

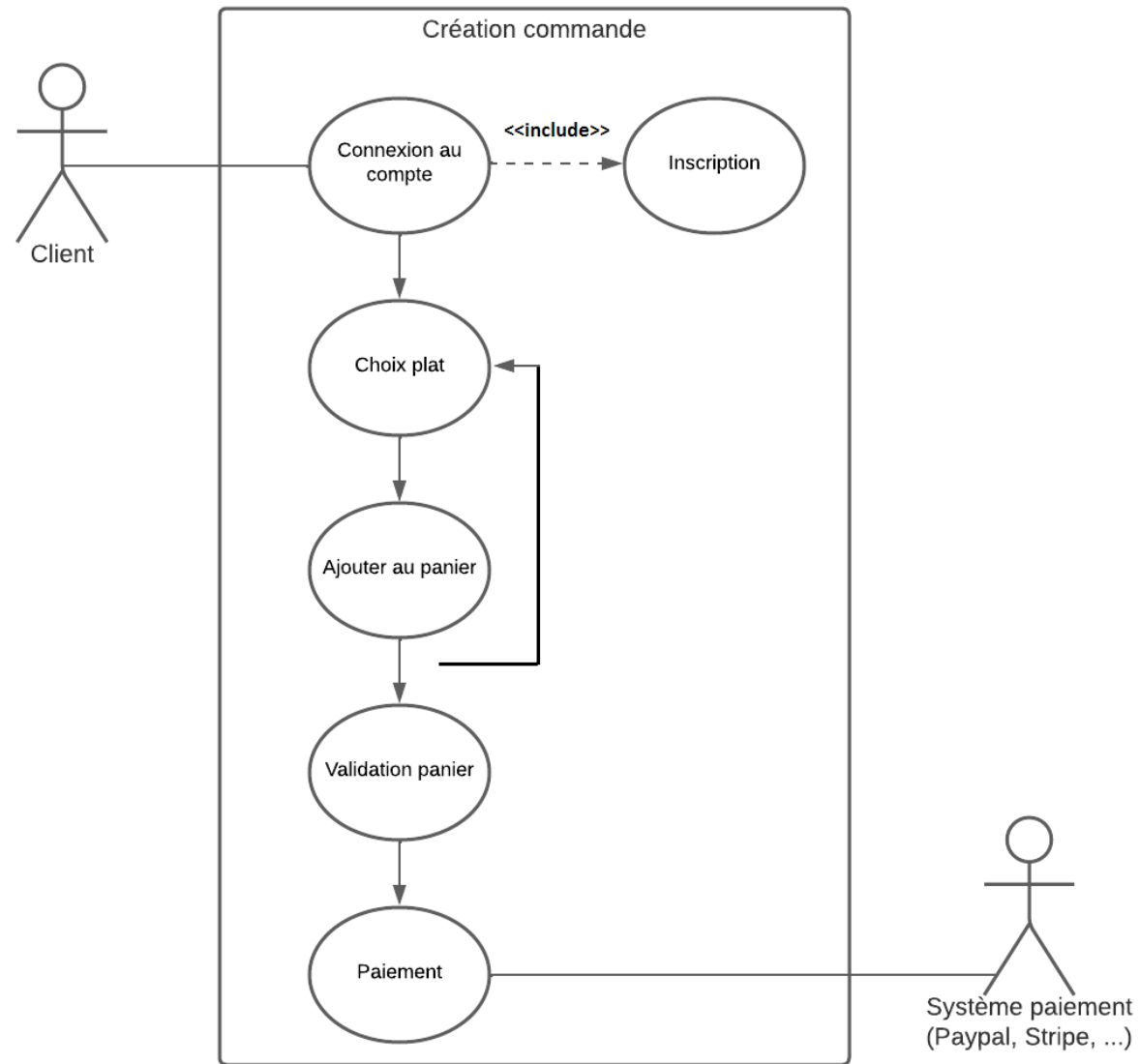
ExpressFood



UML & mySQL

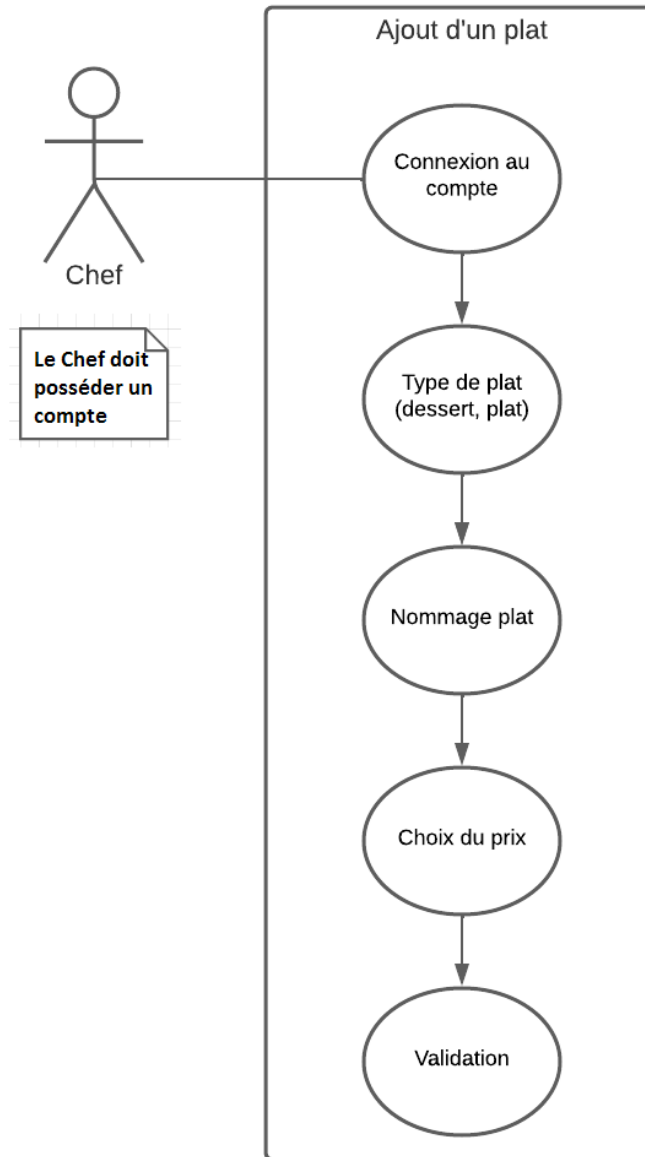
Diagrammes de cas d'utilisations (Usecase) UML

