

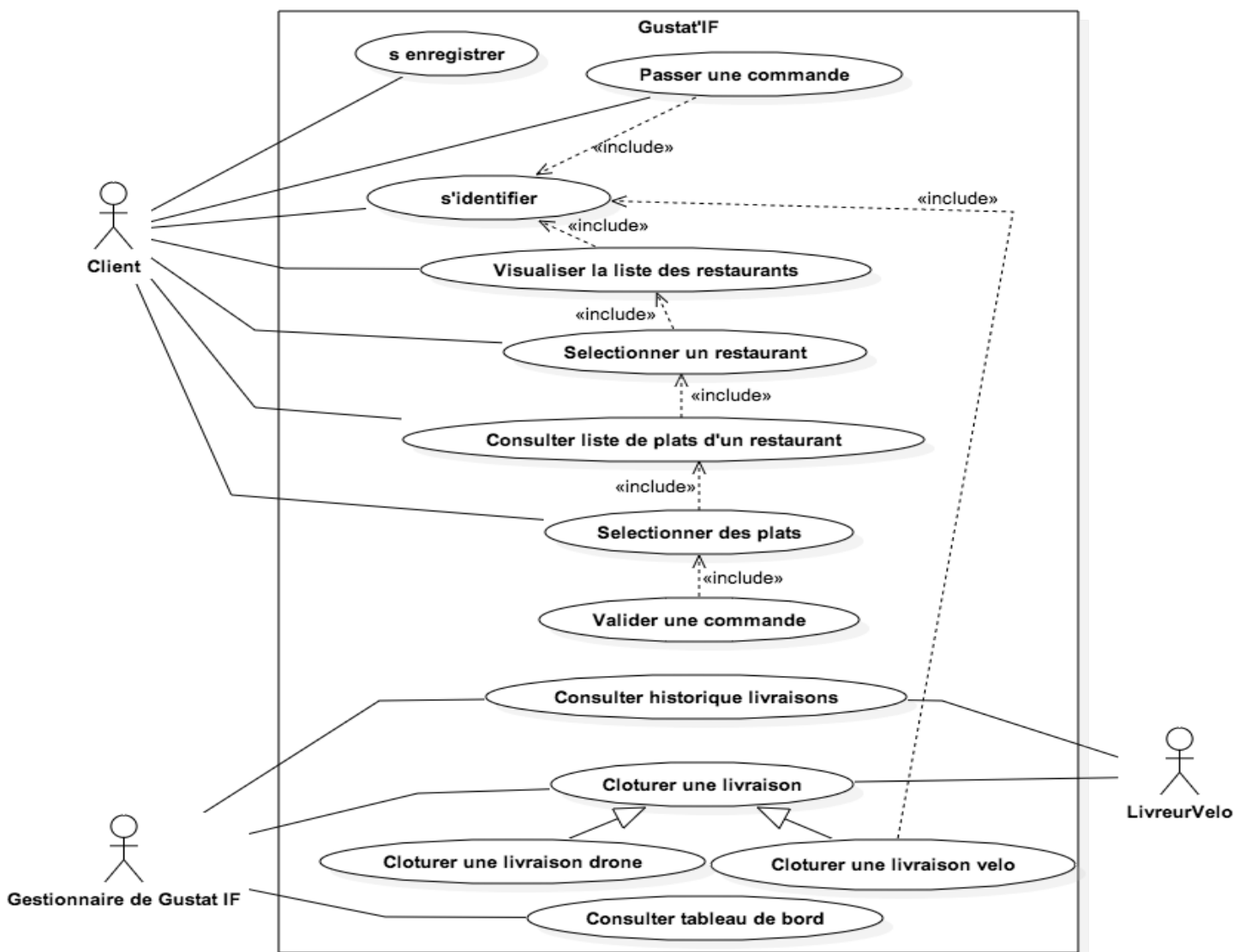
# Compte Rendu

## Introduction :

Nous devons développer une application qui permet au client de commander des plats dans le restaurant de son choix et de se faire livrer par un vélo ou un drone pour une empreinte carbone très réduite. Nous devons donc fournir les 3 interfaces graphiques suivantes :

- Une interface Web pour le client afin de s'inscrire, s'enregistrer et passer des commandes.
- Une IHM pour le gestionnaire des drones afin visualiser sur une carte les emplacements des restaurants, des vélos et des drones et lui permettre de consulter les commandes et confirmer la clôture d'une livraison faite par un drone.
- UNE IHM pour le livreur à vélo afin de s'identifier consulter ses commande et clôturer ses livraison.

