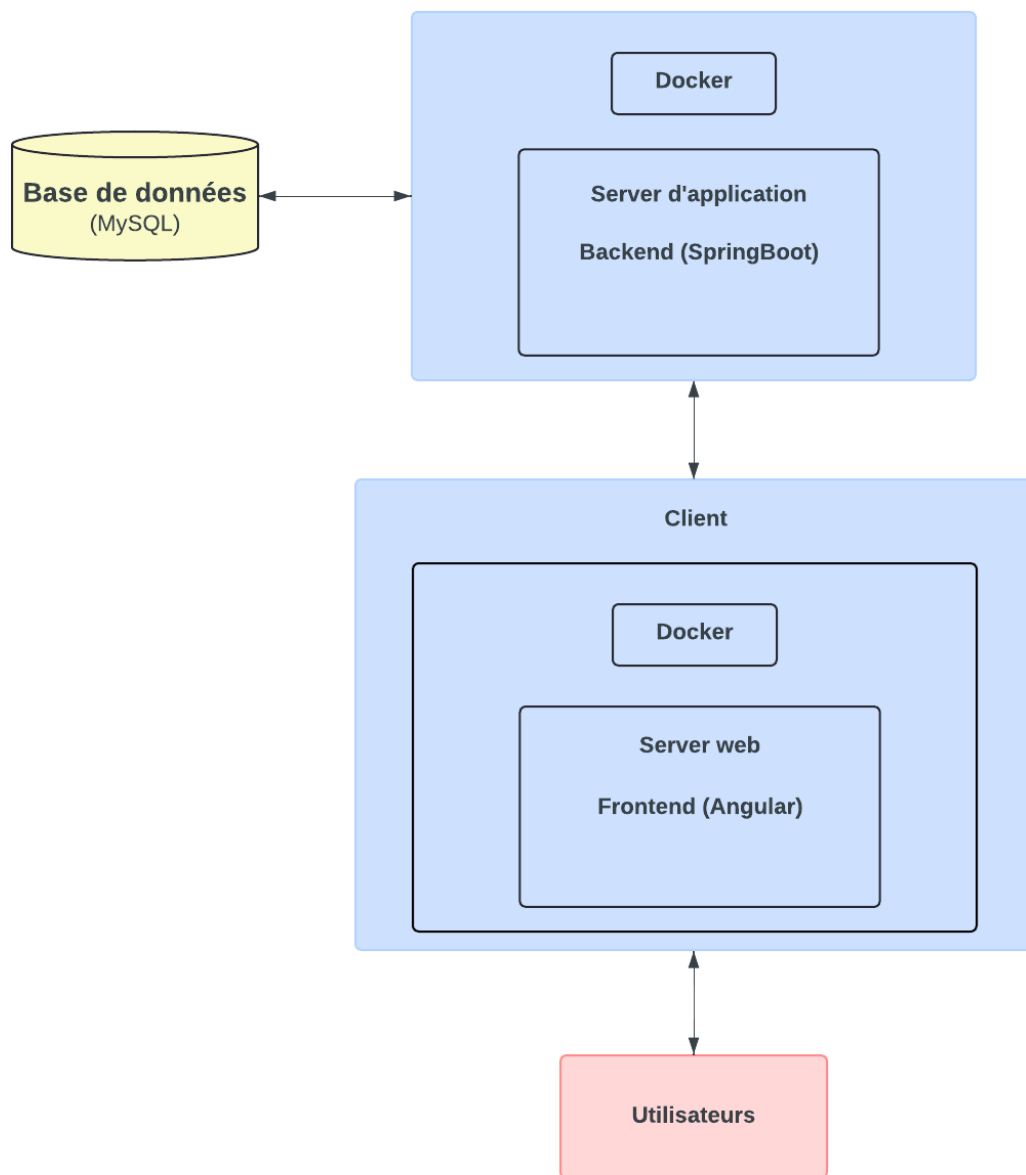
L'application doit être conçue en tenant compte des normes d'accessibilité web (WCAG) pour garantir que les personnes en situation de handicap puissent utiliser toutes les fonctionnalités de l'application.

