



Cahier des charges

1. Contexte du projet

a. Besoins du client

Livrai est une société de livraison de marchandises en gros volume pour professionnels à l'échelle nationale. Suite à un succès important et rapide, elle doit mettre à jour son application de gestion de clients et de livraison pour répondre aux nouvelles exigences de ses services.

Actuellement, la gestion des clients est réalisée par son service commercial qui doit réaliser pour les clients leur enregistrement dans l'application. Cependant, avec l'augmentation du nombre de clients, il faut que la nouvelle application puisse assurer une gestion automatisée des nouveaux clients. Cette évolution doit toutefois toujours permettre au service commercial d'enregistrer de nouveaux clients, comme sur l'ancien modèle, pour continuer à assurer ce service pour les plus gros utilisateurs.

b. Objectifs du projet

Le projet doit pouvoir intégrer la création de compte (par le client ou le service commercial), la gestion des informations des comptes clients, assurer la création et le suivi de commandes de livraison, gérer la facturation et conserver un historique des livraisons.

L'application doit pouvoir permettre l'existence de trois types d'utilisateurs aux rôles définis : les clients (accès à toutes les fonctionnalités associées à leur compte), le service commercial de Livrai (qui a accès aux-dites fonctionnalités pour les différents comptes clients) et le service Livraison (qui doit pouvoir avoir accès aux services de livraison et de facturation)

L'application doit comporter un backend et un frontend séparés dans des dossiers dédiés, mais intégrés au même repo Git. Le frontend de l'application doit être une SPA Angular, le backend développé en Java avec la dernière version LTS. Le déploiement en production doit de son côté s'effectuer via Docker.

c. Mesure de la réussite du projet

Chacune des fonctionnalités précédemment citée doit être testée et ne doit pas provoquer d'erreur. Les utilisateurs ne doivent pas pouvoir effectuer des actions en lien avec d'autres clients ou niveau de privilège (comme accéder à une page interdite par modification de l'url).

L'application doit être fonctionnelle et ne pas montrer de bug ou d'erreur dans les consoles lors des vérifications.

d. Contraintes réglementaires

Respect des normes RGPD :

- Limitation des données recueillies pour la création du compte aux informations nécessaires ;
- Données des utilisateurs supprimées lors de la fermeture d'un compte ;
- Cryptage des données sensibles : aucune donnée sensible ne doit être stockée en dur dans les serveurs ou le code ;
- Limitation de l'accès aux données aux seuls comptes qui en ont les droits ;
- Information des utilisateurs du cadre de l'utilisation des données via un texte informatif ;

e. Éléments hors périmètre

En l'état, le projet ne prévoit pas :

- La modification des comptes ou des commandes ;
- Une page dédiée aux informations personnelles pour les comptes clients ;
- Gestion de la base de données liée à l'application ;
- Possibilité de régler directement dans l'application ;
- La sélection du lieu de livraison.

Ces éléments pourraient faire l'objet d'évolutions ultérieures de l'application.

2. Description fonctionnelle

a. Liste des fonctionnalités

Fonctionnalités générales :

- Possibilité de se connecter via une page dédiée et de se déconnecter ;
- Possibilité de créer un nouveau compte client via la page d'accueil ;
- Navigation entre les différentes pages de l'application via une barre de navigation ;
- Existence de 3 niveaux de rôle : clients, livraison et admin (service commercial) ;

Fonctionnalités relatives à la gestion des comptes utilisateurs :

- Existence d'une page réservée aux admins (service commercial) avec les fonctionnalités relatives à la gestion des comptes utilisateurs ;
- Possibilité de créer un nouveau compte depuis la page de gestion des utilisateurs ;
- Visuel sur l'ensemble des comptes existants (admin) ;

Fonctionnalités relatives à la gestion des commandes :

- Page dédié à la réalisation de nouvelle commande pour les comptes clients ;
- Possibilité pour les comptes clients de poster une nouvelle commande en choisissant le volume et la quantité ;

Fonctionnalités relatives au suivi et à l'historique des livraisons :

- Page dédiée au suivi des livraisons accessible à tous les comptes ;
- Possibilité de suivre l'ensemble des livraisons en cours ou passées (compte livraison et

admin) ;

- Possibilité de voir les livraisons en cours ou passées effectuées par son compte (client) ;
- Possibilité d'accepter ou refuser une livraison (livraison, admin) ;

Fonctionnalités relatives à la facturation :