## Diagramme de cas d'utilisation :

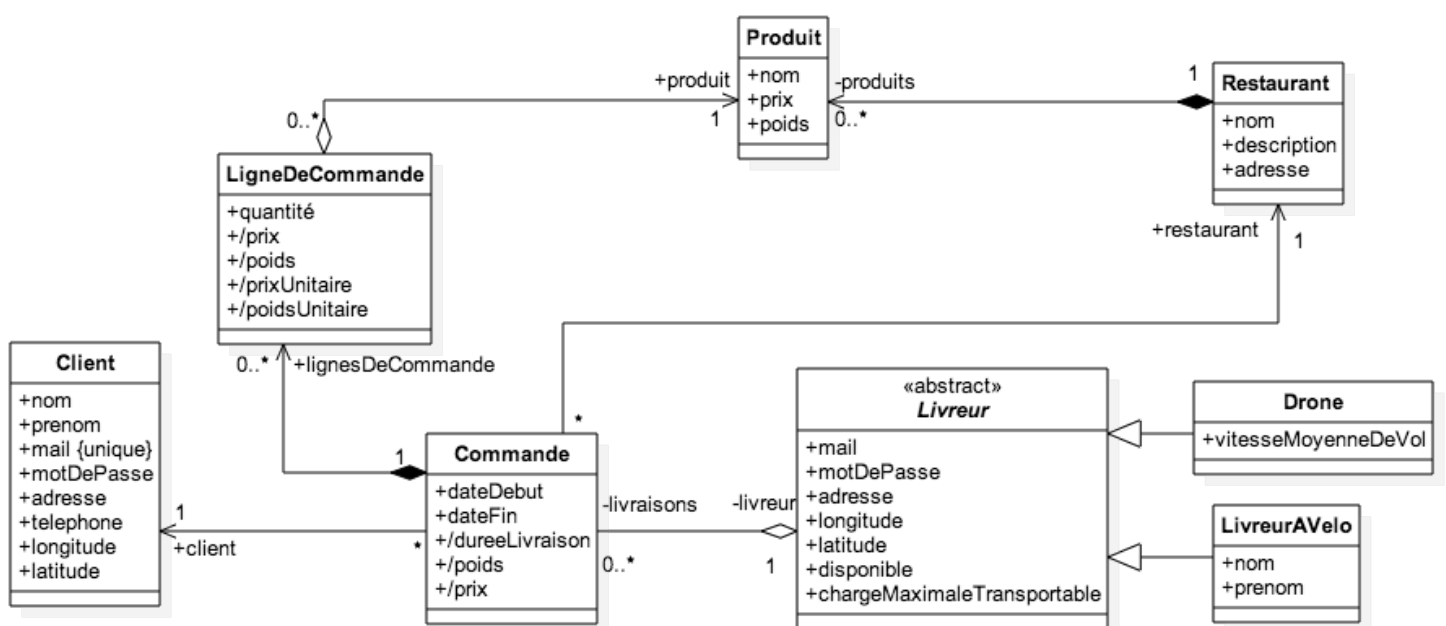


Un client peut s'inscrire à Gustat'IF. Une fois inscrit, celui-ci doit se connecter afin de pouvoir visualiser les différents restaurants. Ensuite il peut sélectionner un de ces restaurant, choisir des produits de ce restaurant et valider sa commande.

Un livreur à vélo peut consulter l'historique de ses livraisons qu'il a effectuées et clôturer celles-ci ; il doit au préalable se s'identifier.

Le Gestionnaire de Gustat'IF peut quant-à lui clôturer les livraisons effectuées par les drones. Il peut consulter le tableau de bord lui permettant de visualiser sur une carte les différents livreurs, restaurants clients, etc.

## Modèle du domaine :



## Spécifications des services fournis :

### ➤ ServiceMetier.java

boolean createClient(Client client)	Ce service prend en paramètre un Client et que l'enregistre en base de données. Cette méthode renvoie true si succès et false sinon. Ce service envoie un mail indiquant au client pour lui indiquer si l'enregistrement à bien fonctionné. Attention, le mail du Client ne doit pas être déjà présent en base de données. Tous les attributs du Client doivent être setter sauf la latitude et la longitude qui sont calculées dans cette méthode.
boolean clientExists(String mail, String motDePasse)	Ce service vérifie si un Client, ayant l'adresse mail et le mot de passe donnés en paramètre, est présent en base de données. Renvoie true si le Client existe déjà dans la base de donnée, false sinon.