Description technique

Composant	Version
Architecture	Architecture Client-Server
Languages	
Java	17 (LTS)
Angular	16.1.0
Frameworks	
Spring Boot	3.1.2
Spring Security	6.1.0
Spring Data JPA	3.1.2
Hibernate ORM	6.2.7.Final
Lombok	1.18.30

Maven	3.8.7
MySQL	8.1.0
RxJS	7.8.0
Plugins Maven	
maven-clean-plugin	3.1.0
maven-resources-plugin	3.0.2
maven-compiler-plugin	3.11.0
maven-surefire-plugin	3.0.0-M7
maven-failsafe-plugin	3.0.0-M7
jacoco-maven-plugin (coverage)	0.8.10
Sécurité	
OAuth2 / JWT	6.1.0

Tests	
JUnit5	5.10.0
Spring Boot Test	3.1.2
Mockito	4.11.0
Cypress	13.3.0
Jest	29.6.4
Déploiement	
Docker	24.0.2



LiVrai

LiVrai est une entreprise spécialisée dans la livraison de marchandises en grande quantité pour les professionnels. Leur application ne répond plus aux besoins en expansion de l'entreprise et de sa clientèle.

Un audit approfondi a été réalisé pour évaluer les forces et les faiblesses de l'application existante, ainsi que pour identifier les domaines nécessitant des améliorations significatives.

Et un nouveau cahier des charges a été également réalisé afin de définir les objectifs, les fonctionnalités et les spécifications techniques pour une refonte complète de l'application, afin de répondre aux exigences actuelles et futures de LiVrai.

Partie Audit

Contexte

LiVrai est confrontée à des défis croissants dus à l'expansion de son offre. L'application actuelle de gestion des clients et des livraisons ne répond plus aux besoins en évolution de l'entreprise. Cette situation nécessite une évaluation approfondie et une modernisation de l'application pour assurer la continuité et l'amélioration des suivis des services de livraison.

Expansion de l'offre

LiVrai élargit ses services, nécessitant une application plus robuste et flexible.

Besoins croissants

Besoin de fonctionnalités plus avancées et une meilleure expérience utilisateur.

Nécessité de modernisation

L'application actuelle atteint ses limites et requiert une refonte pour s'adapter aux nouvelles exigences du marché.

Fonctionnalités

L'application actuelle offre des fonctionnalités distinctes pour les clients et les administrateurs. Les fonctionnalités générales incluent la connexion et la déconnexion pour tous les utilisateurs. Les clients peuvent commander de nouvelles livraisons et consulter leurs livraisons à venir et passées. Les administrateurs ont des capacités supplémentaires telles que l'ajout de nouveaux clients, la gestion des livraisons (acceptation, rejet, facturation) et la visualisation des livraisons à venir et passées pour tous les clients.

Fonctionnalités Clients

- Commander une nouvelle livraison
- Afficher les livraisons à venir
- Afficher les livraisons passées

Fonctionnalités Administrateurs

- Ajouter un nouveau client
- Gérer les livraisons
- Afficher les livraisons à venir de tous les clients
- Afficher les livraisons passées de tous les clients

Fonctionnalités Générales

- Se connecter
- Se déconnecter

Expérience Utilisateur

L'expérience utilisateur de l'application actuelle présente plusieurs problèmes. L'interface manque d'attrait visuel et la présentation des erreurs est intrusive. La page d'accueil est peu informative, et la navigation dans les tableaux de données est difficile. Des problèmes de formatage affectent la lisibilité des tableaux de livraisons, en particulier pour les administrateurs.

1

Interface peu attrayante

L'interface simpliste décourage les utilisateurs et manque d'attrait visuel.

2

Gestion des erreurs

Les erreurs sont présentées de manière intrusive, interrompant la navigation.

3

Navigation difficile

Les tableaux longs nécessitent un défilement excessif et manquent de fonctionnalités de recherche.

4

Problèmes de formatage

Les tableaux de livraisons présentent des décalages et des confusions dans l'affichage des données.

Problèmes d’Affichage des Livraisons

Les tableaux des livraisons passées présentent des problèmes de formatage spécifiques pour les administrateurs et les utilisateurs. Pour les administrateurs, certaines colonnes sont décalées, avec les noms apparaissant dans la colonne "volume" et d'autres informations mal positionnées. Pour les utilisateurs, le poids et le prix sont affichés dans la même colonne, déplaçant le statut dans la colonne des prix, ce qui rend le tableau difficile à lire et confus.

