



Cahier des charges

1. Contexte du projet

a. Besoins du client

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.

Les clients doivent passer par le service commercial pour établir un contrat de collaboration, ce service se charge de créer un compte et de l'attribuer au client pour réserver des livraisons.

b. Objectifs du projet

Compte tenu des limites de la solution existante, à savoir une forte croissance de notre activité, nous souhaitons que nos clients puissent être autonomes pour utiliser le service de livraison. Nous allons refaire entièrement l'application.

c. Mesure de la réussite du projet

Afin de mesurer la réussite du projet, nous voulons livrer une application qui, non seulement, répond aux nouveaux besoins des clients, mais qui présente des performances supérieures à la version précédente pour s'adapter aux flux croissants de nos utilisateurs.

d. Contraintes réglementaires

L'application devra se plier aux 7 règles générales de protection des données (RGPD):

1. Légalité, équité et transparence
2. Limitation du but
3. Minimisation des données
4. Précision
5. Limites de stockage
6. Intégrité et confidentialité
7. Responsabilité

Plus d'informations : <https://www.cyberpilot.io/cyberpilot-blog/data-protection-principles-the-7-principles-of-gdpr-explained/>

e. Éléments hors périmètre

Pour l'instant, rien de particulier à signaler, nous allons nous contenter de ce qui est indiqué précédemment.

2. Description fonctionnelle

a. Liste des fonctionnalités

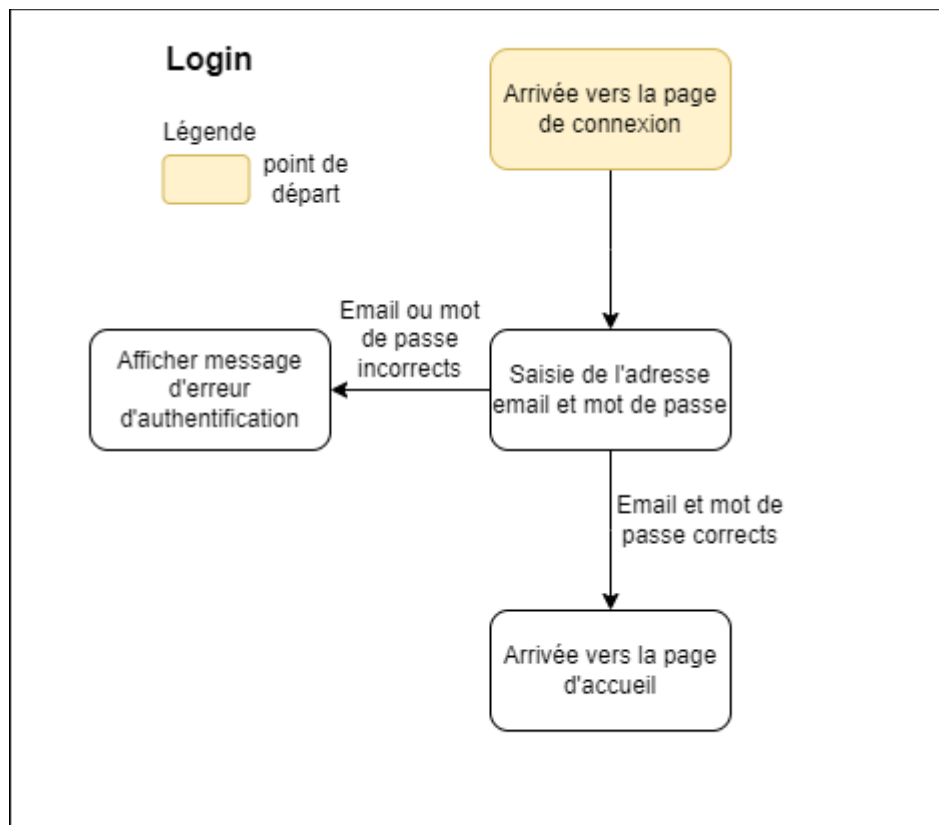
D'abord, nous devons distinguer trois types d'utilisateurs: les clients, le service commercial et le service livraisons.

Concernant les fonctionnalités, l'application devra au minimum:

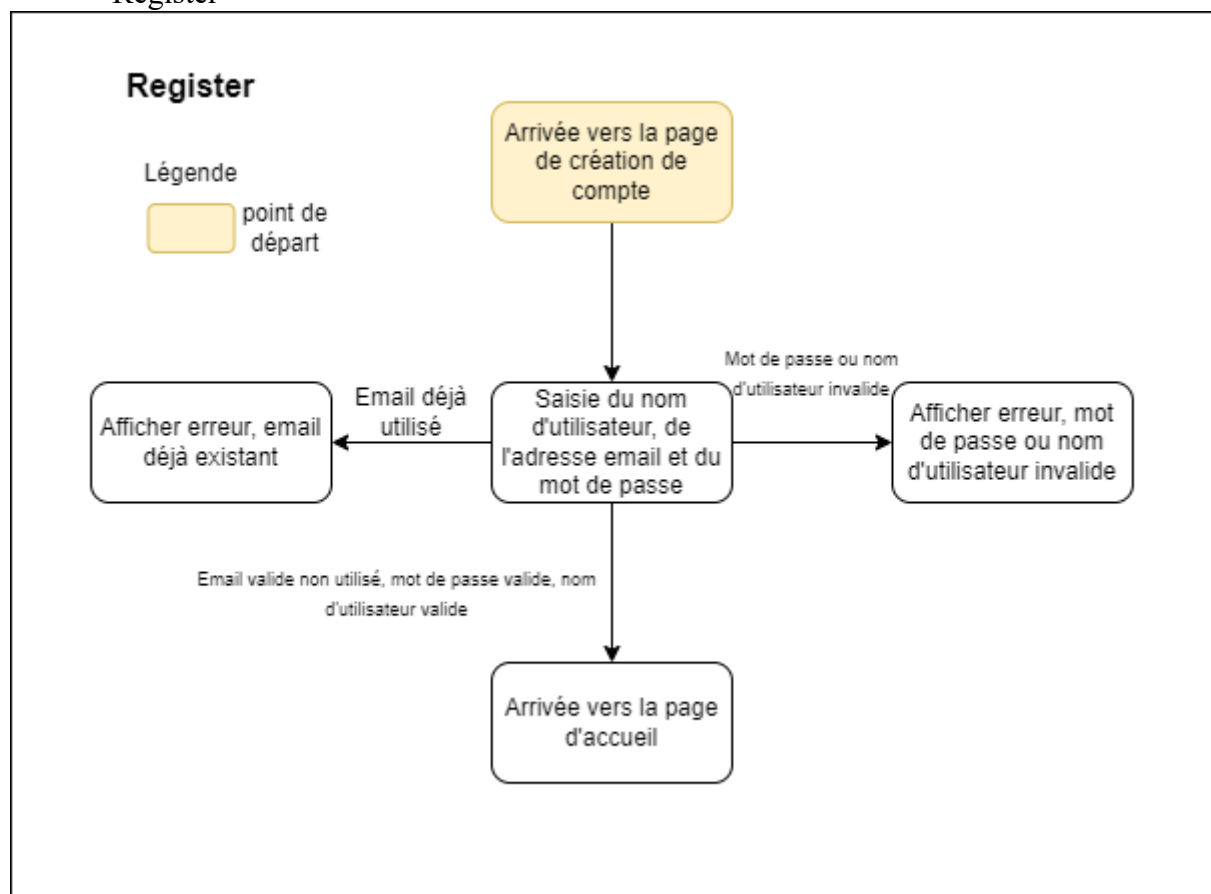
- Permettre la création de compte par les clients ou par notre service commercial: un nom d'utilisateur (au moins 3 caractères), une adresse email valide, un mot de passe (au moins 8 caractères dont un chiffre, une lettre minuscule, une lettre majuscule et un caractère spécial), une adresse postale facultative.
- Authentification de l'utilisateur: saisir son adresse email et son mot de passe.
- Gérer les informations du client par le service commercial: modifier le mot de passe, le nom, l'adresse email ou l'adresse postale du client.
- Gérer les informations du client de manière autonome: modifier son mot de passe, son nom ou son adresse email.
- Commander une livraison: le client devra saisir le volume (en L), le poids (en kg) et l'adresse postale requise de la livraison.
- Appliquer des tarifs pour la facturation: le service commercial devra définir les tarifs du volume au litre et du poids au kilogramme afin d'automatiser les prix des livraisons.
- Suivre une livraison en cours: le service livraison pourra suivre une livraison en cours, il pourra l'accepter ou le refuser.
- Gérer la facturation: le service livraisons peut accéder à la facturation des livraisons.
- Consulter l'historique des livraisons passées: le service livraisons peut voir la liste des livraisons passées avec les attributs suivants: nom d'utilisateur, volume, poids, prix, statut (Terminée ou Refusée).
- Déconnexion de l'utilisateur.