Client updateClient(Client client)	Ce service permet de mettre à jour un Client en base de données. Revoie le Client mis à jour ou null si un problème s'est produit.
Client findClientById(Long id)	Ce service permet de retrouver un Client en fonction de son identifiant en base de données. Renvoie le Client trouvé. Renvoie null si aucun Client ayant cette id n'est présent ou si une erreur s'est produite.
Client findClientByMailAndPassword(String mail, String password)	Ce service retourne un Client, ayant l'adresse mail et le mot de passe donnés en paramètre. Retourne null si aucun Client n'est trouvé ou si une erreur s'est produite.
Client findClientByMail(String mail)	Ce service retourne un Client, ayant l'adresse mail donné en paramètre. Retourne null si aucun Client n'est trouvé ou si une erreur s'est produite.
List<Client> findAllClients()	Ce service retourne tous les objets de type Client présents dans la base de données. Retourne null si une erreur s'est produite.
Restaurant findRestaurantById(Long id)	Ce service permet de retrouver un Restaurant en fonction de son identifiant en base de données. Renvoie le Restaurant trouvé. Renvoie null si aucun Restaurant ayant cette id n'est présent ou si une erreur s'est produite.
List<Restaurant> findAllRestaurants()	Ce service retourne tous les objets de type Restaurant présents dans la base de données. Retourne null si une erreur s'est produite.
Produit findProduitById(Long id)	Ce service permet de retrouver un Produit en fonction de son identifiant en base de données. Renvoie le Produit trouvé. Renvoie null si aucun Produit ayant cette id n'est présent ou si une erreur s'est produite.
List<Produit> findAllProduits()	Ce service retourne tous les objets de type Produit présents dans la base de données. Retourne null si une erreur s'est produite.
boolean createCommande(Commande commande)	Ce service créer une Commande, donnée en paramètre, en base de donnée. Ce service affecte à cette commande un Livreur disponible. Si une erreur se produit lors de l'affectation d'un livreur, ce service réessaye d'affecter un livreur (jusqu'à 10 tentatives). Renvoie true si la commande a bien été enregistré dans la base données, false sinon. Ce service envoie un mail au Livreur si c'est un LivreurVelo et que l'enregistrement a fonctionné et envoie un mail au Client qui passe la commande pour lui indiquer si la commande est validée ou non.
Commande validateCommande(Commande commande)	Ce service valide une Commande donnée en paramètre. Il faut que la datefin de la Commande soit settée. Ce service met à true l'attribut disponible du livreur effectuant la livraison de la Commande. Renvoie true si la validation a fonctionné, false sinon.

List<Commande> findCommandesByLivreurId(Long id)	Ce service renvoie les objets de type Commande associés au Livreur ayant l'id donné en paramètre. Revoie une List<Commande> si aucun problème, null sinon.
List<Commande> findAllCommande()	Ce service retourne tous les objets de type Commande présents dans la base de données. Retourne null si une erreur s'est produite.
List<Commande> findAllCommandeNotEnded()	Ce service retourne tous les objets de type Commande ayant une datefin null présents dans la base de données. Retourne null si une erreur s'est produite.
Livreur updateLivreurById(Livreur livreur)	Ce service permet de mettre à jour un Livreur en base de données. Revoie le Livreur mis à jour ou null si un problème s'est produit.
LivreurVelo findLivreurVeloByMailAndPassword(String mail, String password)	Ce service retourne un LivreurVelo, ayant l'adresse mail et le mot de passe donnés en paramètre. Retourne null si aucun LivreurVelo n'est trouvé ou si une erreur s'est produit.
List<LivreurDrone> findLivieursDronesByMailAndPassword(String mail, String password)	Ce service retourne tous les objets de type LivreurDrone ayant le mail et le mot de passe données en paramètre présents dans la base de données. Retourne null si une erreur s'est produite.
Livreur findLivreurById(Long id)	Ce service permet de retrouver un Livreur en fonction de son identifiant en base de données. Renvoie le Livreur trouvé. Renvoie null si aucun Livreur ayant cette id n'est présent ou si une erreur s'est produite.
List<Livreur> findAllLivieurs()	Ce service retourne tous les objets de type Livreur présents dans la base de données. Retourne null si une erreur s'est produite.
List<LivreurDrone> findAllLivieursDrones()	Ce service retourne tous les objets de type LivreurDrone présents dans la base de données. Retourne null si une erreur s'est produite.
List<LivreurVelo> findAllLivieursVelos()	Ce service retourne tous les objets de type Livreur Velo présents dans la base de données. Retourne null si une erreur s'est produite.
List<Livreur> findAllLivieursDisponibles()	Ce service retourne tous les objets de type Livreur ayant leur attribut disponible égale à true présents dans la base de données. Retourne null si une erreur s'est produite.