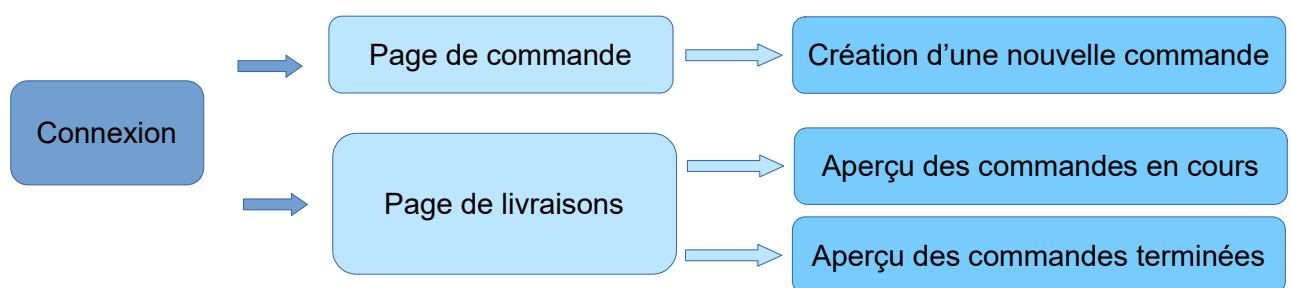
- Possibilité d'établir la facturation lors de l'acceptation d'une commande (livraison, admin) ;
- Possibilité de voir la facturation d'une commande à laquelle le compte a accès une fois celle-ci réalisée.

b. Fonctionnement de l'application

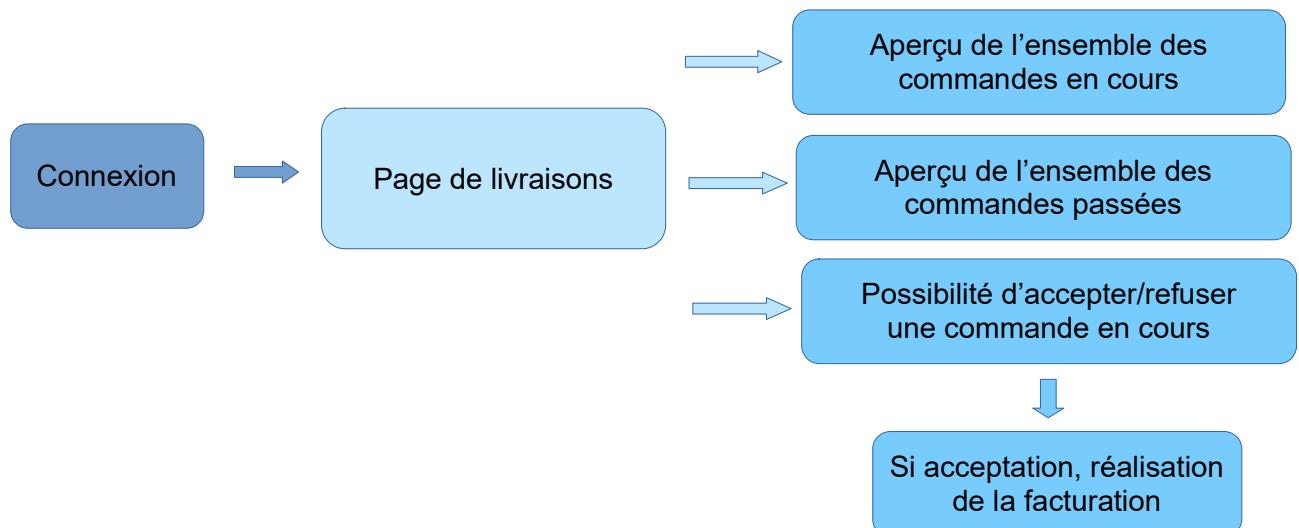
Arrivée sur la page d'accueil de l'application :