Mes livraisons passées

Id	Volume	Poids	Prix	Statut
1	BERNADOU	12	12 123.0	Terminée
3	BERNADOU	78	67 19.0	Terminée
6	BERNADOU	57	99 1.0	Terminée

Vue Administrateur

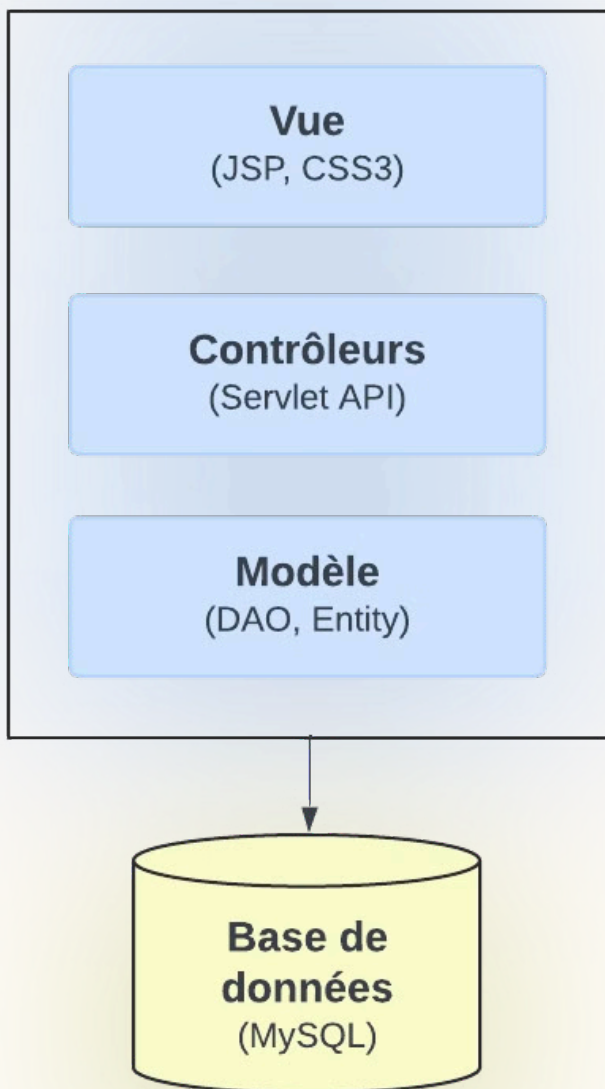
Colonnes décalées, noms dans la colonne "volume"

Mes livraisons passées

Id	Volume	Poids	Prix	Statut
1	12	12 123.0	Terminée	
3	78	67 19.0	Terminée	
6	57	99 1.0	Terminée	

Vue Utilisateur

Poids et prix dans la même colonne, statut déplacé



Description Technique

L'application utilise une architecture en couches (MVC) principalement avec Java 6, JSP 3.0 et CSS3. Elle s'appuie sur le framework Servlet API 3.0.1 et MySQL 8.0.32 pour la gestion de la base de données. Plusieurs plugins Maven sont utilisés pour la gestion du projet, notamment pour le nettoyage, la compilation, les tests et le déploiement.

Composant	Version
Java	6
JSP	3.0
CSS	3
Servlet API	3.0.1
MySQL	8.0.32

Points Forts et Déficiences

L'application présente des points forts tels qu'une bonne séparation des préoccupations, une sécurité des requêtes SQL, une gestion partielle de l'authentification et des performances optimales. Cependant, elle souffre de déficiences importantes, notamment un manque de réactivité, l'exposition de données sensibles, une gestion incorrecte des ressources de base de données, l'absence de tests unitaires et une architecture monolithique limitant la scalabilité.



Architecture en couches

Bonne séparation des préoccupations facilitant la maintenance.



Sécurité SQL

Utilisation de requêtes préparées pour prévenir les injections SQL.



Non responsive

L'application n'est pas adaptée aux appareils mobiles.