### ➤ ServiceTechnique.java

public boolean sendRealMail(String recipientMail, String subject, String body)	Ce service permet d'envoyer un vrai mail de la part de <u>gustatif.b3125@gmail.com</u> . Le mail est envoyé à recipientMail. Le mail à pour objet subject et pour corps body.
--	---

public void sendFakeMail(String recipientMail, String subject, String body)	Affiche un mail en console dont le contenu est body, dont le destinataire est recipientMail, dont l'objet est subject et dont l'expéditeur est <u>gustatif.b3125@gmail.com</u> .
public Livreur chooseBestLivreur(Commande commande, List<Livreur> livreurs )	Ce service permet de retourner le Livreur qui pourra effectuer la Commande le plus rapidement. Retourne null si un Livreur n'est pas de type LivreurVelo ou LivreurDrone ou si la List livreurs est vide. Attention, ce service ne vérifie pas la disponibilité des livreurs.

### ➤ Autres

Le main présent dans la classe fr.insalyon.dasi.gustatif.test.JeuDeDonnees permet de créer 5 objets de type LivreurVelo et 5 objets de type LivreurDrone.

## Specification des IHM :

### 1. Client

#### ➤ Inscription ou identification du client

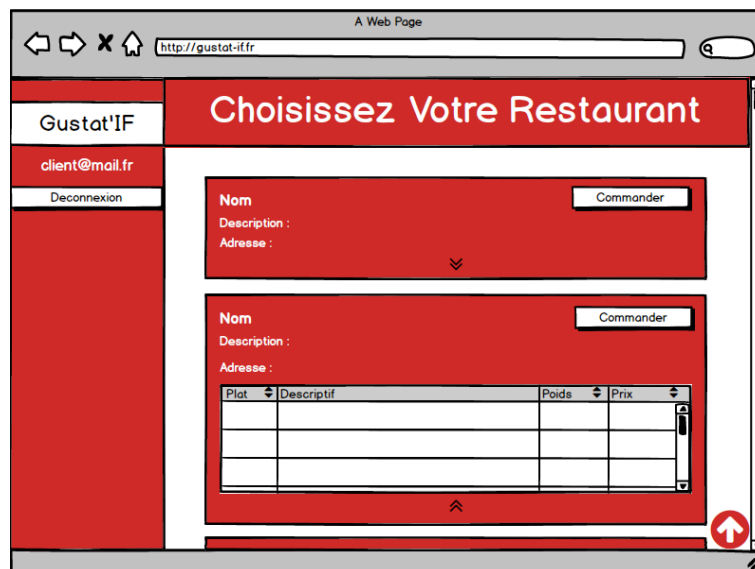
Pour interagir avec l'application, le client doit d'abord s'inscrire puis s'identifier. Il a donc accès à une page d'inscription/identification. S'il est déjà inscrit il peut s'identifier avec son mail et son mot de passe, sinon il doit renseigner les informations dans le formulaire d'inscription ci-dessous afin d'être ajouté à la base de donnée. Le client reçoit un mail de confirmation/infirmation de son inscription.