Création d'une commande



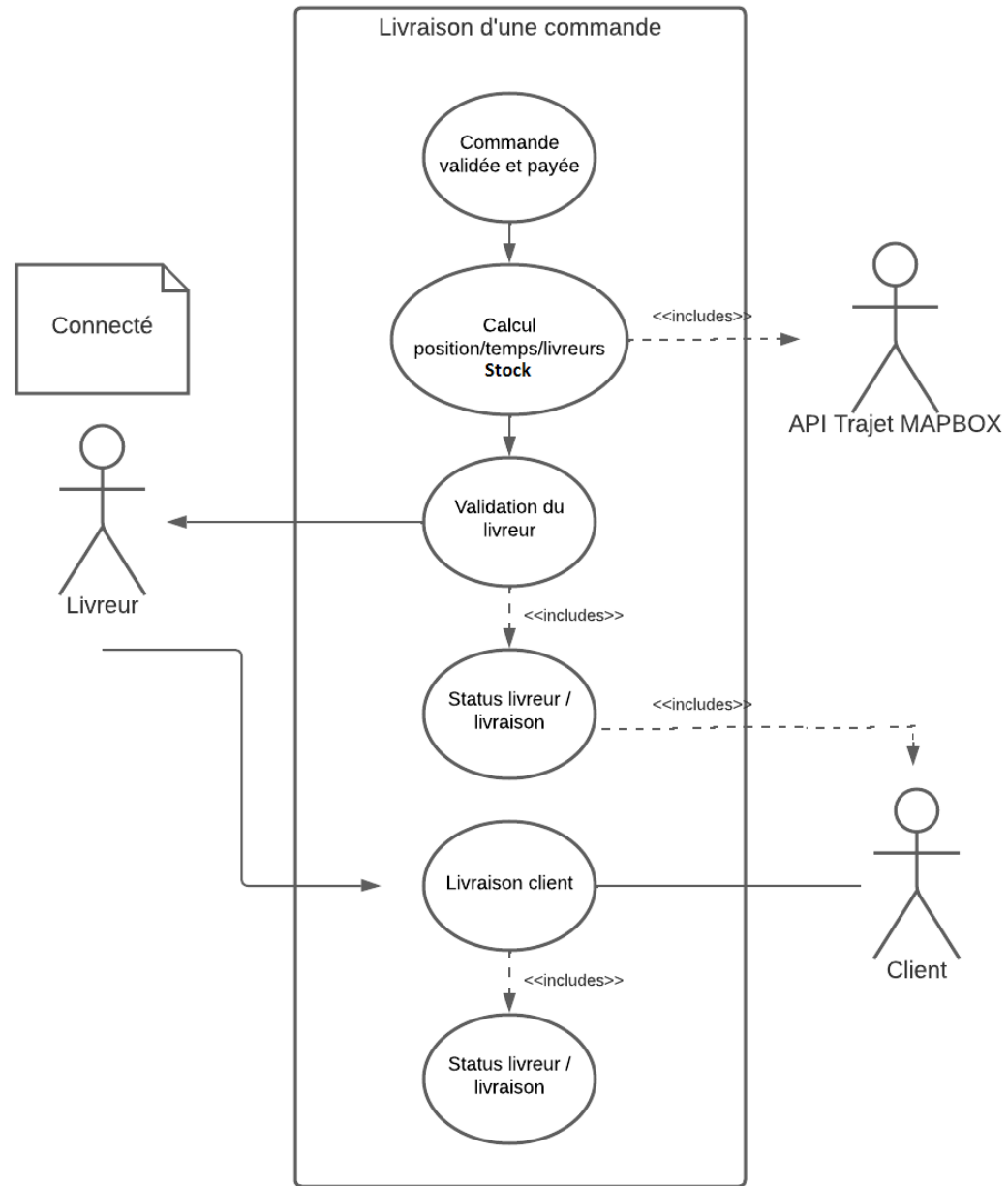
Pour se connecter au système le client doit posséder un compte, dans le