Absence de tests

Aucune suite de tests unitaires pour vérifier les fonctionnalités.

Conclusion

L'audit de l'application LiVrai révèle des points forts en termes d'architecture et de performances, mais souligne également des déficiences critiques en matière d'expérience utilisateur, de sécurité et d'évolutivité. La modernisation de l'application est indispensable pour répondre aux besoins croissants de l'entreprise. Cela implique une refonte de l'interface utilisateur, une amélioration de la gestion des erreurs, un renforcement de la sécurité des données et une transition vers une architecture plus flexible. Ces améliorations permettront à LiVrai de maintenir un service de qualité et de s'adapter efficacement à sa croissance future.

Partie Cahier des Charges

Contexte du projet

1 Dépendance au service commercial

Chaque client doit passer par le service commercial pour la création d'un compte, ralentissant le processus et limitant l'autonomie des clients.

2 Absence de fonctionnalités clés

La création de compte et la gestion de la facturation ne sont pas intégrées dans l'application actuelle.

3 Gestion limitée des utilisateurs

L'application ne différencie pas assez clairement les différents types d'utilisateurs, compliquant la gestion des permissions et des fonctionnalités spécifiques à chaque rôle.

Objectifs du projet

Fonctionnalités incontournables

- Création de compte
- Gestion des informations clients
- Commande et suivi des livraisons
- Gestion de la facturation
- Historique des livraisons

Types d'utilisateurs

- Les clients
- Le service commercial
- Le service livraisons

Exigences techniques

- Frontend: Angular (SPA)
- Backend: Java (version LTS)
- Déploiement: Docker

Contraintes réglementaires

1

Protection des données personnelles

Garantir le stockage sécurisé et le traitement conforme au RGPD des données utilisateurs.

2

Droit d'accès et de rectification

Permettre aux utilisateurs de consulter, corriger, ou supprimer leurs données personnelles via une interface dédiée.

3

Gestion des consentements

Intégrer des fonctionnalités pour obtenir, gérer et enregistrer le consentement des utilisateurs.

4

Notification de violation de données

Mettre en place des procédures pour détecter, signaler, et gérer toute violation de données dans les délais requis par le RGPD.

Éléments hors périmètre

Intégration d'un système de paiement

Permet aux clients de régler directement leurs factures en ligne via l'application.

Notifications en temps réel

Envoi de notifications (SMS, email, push) aux clients et aux administrateurs pour les mises à jour des livraisons.

Suivi des livraisons en temps réel via GPS

Permet aux clients de voir l'emplacement en temps réel de leurs marchandises en cours de livraison.

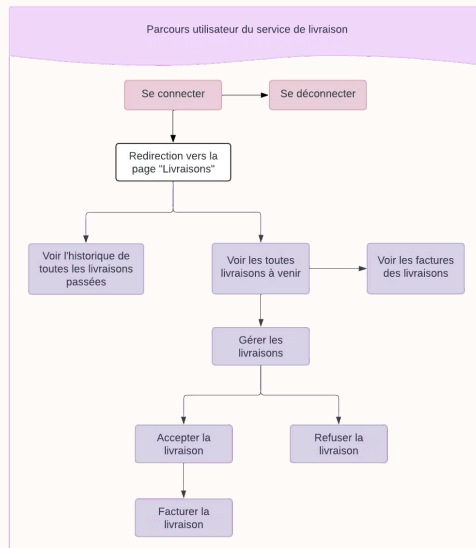
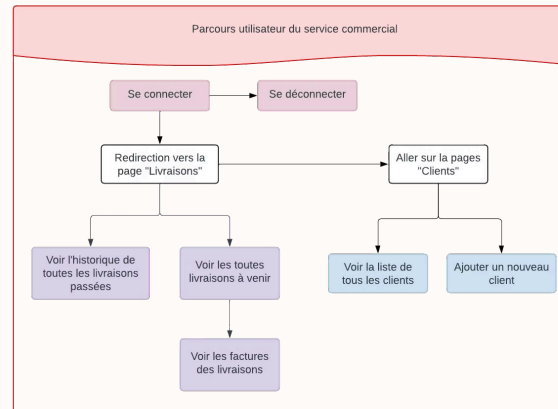
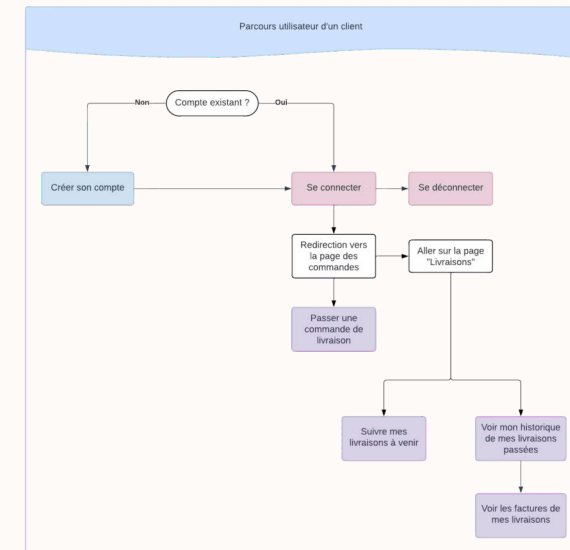
Gestion des retours ou des réclamations