Intention / Besoin	Bouton / Contrôle	Action	Réponse
S'inscrire	Champs texte	Remplir tous les champs	Aucune
	Bouton « inscription »	Clic	Envoi mail au client

S'identifier	Champs texte	Remplir tous les champs	Aucune
	Bouton « Connexion »	Clic	Charge la page Choix du restaurant

## ➤ Choix du restaurant

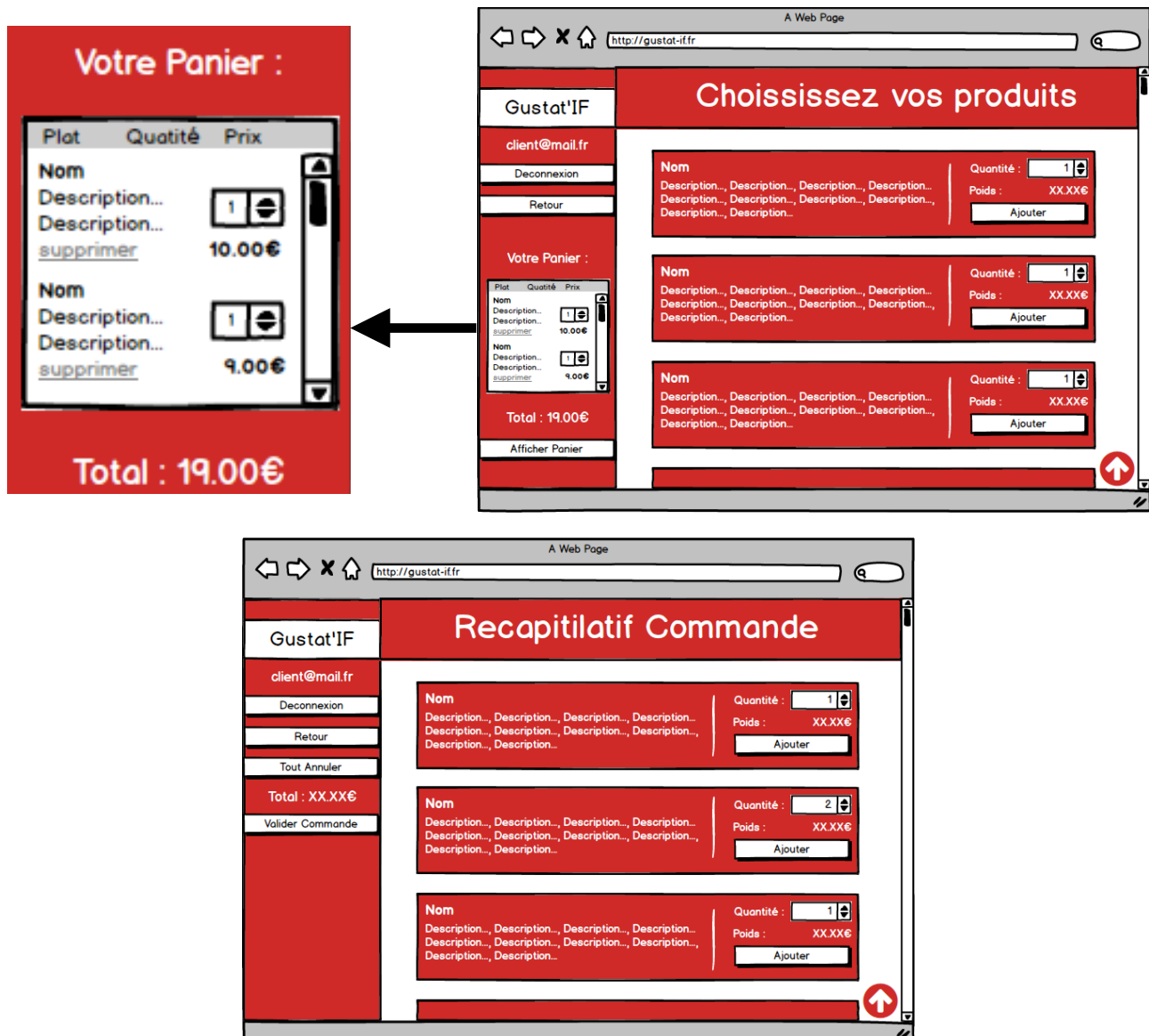
Une fois le client connecté, le client est redirigé vers la liste de tous les restaurants avec leurs brèves descriptions respectives. Lorsque le client souhaite afficher les produit d'un restaurant donné, il lui suffit d'appuyer sur le bouton « ≡ » (le bouton « ≡ » servant alors à les masquer). Cela affiche un tableau contenant les détails de tous les produits proposé par ce restaurant. Cet ordre a été choisi afin d'interdire le choix de plusieurs plats dans des restaurant différent et ainsi avoir un restaurant par commande. Pour valider le choix de ce restaurant et être dirigé vers la page suivante afin de sélectionner les plats, le client doit appuyer sur le bouton « Commander ».