b. Fonctionnement de l'application

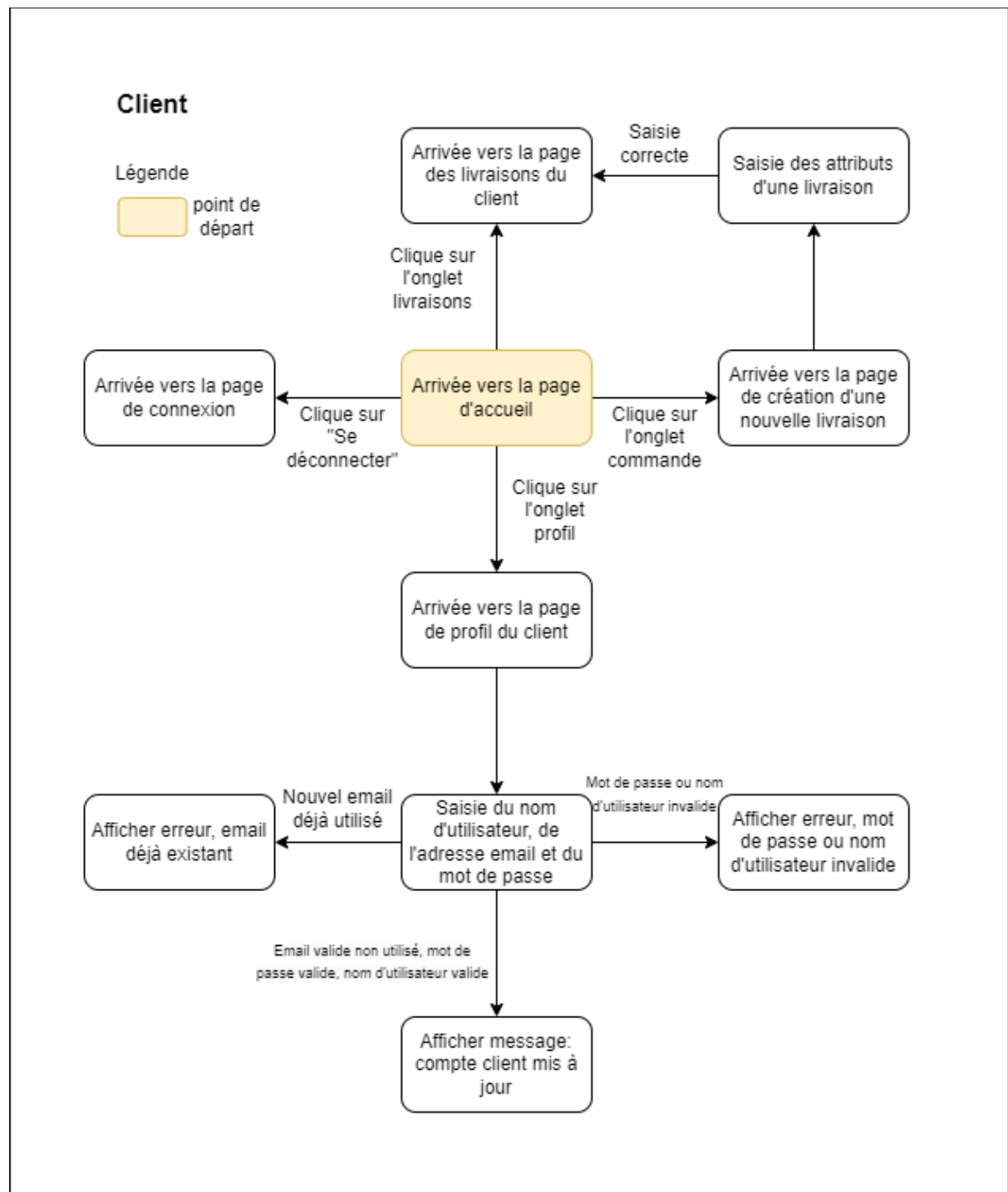
- Login



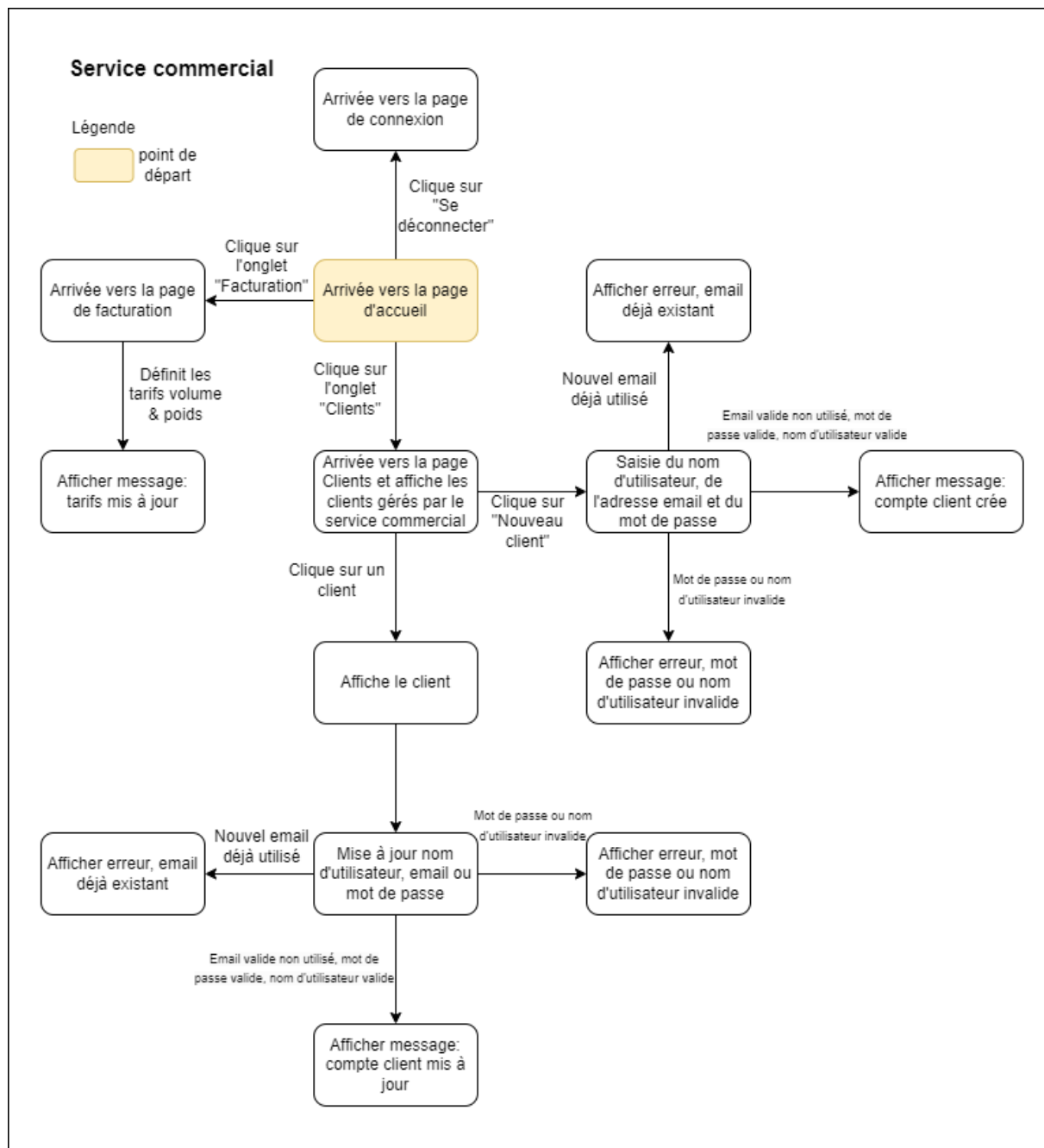
- Register



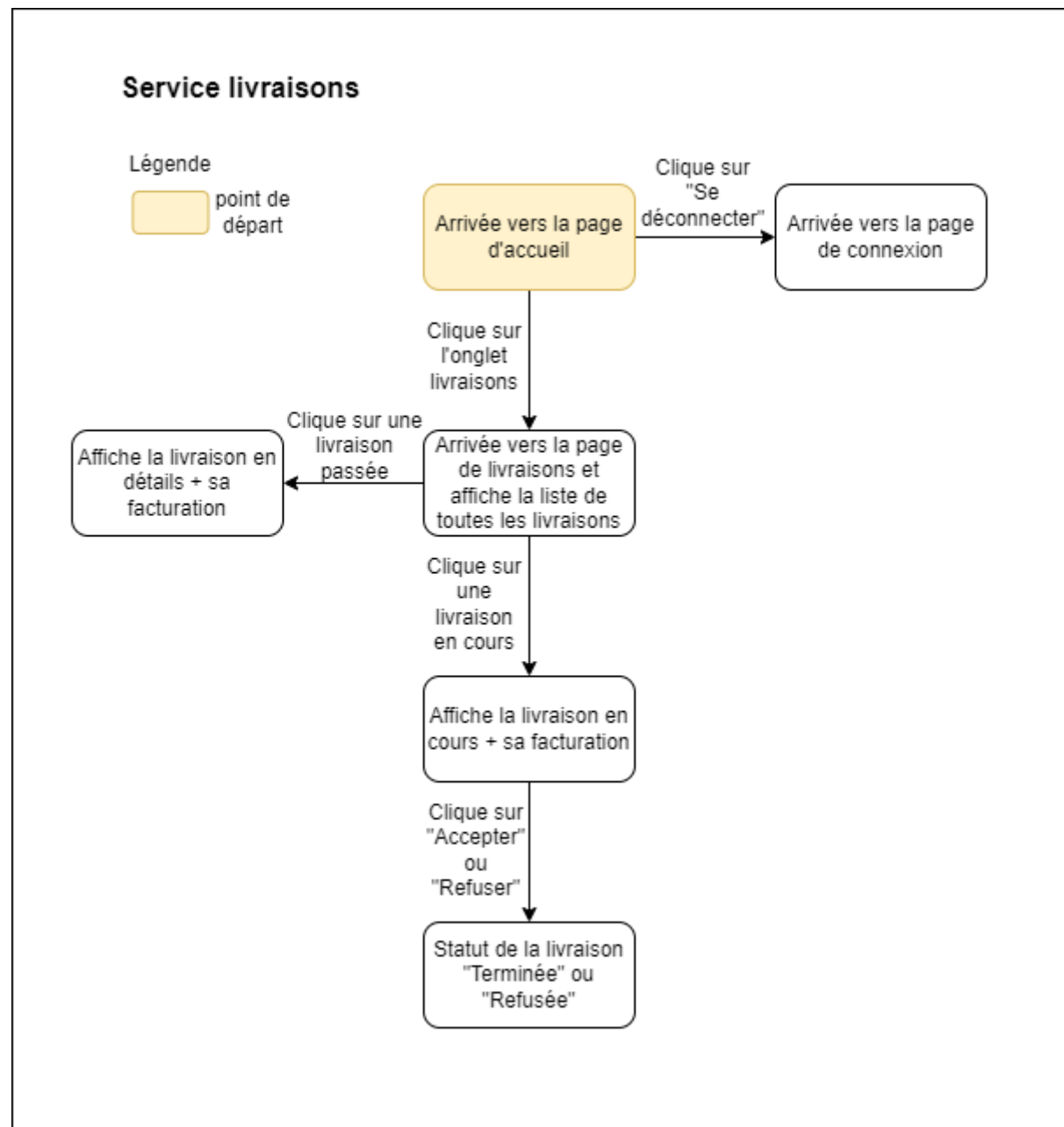
- Client

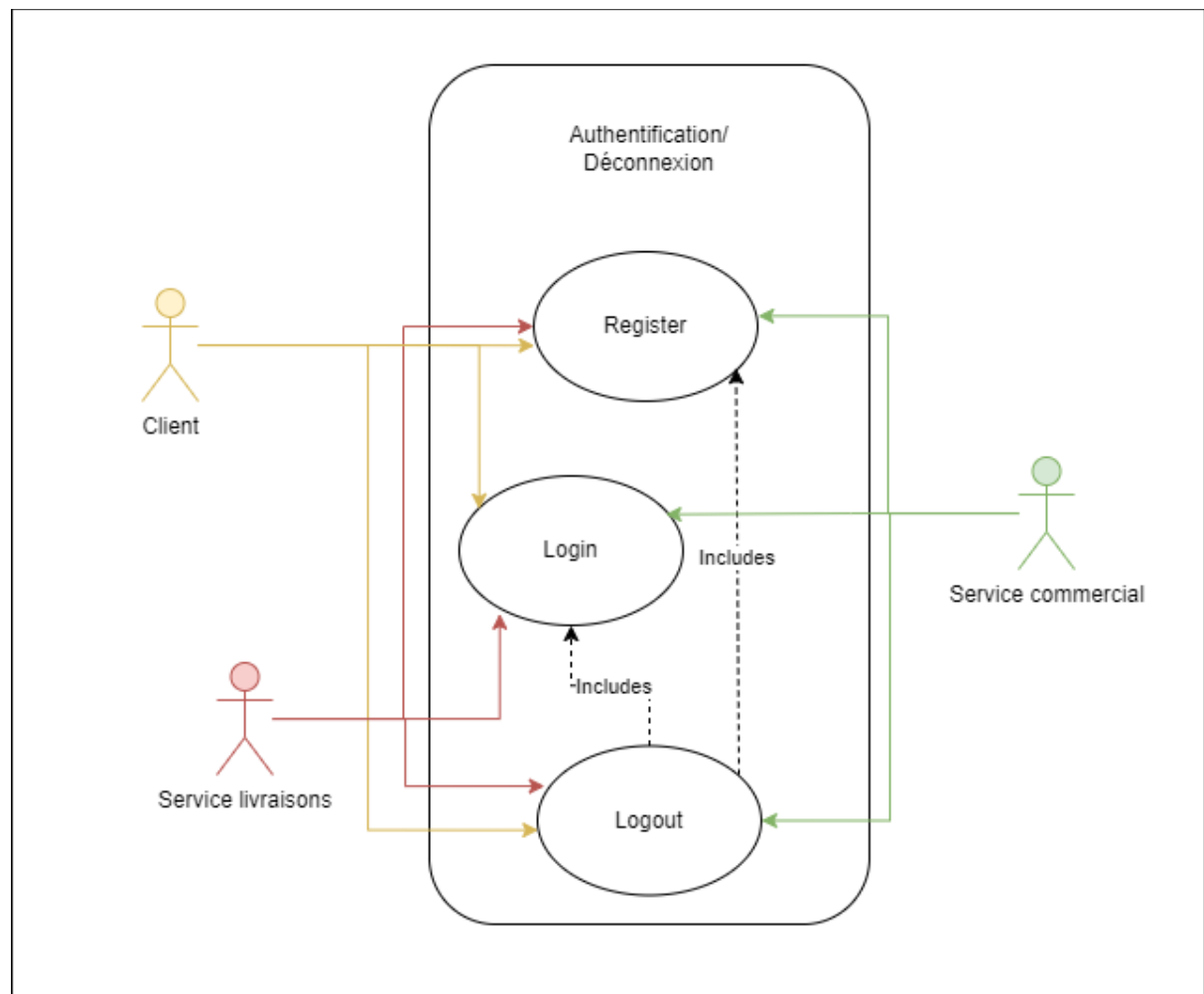


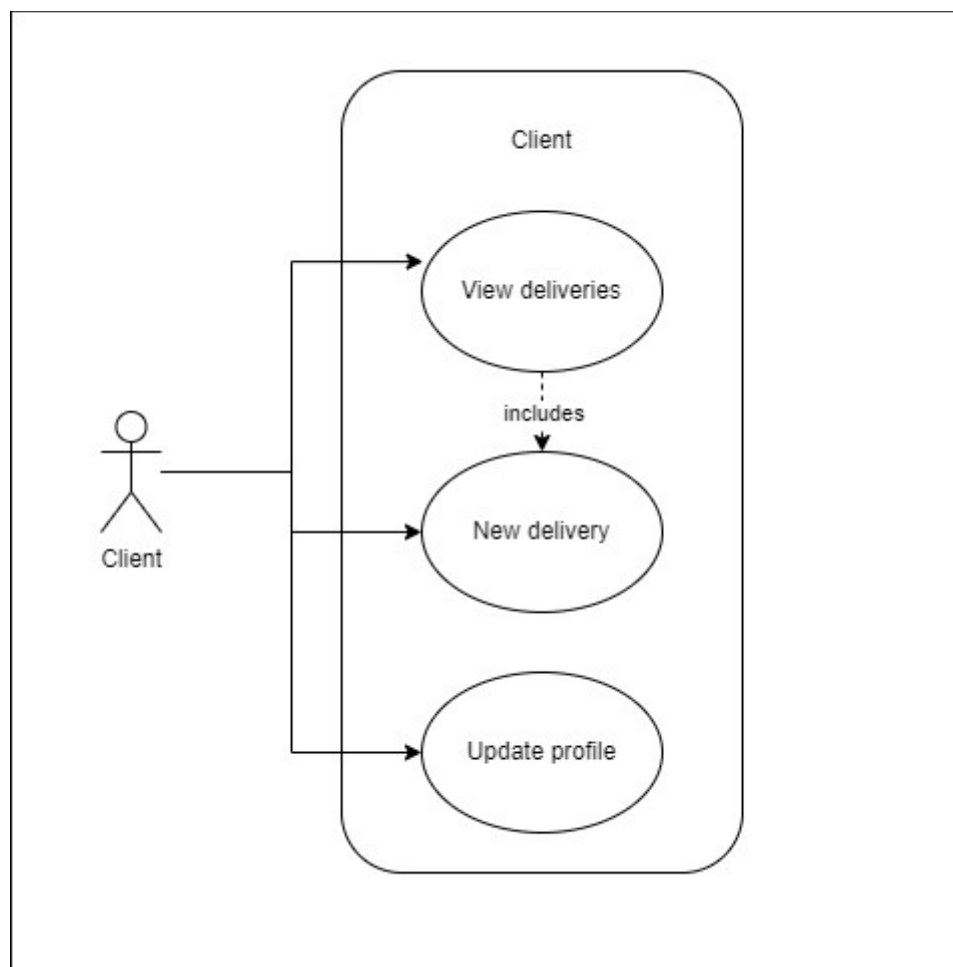