Implémenter une fonctionnalité pour gérer les retours de marchandise ou soumettre des réclamations directement depuis l'application.

Description fonctionnelle

/	Type d'utilisateur	Fonctionnalités	Besoin client
Général	Tous	Se connecter	Accéder à l'application et aux fonctionnalités propres à chaque rôle
Général	Tous	Se déconnecter	Sécuriser l'accès en fermant la session
Gestion des comptes clients	Service commercial / Client	Création de compte client	Permettre l'accès au service en autonomie pour les clients ou via le service commercial
Gestion des comptes clients	Service commercial	Voir la liste de tous les clients	Permet au service commercial de gérer et de suivre l'ensemble des comptes clients dans l'application.
Gestion des comptes clients	Client	Modification du compte client	Permettre aux clients de mettre à jour leurs informations personnelles (RGPD)
Gestion des comptes clients	Client	Suppression du compte client	Offrir aux clients la possibilité de supprimer leur compte si besoin (RGPD)
Gestion des livraisons	Service commercial / Service de livraison	Suivre toutes les livraisons à venir	Facilite la gestion et le suivi des livraisons pour le service commercial et de livraison.
Gestion des livraisons	Service de livraison	Accepter ou refuser des livraisons	Donne au service de livraison le contrôle sur les commandes.
Gestion des livraisons	Service de livraison	Facturer les livraisons	Permet au service de livraison de facturer les clients après chaque livraison effectuée.
Gestion des livraisons	Service commercial	Voir tout l'historique des livraisons passées des clients	Offrir un historique pour le suivi client
Gestion des livraisons	Service commercial / Service de livraison	Voir les factures de toutes les livraisons passées	Gérer la facturation et les archives comptables
Gestion des livraisons	Client	Passer une commande de livraison	Offrir un service autonome pour commander des livraisons
Gestion des livraisons	Client	Suivre ses livraisons à venir	Offrir un suivi en temps réel des commandes en cours
Gestion des livraisons	Client	Voir son historique de ses livraisons passées	Fournir un accès à l'historique des livraisons
Gestion des livraisons	Client	Voir les factures de ses livraisons passées	Permettre aux clients de suivre leurs factures de leurs livraisons