Intention / Besoin	Bouton / Contrôle	Action	Réponse
Afficher les plats d'un restaurant	Bouton « ≡ »	Clic	Affiche le tableau des produit du restaurant ainsi que le bouton « Choisir ce restaurant »
Afficher les plats d'un restaurant	Bouton « ≡ »	Clic	Affiche le tableau des produit du restaurant ainsi que le bouton « Choisir ce restaurant »
Choisir un restaurant	Bouton « Commander »	Clic	Charge la page choix des produits du restaurant choisi
Se déconnecter	Bouton « Déconnexion »	Clic	Charge la page de d'identification / inscription

## ➤ Choix des produits

Le client peut alors modifier la quantité de chaque plat qu'il souhaite commander puis l'ajouter à sa commande. Il peut visualiser son panier dans le menu de gauche. Lorsqu'il a fini, il peut afficher son panier pour voir un récapitulatif de la commande, le total à payer et ainsi pouvoir la valider.

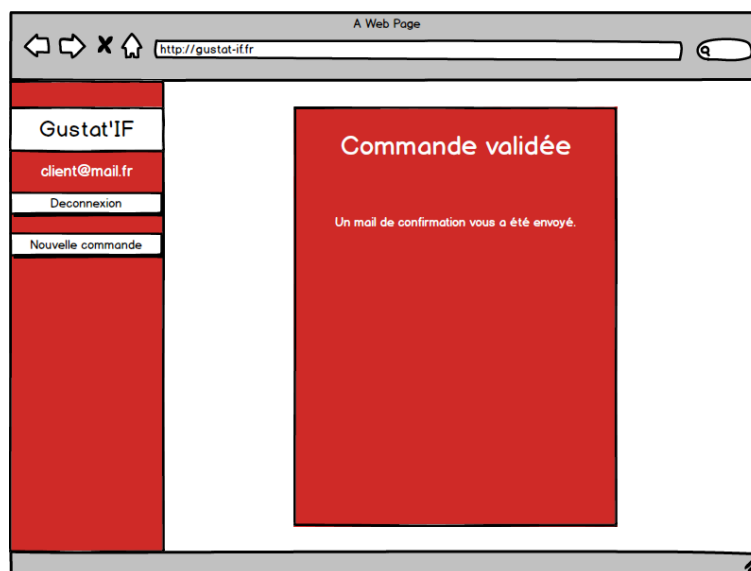


Intention / Besoin	Bouton / Contrôle	Action	Réponse
Choisir quantité du plat	Spinbox/Spinner	Incrémenter/ décrémenter	Aucune
Ajouter la selection au panier	Bouton « Ajouter »	Clic	Ajouter le produit au panier et met à jour le poids et le prix
Modifier la quantité d'un produit du panier	Spinbox/Spinner	Incrémenter/ décrémenter	Aucune

Supprimer un produit du panier	Bouton « supprimer » dans le panier	Clic	Enlève le produit du panier
Afficher le panier	Bouton « Afficher le panier »	Clic	Affiche le total ainsi que le bouton « Valider Commande »
Valider la Commande	Bouton « Valider commande »	Clic	Charge la page suivante.
Se déconnecter	Bouton « Déconnexion »	Clic	Charge la page de d'identification / inscription

### ➤ Commande validée

Cette interface sert de confirmation au client que sa commande a bien été acceptée. Elle lui donne la possibilité d'effectuer une nouvelle commande.



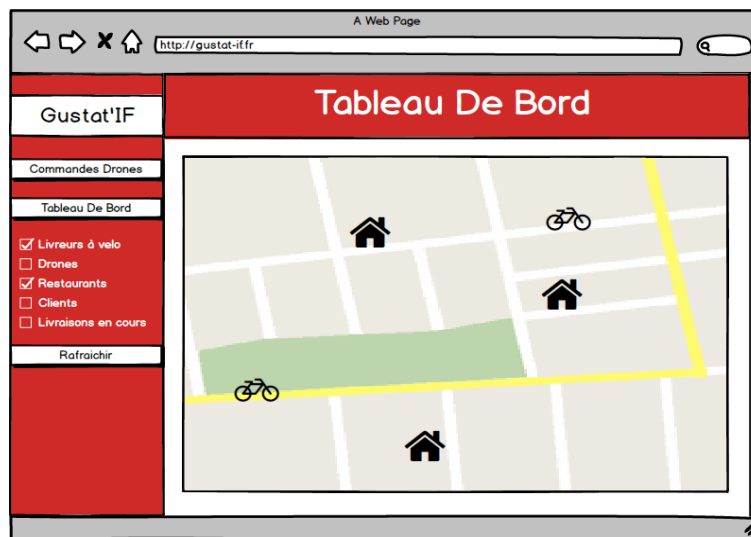
Intention / Besoin	Bouton / Contrôle	Action	Réponse
Passer une nouvelle commande	Bouton « Nouvelle commande »	Clic	Retour à la page choix du restaurant
Se déconnecter	Bouton « déconnexion »	Clic	Charge la page de d'identification / inscription