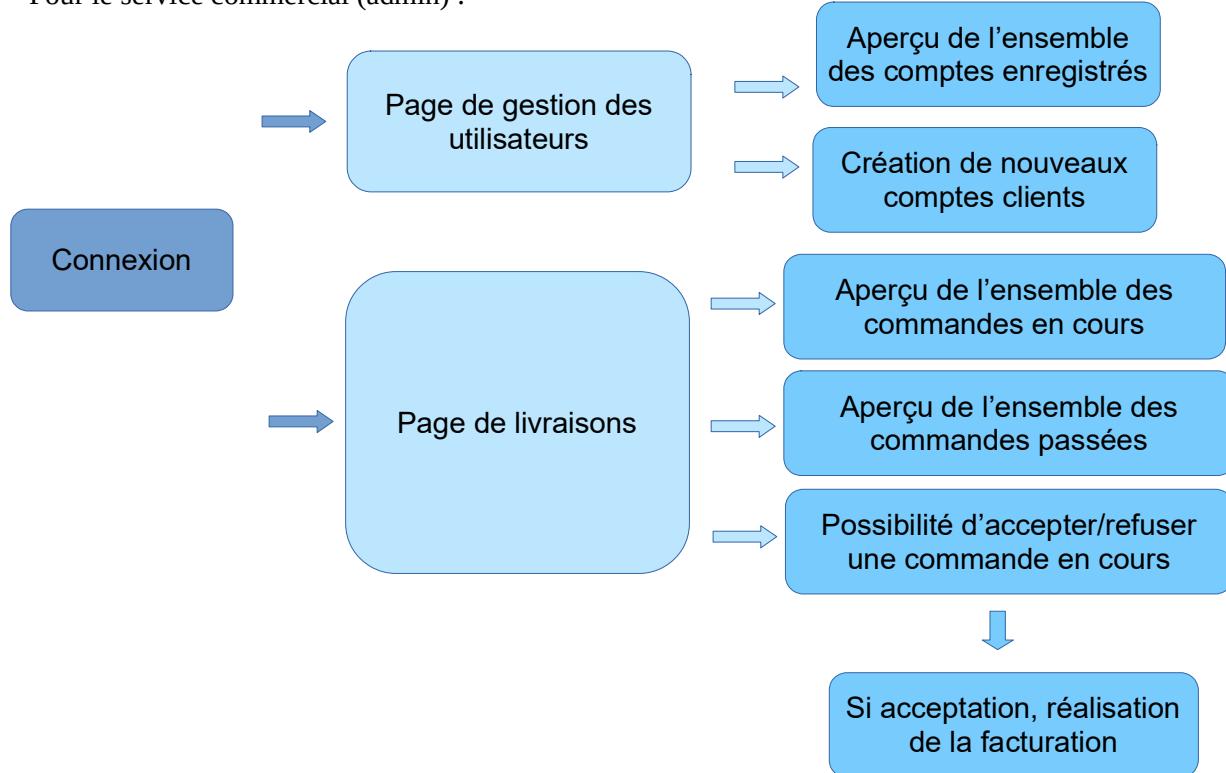
Pour les comptes clients :



Pour le service livraison :



Pour le service commercial (admin) :



c. Impact Social

L’application doit pouvoir être accessible au plus grand nombre, respecter les bonnes pratiques en terme de mise en page et de style (choix des balisages et des couleurs adaptées aux lecteurs d’écrans et personnes daltonniennes).

La nouvelle application doit également intégrer les dernières évolutions en terme de bonnes pratiques et de technologies : séparation du frontend et du backend, choix d’une version LTS de Java pour la backend et utilisation du framework Angular pour le frontend. Cela permet d’éviter les erreurs de compatibilité entre les JDK et IDE utilisés par les équipes de développement et l’application, et éviter l’apparition de bugs liés à l’ancienneté des outils utilisés. Les différentes erreurs et bugs de l’application doivent en outre être anticipée pour que les utilisateurs puissent savoir comment réagir face à ces dernières.

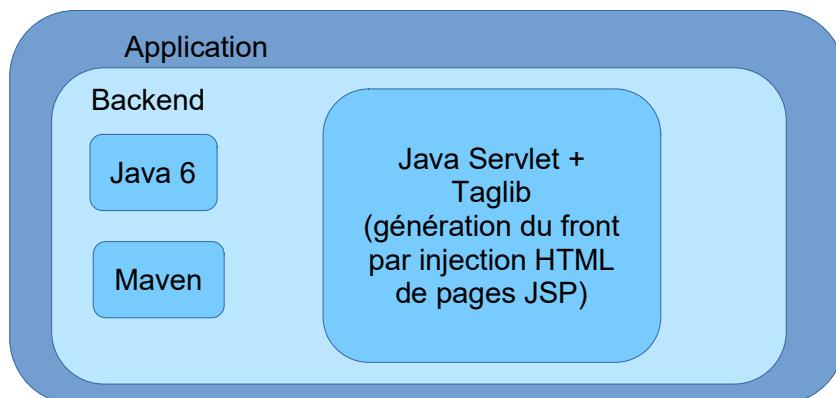
L’application doit également intégrer des mesures de sécurité afin de protéger les données sensibles des utilisateurs. Le choix de nouveaux frameworks, tel que Spring Boot et sa librairie Spring Security, permet de prendre en compte ce besoin et de l’intégrer avec les nouvelles fonctionnalités.

3. Description technique

La nouvelle application doit disposer d'un frontend et d'un backend séparé et reliés par une API, ce qui présente une différence d'architecture par rapport à l'ancienne version de l'application. Cela se traduit par l'adoption du framework Angular pour la réalisation du frontend de l'application, qui apporte robustesse et efficacité pour la génération d'une SPA. L'usage d'Angular permet par ailleurs de profiter de la librairie Angular Material pour la génération de composant matériel.

Le backend évolue également pour profiter des évolutions du langage Java, en passant de Java 6 à Java 21. L'application va également profiter de l'utilisation du framework Java SpringBoot, afin de gagner en temps de production et de la compatibilité de nombreuses librairies. Spring Security, la librairie de sécurité des frameworks Spring, va par ailleurs être intégrée au projet pour assurer la sécurité des requêtes serveurs.

Ancienne architecture :



Nouvelle architecture :