cas inverse il doit s'inscrire.

Ajout d'un plat



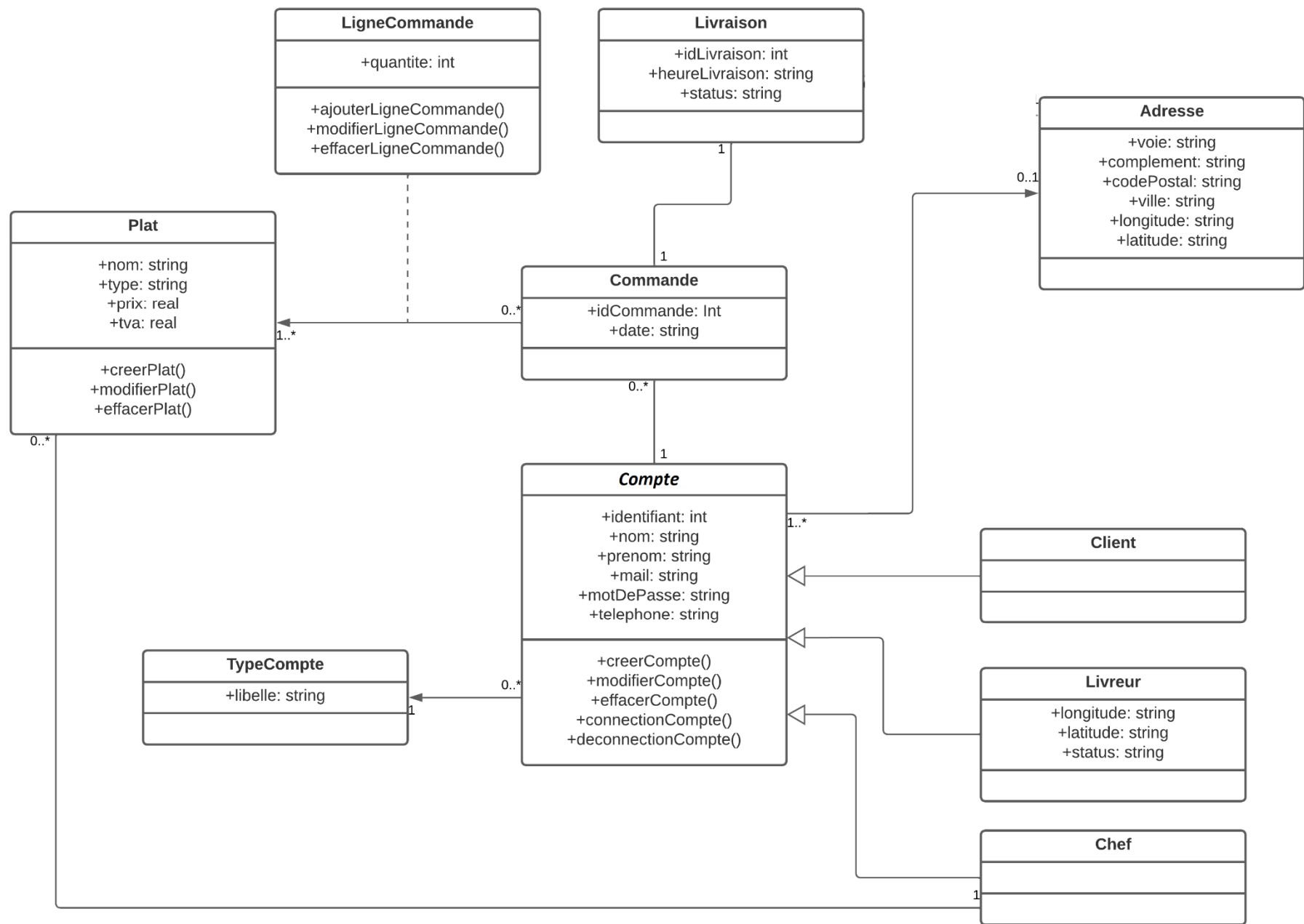
Prérequis : Le chef doit posséder un compte.