## Gestionnaire de l'application :

### ➤ Tableau de bord

Lorsque le gestionnaire ouvre l'application il doit pouvoir visualiser l'ensemble du contenu de sa base de donnée (restaurants, livreurs, clients) avec leurs emplacement sur une carte. Pour favoriser la lisibilité des donnée, on lui offre la possibilité de sélectionner quelles informations il souhaite afficher en cochant/décochant les checkbox.

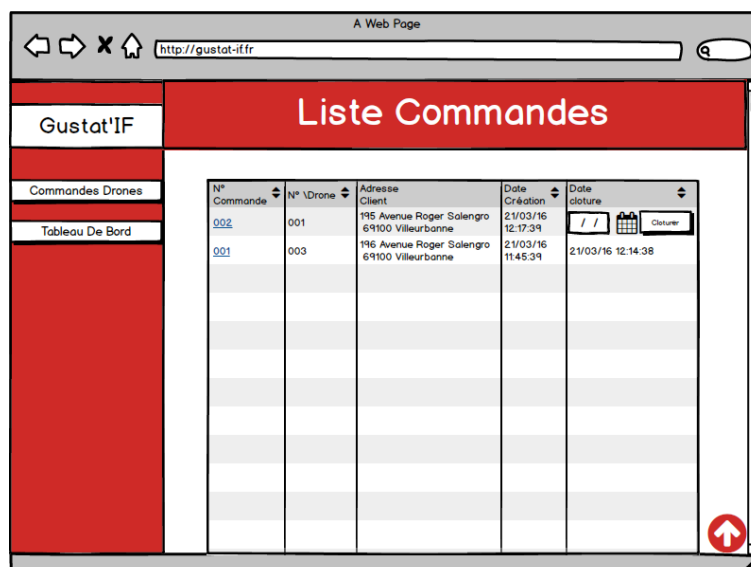




Intention / Besoin	Bouton / Contrôle	Action	Réponse
Afficher les commande des drones	Bouton « commandes drones »	Clic	Affiche les commandes des drones.
Choisir les emplacements à afficher	Check boxes	Cocher/ Décocher	
Afficher la nouvelle sélection	Bouton « Rafraichir »	Clic	Affiche sur la carte les nouvelles informations cochées

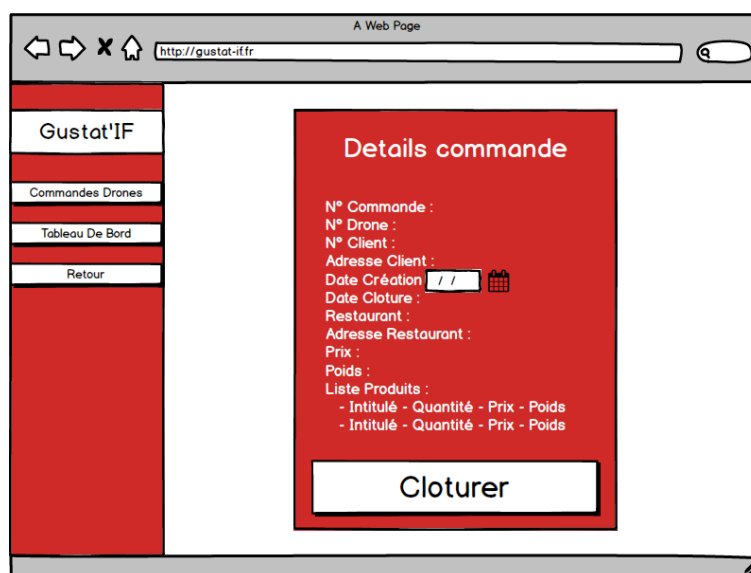
### ➤ Liste des commandes des drones

Cette page permet au gestionnaire de visualiser l'ensemble des commandes (effectuées ou en cours de traitement) par les drones. Le gestionnaire peut clôturer des commandes en sélectionnant une date et en cliquant sur clôturer. Il peut aussi afficher les détails d'une commande en cliquant sur un numéro de commande.