Fonctionnement de l'application



Impact Social

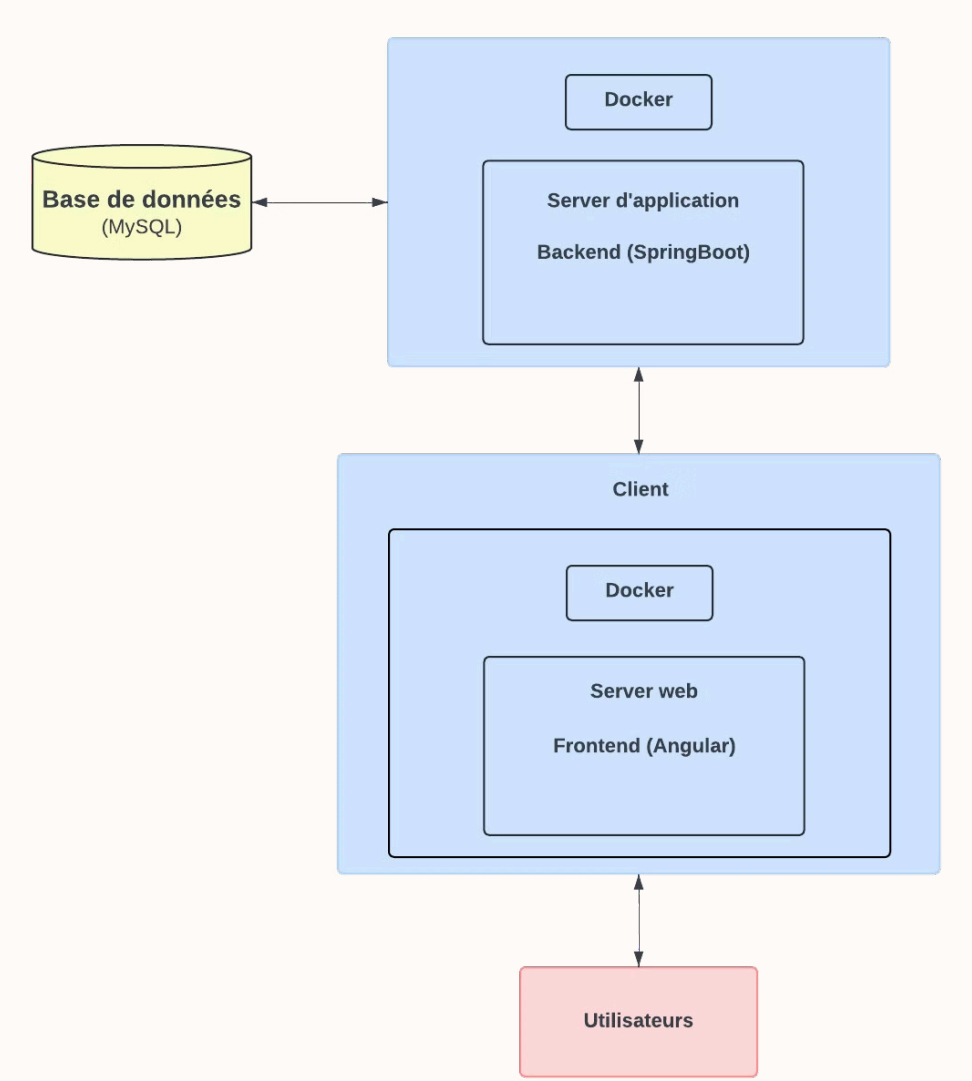
Impact environnemental

La refonte vise à améliorer l'efficacité de l'application, réduisant ainsi les besoins en ressources serveur et, par conséquent, l'empreinte carbone associée. Cette optimisation des ressources contribue à une utilisation plus durable des technologies de l'information.

Accessibilité

L'application doit être conçue en tenant compte des normes d'accessibilité web (WCAG) pour garantir que les personnes en situation de handicap puissent utiliser toutes les fonctionnalités de l'application. Cela inclut des considérations pour la navigation au clavier, le contraste des couleurs, et la compatibilité avec les lecteurs d'écran.

Description technique



Composant	Version
Architecture	Architecture Client-Server
Languages	
Java	17 (LTS)
Angular	16.1.0
Frameworks	
Spring Boot	3.1.2
Spring Security	6.1.0
Spring Data JPA	3.1.2
Hibernate ORM	6.2.7.Final
Lombok	1.18.30
Maven	3.8.7
MySQL	8.1.0
RxJS	7.8.0
Plugins Maven	
maven-clean-plugin	3.1.0
maven-resources-plugin	3.0.2
maven-compiler-plugin	3.11.0
maven-surefire-plugin	3.0.0-M7
maven-failsafe-plugin	3.0.0-M7
jacoco-maven-plugin (coverage)	0.8.10
Sécurité	
OAuth2 / JWT	6.1.0
Tests	
JUnit5	5.10.0
Spring Boot Test	3.1.2
Mockito	4.11.0
Cypress	13.3.0
Jest	29.6.4
Déploiement	
Docker	24.0.2