Livraison d'une commande

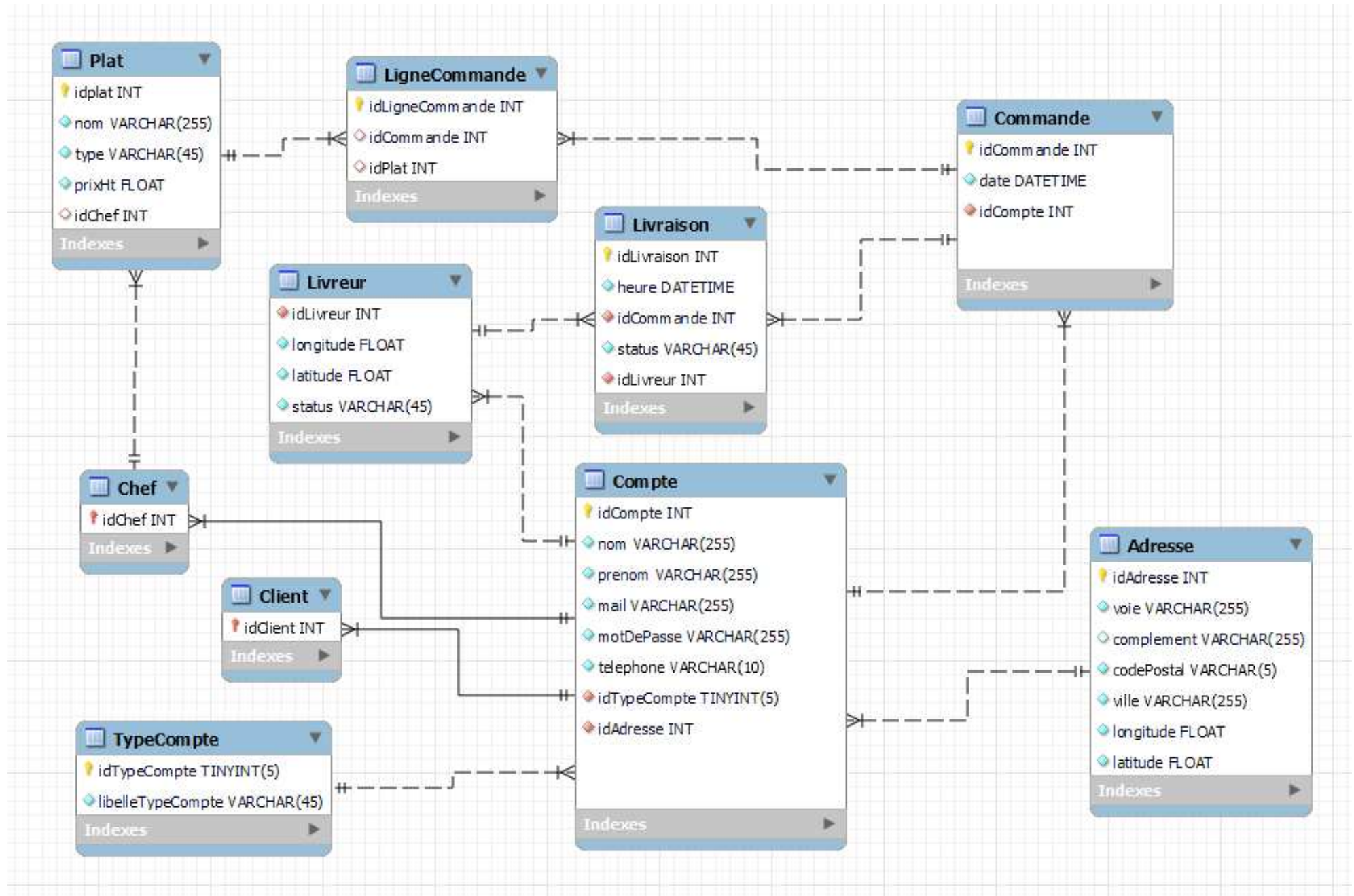


Prérequis : **1**-Le livreur est connecté et le client a validé et payé la commande. **2**-Ce livreur valide la prise de cette livraison.