Intention / Besoin	Bouton / Contrôle	Action	Réponse
Afficher le tableau de bord	Bouton « Tableau de bord »	Clic	Revient sur la page précédente.
Cloturer une commande	Champ date	Remplir le champ	
	Bouton « cloturer »	Clic	Le livreur devient disponible pour recevoir une nouvelle commande. Rafraîchit la page.
Afficher détails d'une commande	Lien sur le numéro de la commande	Clic	Chargement de la page détails commande correspondant

### ➤ Afficher les détails des commandes en cours



Cette page permet de visualiser les détails d'une commande. Le contrôle permettant le choix de la date et le bouton « clôturer » ne sont présent que si la commande n'est pas clôturée.

Intention / Besoin	Bouton / Contrôle	Action	Réponse
Renseigner date	Champ date	Remplir	Met à jour la date de fin dans la BD
Revenir au tableau de bord	Bouton « retour »	Clic	Reviens au tableau de bord
Cloturer la commande	Bouton « cloturer »	Clic	Le livreur devient à nouveau disponible pour une nouvelle livraison

## Livreur à vélo

### ➤ Connexion

Lors de l'ouverture du programme, le livreur a accès à une page d'identification. Il peut s'identifier avec son mail et son mot de passe.

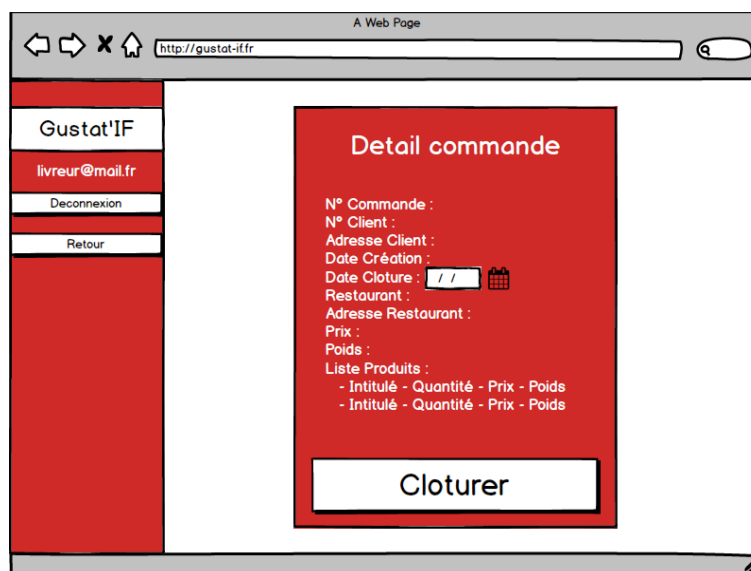
Intention / Besoin	Bouton / Contrôle	Action	Réponse
S'identifier	Champs texte	Remplir tous les champs	Aucune
	Bouton « Connexion »	Clic	Charge la page suivante

### ➤ Liste de commandes du livreur

N° Commande	N° Livreur	Adresse Client	Date Création	Date clôture
002	001	195 Avenue Roger Salengro 69100 Villeurbanne	21/03/16 12:17:39	
001	003	196 Avenue Roger Salengro 69100 Villeurbanne	21/03/16 11:45:39	21/03/16 12:14:38

Intention / Besoin	Bouton / Contrôle	Action	Réponse
Cloturer une commande	Champ date	Remplir le champ	
	Bouton « cloturer »	Clic	Le livreur deviens diponible pour recevoir une nouvelle commande. Rafraîchit la page.
Afficher détails d'une commande	Lien sur le numéro de la commande	Clic	Chargement de la page détails commande correspondant

### ➤ Détails d'une commande



Intention / Besoin	Bouton / Contrôle	Action	Réponse
Revenir à la liste des commandes	Bouton « retour »	clic	Reviens à la liste des commandes
Cloturer une commande	Champ date	Remplir le champ	
	Bouton « cloturer »	Clic	Le livreur deviens diponible pour recevoir une nouvelle commande. Rafraîchit la page.
Se déconnecter	Bouton « déconnexion »	clic	Quitte l'application