- Service commercial

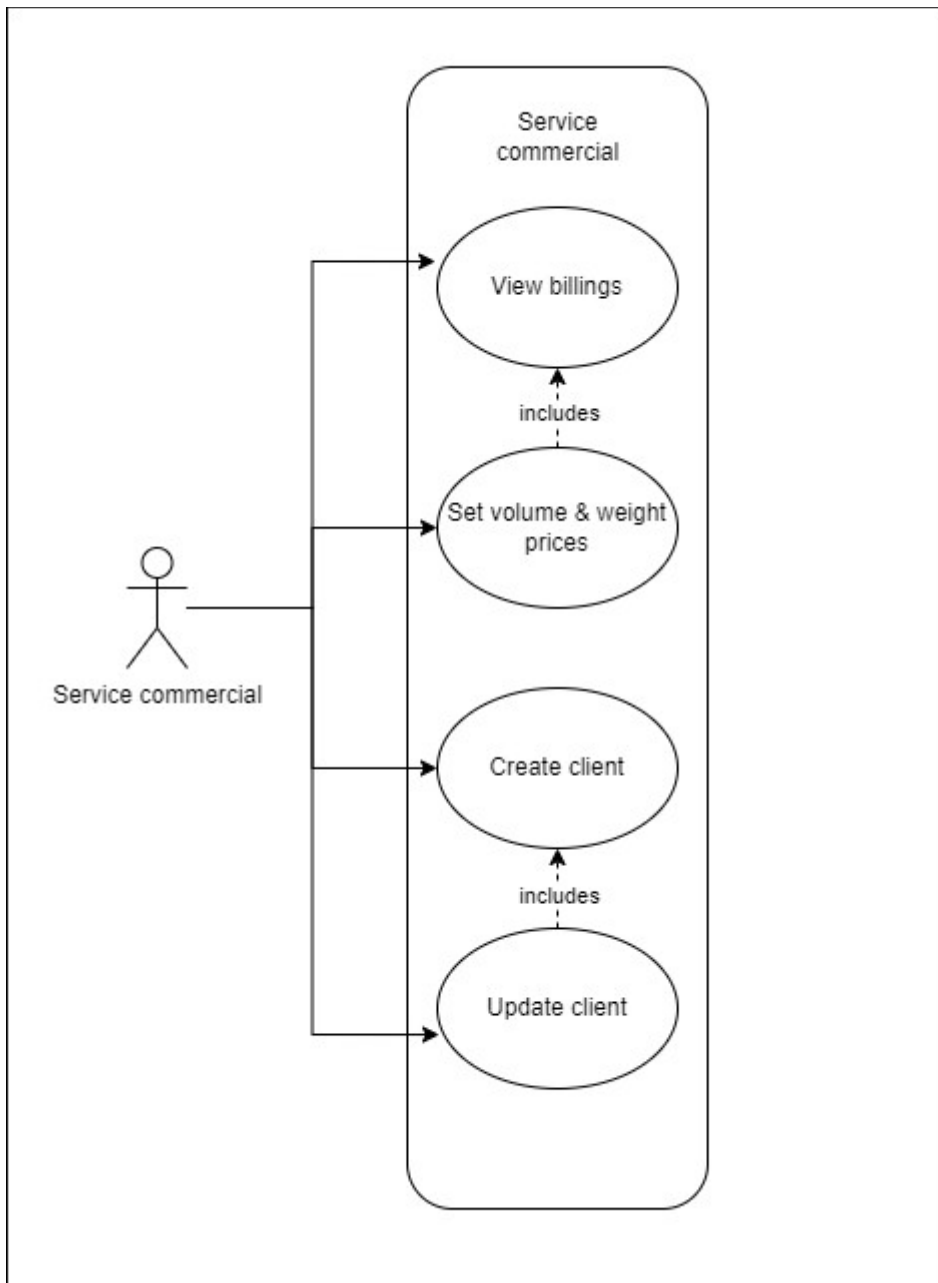


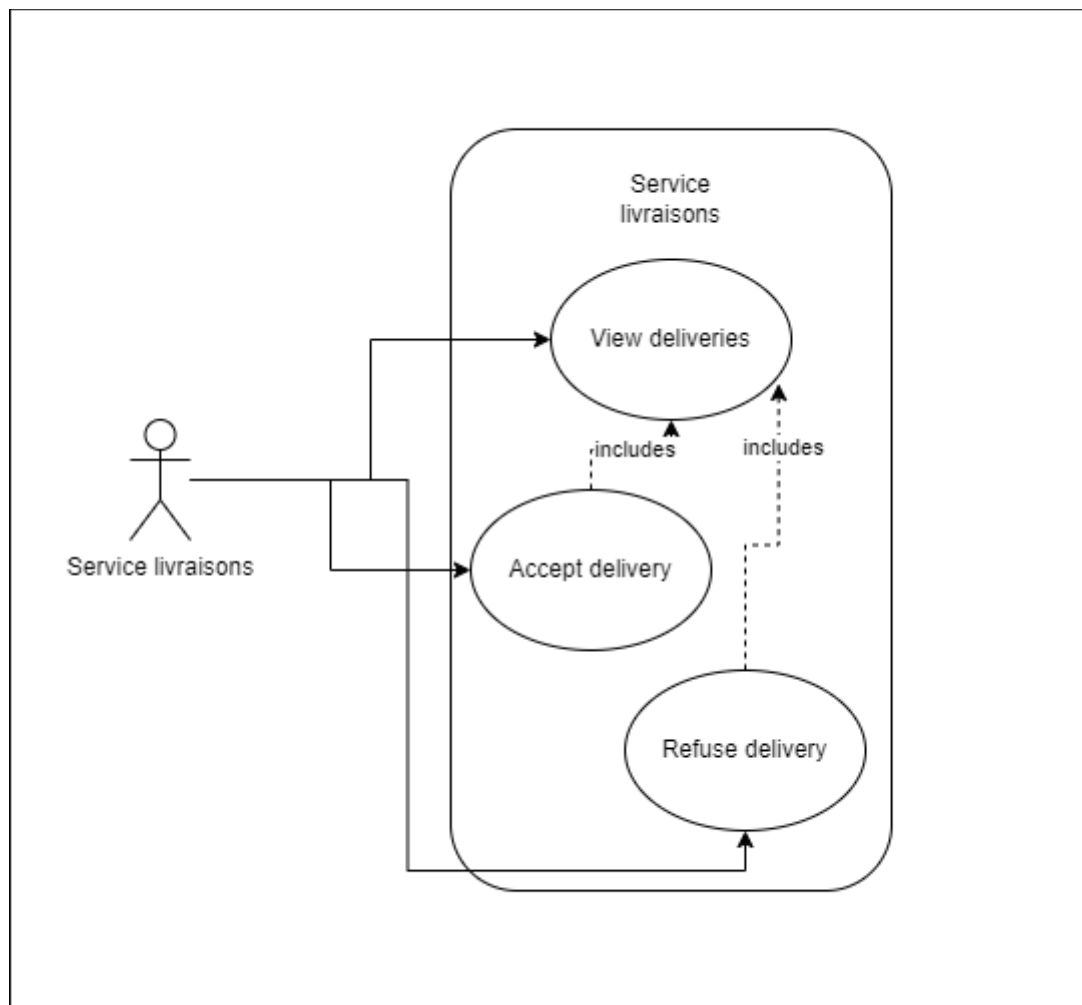
- Service livraison







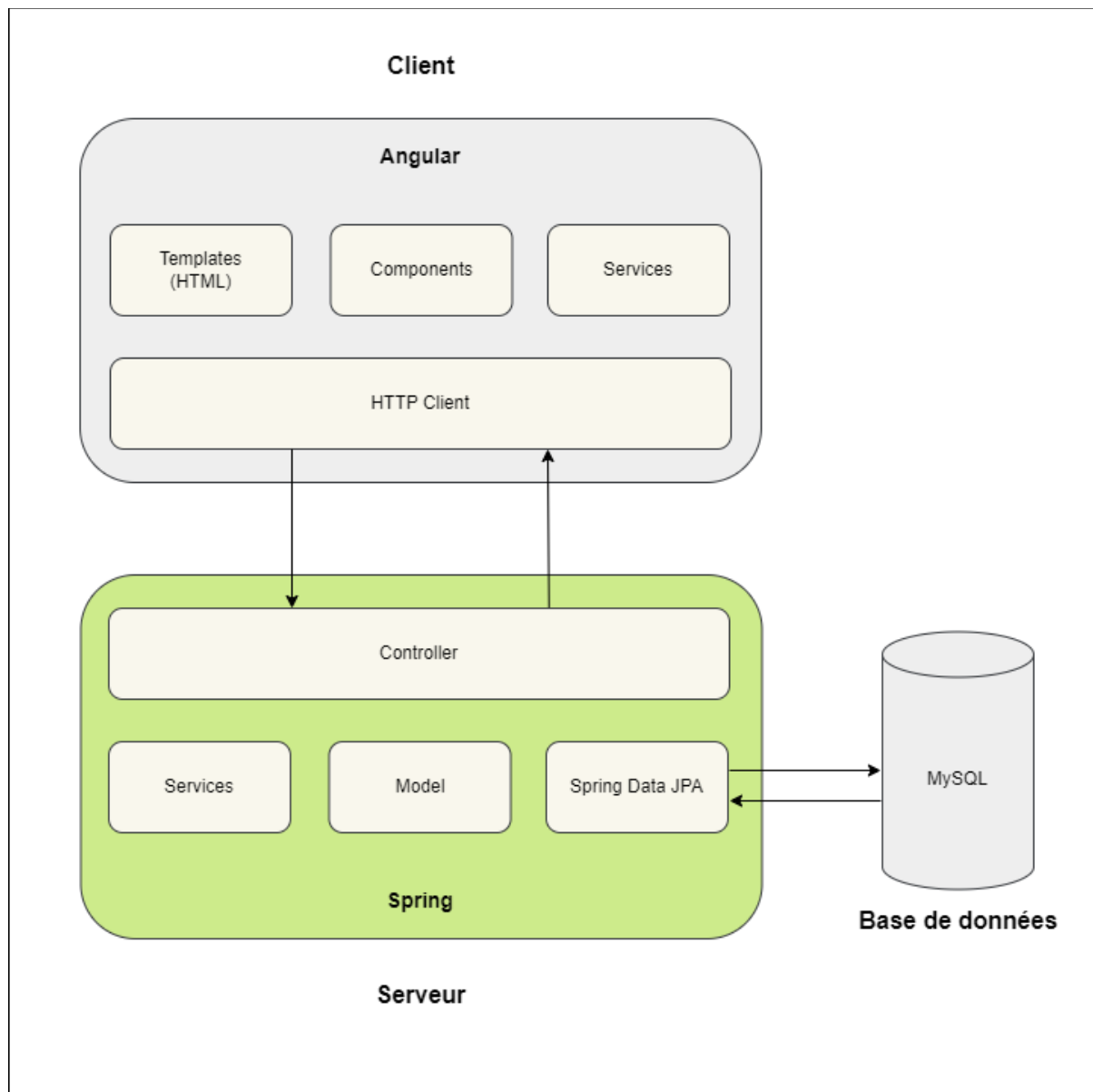




c. Impact Social

Pour les publics en situation de handicap, s'adapter à tous les cas de figures possibles: design contrasté pour les personnes malvoyantes ou dyslexiques, choisir une police lisible, ajouter du texte alternatif aux images, proposer une interface responsive, naviguer au clavier.

3. Description technique



On adoptera l'architecture client-serveur pour réaliser ce projet avec les technologies suivantes:

- Angular: framework front-end, permet de construire des applications SPA (single page application)
- Angular Material: librairie de composants visuels pour l'affichage d'interfaces élégants
- Spring: framework back-end, permet de construire des applications coté serveur avec la version 3.1 donc Java 17 requis.
- Spring Security: librairie de Spring pour sécuriser les applications Java grâce aux méthodes d'authentifications et d'autorisations fournies.
- MySQL: système de gestion de la base de données relationnelles pour stocker les informations comme les clients, les livraisons ...
- Lombok: librairie Java pour simplifier le développement et gagner du temps.
- Git et Github: héberger et sauvegarder le code du projet

- Docker: déployer le projet en production