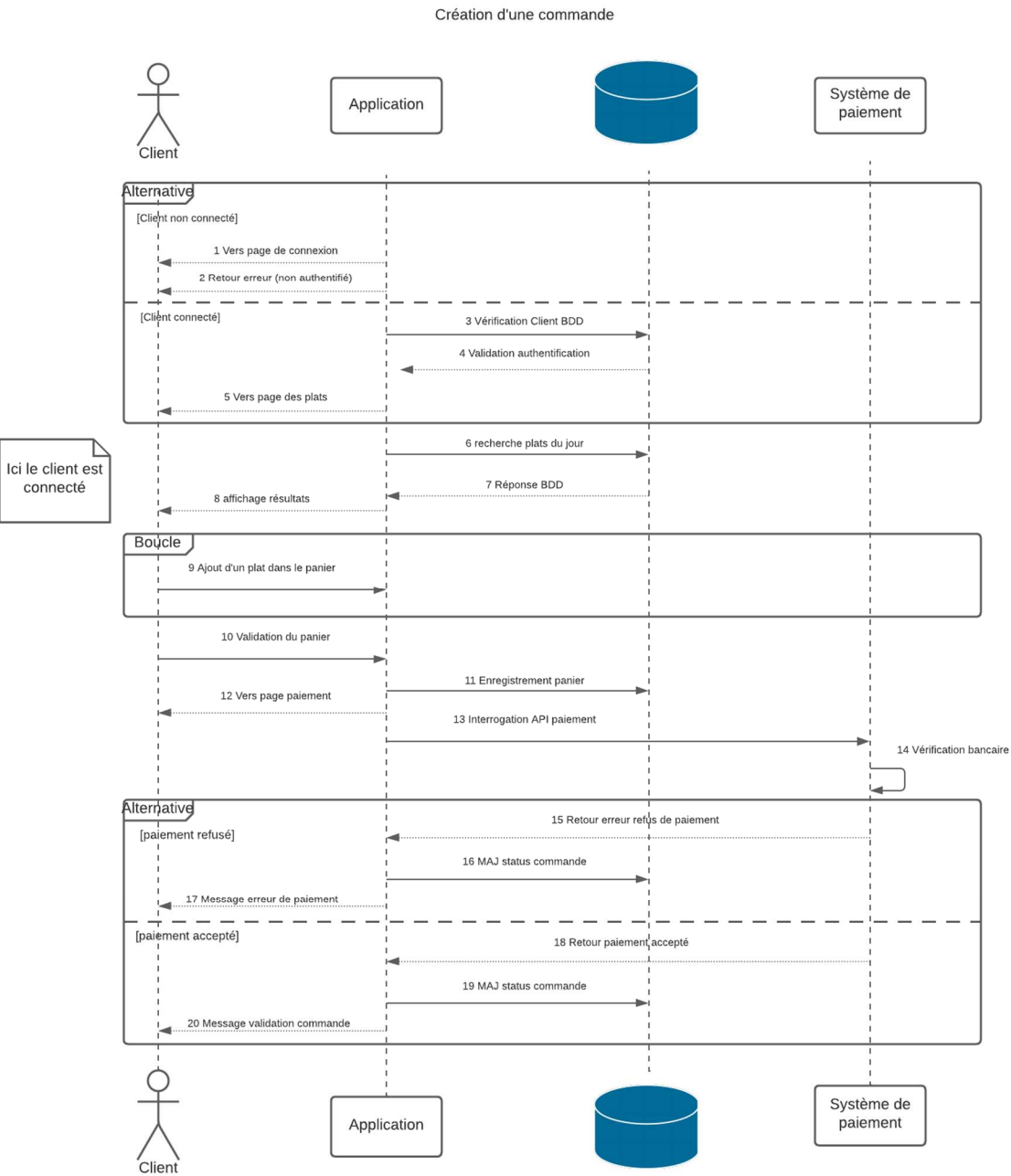
Diagramme de classes UML



Modèle de données

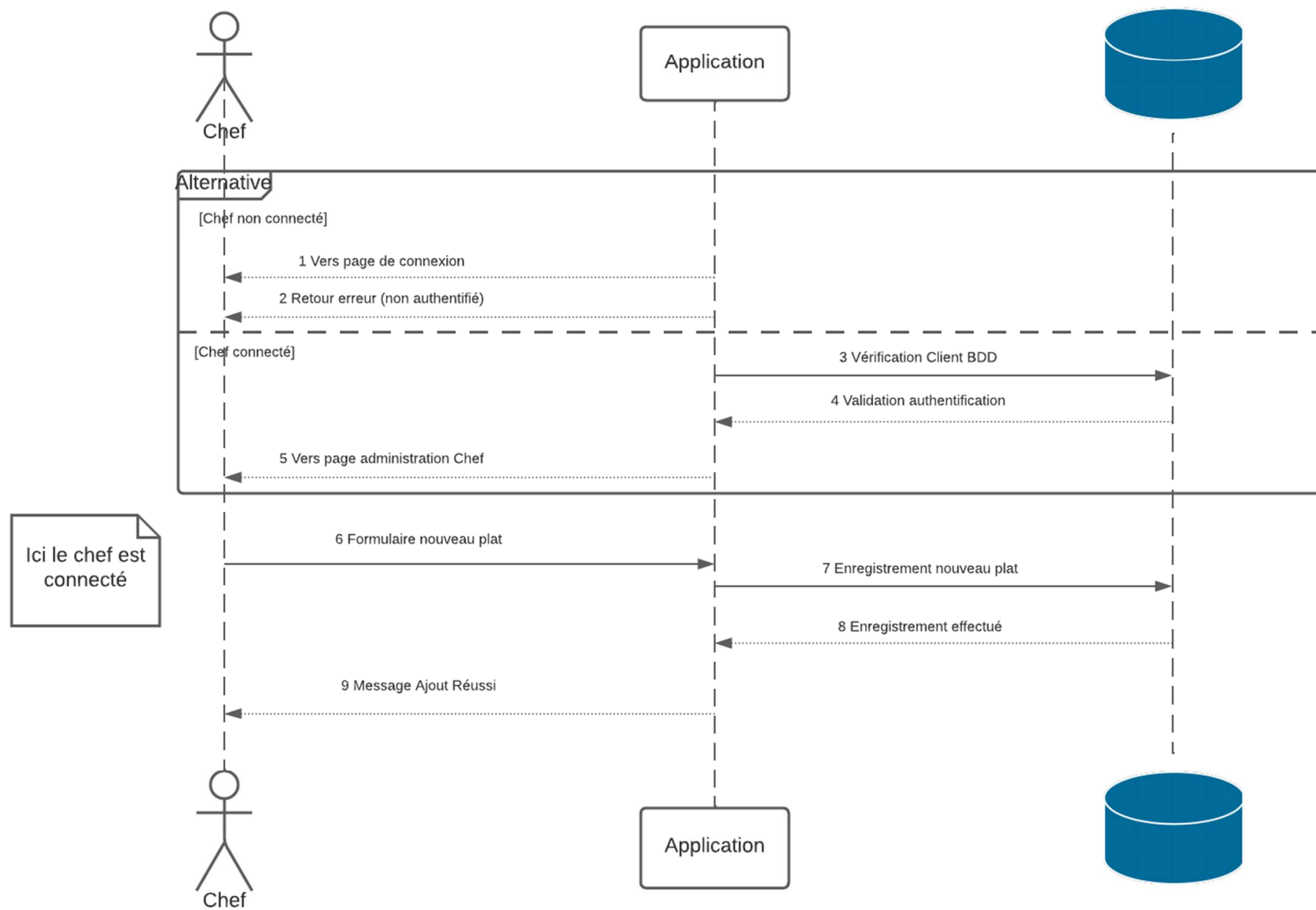


Diagrammes de séquences



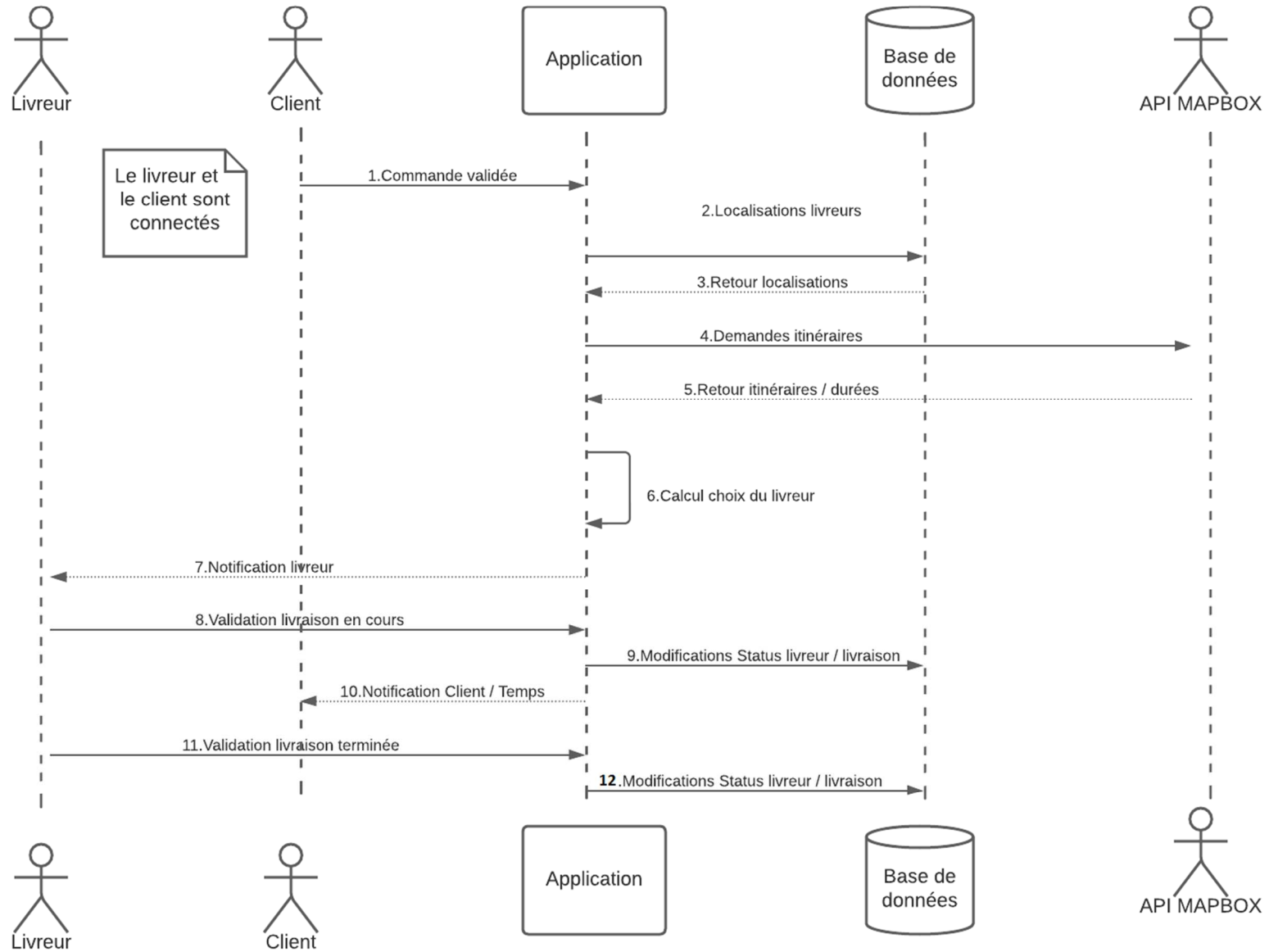
- 1) Si le client n'est pas connecté l'application dirige le client sur la page d'authentification.
- 2) Renvoi d'erreur de l'application vers le client pour notifier qu'il faut se connecter.
- 3) Les données de connexion du client sont comparées à la BDD
- 4) Le retour de la BDD indique à l'application la validation de l'authentification.
- 5) L'application se prépare à diriger le client sur la page des plats
- 6) L'application interroge la BDD sur les plats du jour
- 7) Réponse de la base de données
- 8) L'application affiche la sélection des plats possibles
- 9) Le client ajoute un à un les plats dans son panier (LOOP)
- 10) Le client valide son panier
- 11) L'application enregistre le panier en BDD
- 12) L'application dirige le client vers la page de paiement
- 13) L'application interroge l'API de paiement
- 14) L'API de paiement effectue ses opérateurs de contrôles et de sécurités
- 15) Si retour API = refus de paiement -> 16) L'application mets à jour le statut de la commande en BDD -> 17) Notifie le client de l'erreur
- 18) Si retour API = OK -> 19) L'application mets à jour le statut de la commande en BDD -> 20) Notifie le client de la validation de commande

Ajout d'un plat



- 1) Si le chef n'est pas connecté, l'application dirige le chef vers la page de connexion
- 2) L'application envoie un message d'erreur au chef (non connecté)
- 3) Si identifiants de connexion saisis -> l'application interroge la BDD avec les identifiants saisis
- 4) La BDD réponds à l'application avec une correspondance (validation pour l'application de l'authentification)
- 5) L'application redirige le chef vers la page d'administration du chef
- 6) Le chef saisie le formulaire de nouveau plat
- 7) L'application enregistre les données en BDD
- 8) La BDD réponds à l'application que l'enregistrement est validé
- 9) L'application notifie le chef de la bonne réalisation de l'ajout d'un plat

Livraison



- 1) La commande vient d'être validée (donc payée).
 - 2) L'application interroge la BDD pour récupérer les localisations des livreurs disponibles et ayant le stock nécessaire pour honorer la commande ⚡ (on suppose le système être capable de connaître les positions des livreurs en temps réel).
 - 3) Application reçoit la réponse de la BDD
 - 4) L'application demande à l'API Mapbox les itinéraires des livreurs sélectionnés entre leurs positions et l'adresse de livraison et leurs durées de déplacements.
 - 5) L'API répond à l'application.
 - 6) L'application calcul le « meilleur » livreur (le plus rapide).
 - 7) L'application notifie ce livreur
 - 8) Le livreur valide la prise de livraison à l'application
 - 9) L'application modifie le statut de la livraison (en cours) et celui du livreur (en cours) dans la BDD
 - 10) L'application notifie le client que la livraison est en cours et le temps estimé.
 - 11) Le livreur a bien livré le client et valide la livraison à l'application.
 - 12) L'application modifie le statut de la livraison (livré) et du livreur (libre) dans la BDD
-
- ⚡ L'application calcul le stock du livreur par rapport au nombre standard de plats emportés par un livreur moins la liste des plats qu'il a déjà livré sans revenir compléter son stock transportable. On pose donc l'hypothèse que chaque livreur transporte un nombre de plats déterminés et commun à chaque fois qu'ils partent de chez Expressfood pour « marauder ».