Page de garde

Table des matières

[Annexe 9 – 1 B 8](#_Toc196823060)

[Annexe 10 – B 12](#_Toc196823061)

[1ClickAllEat 14](#_Toc196823062)

[Cahier des charges 14](#_Toc196823063)

[Diagramme de Gantt : 18](#_Toc196823064)

[1ClickAllEat 19](#_Toc196823065)

[UML 19](#_Toc196823066)

[1. Diagramme de contexte 19](#_Toc196823067)

[2. Diagramme de packages 19](#_Toc196823068)

[19](#_Toc196823069)

[3. Diagramme Use\_Case 20](#_Toc196823070)

[4. Cartographie technique 21](#_Toc196823071)

[5. Diagramme de classe 21](#_Toc196823072)

[1ClickAllEat 22](#_Toc196823073)

[Merise – Modélisation BDD 22](#_Toc196823074)

[1. Dictionnaire de données 22](#_Toc196823075)

[2. MCD (Modelé Conceptuel de données) 24](#_Toc196823076)

[24](#_Toc196823077)

[3. MLD (Modèle Logique de données) 24](#_Toc196823078)

[4. MPD (Modèle Physique de données) 25](#_Toc196823079)

[25](#_Toc196823080)

[1ClickAllEat 26](#_Toc196823081)

[Extraits de code commentés 26](#_Toc196823082)

[1. Réservation d’une table 26](#_Toc196823083)

[2. Affichage différencié des commandes et réservations (client vs restaurateur) 27](#_Toc196823084)

[3. Passage d’une commande (création d’une commande à partir du panier) 28](#_Toc196823085)

[4. Annulation d’une réservation 29](#_Toc196823086)

[5. Ajout d’un plat au panier (extrait simplifié) 29](#_Toc196823087)

[1ClickAllEat 31](#_Toc196823088)

[Plan de test qualité 31](#_Toc196823089)

[Objectif 31](#_Toc196823090)

[Fonctionnalités MVP 31](#_Toc196823091)

[Fonctionnalités MLP 31](#_Toc196823092)

[Contraintes techniques 31](#_Toc196823093)

[Méthodologie 32](#_Toc196823094)

[Tableau du plan de test 32](#_Toc196823095)

[Exécution, reporting et validation 33](#_Toc196823096)

[Conclusion 33](#_Toc196823097)

[1ClickAllEat 34](#_Toc196823098)

[Guide utilisateur 34](#_Toc196823099)

[Section 1: Interface Restaurateur 34](#_Toc196823100)

[Dashboard principal 34](#_Toc196823101)

[Gestion des tables 34](#_Toc196823102)

[Gestion du menu 34](#_Toc196823103)

[Gestion des commandes 35](#_Toc196823104)

[Paramètres du restaurant 35](#_Toc196823105)

[Section 2: Interface Client 35](#_Toc196823106)

[Parcourir les restaurants 35](#_Toc196823107)

[Commander en ligne 36](#_Toc196823108)

[Suivi des commandes et réservations 36](#_Toc196823109)

[Profil utilisateur 37](#_Toc196823110)

[Section 1: Interface Restaurateur 38](#_Toc196823111)

[1. Dashboard principal 38](#_Toc196823112)

[2. Gestion des tables 38](#_Toc196823113)

[3. Gestion du menu 39](#_Toc196823114)

[4. Gestion des commandes 39](#_Toc196823115)

[5. Statistiques et performances 40](#_Toc196823116)

[6. Paramètres du restaurant 40](#_Toc196823117)

[Section 2: Interface Client 41](#_Toc196823118)

[1. Parcourir les restaurants 41](#_Toc196823119)

[2. Réserver une table 41](#_Toc196823120)

[3. Commander en ligne 42](#_Toc196823121)

[4. Suivi des commandes et réservations 42](#_Toc196823122)

[5. Profil utilisateur 42](#_Toc196823123)

[LDAP – Telora 44](#_Toc196823124)

[Cahier des charges détaillé 44](#_Toc196823125)

[1. Contexte et Présentation générale 44](#_Toc196823126)

[2. Objectifs principaux 44](#_Toc196823127)

[3. Public cible 44](#_Toc196823128)

[4. Expression des besoins et fonctionnalités 44](#_Toc196823129)

[4.2. Gestion des entités 45](#_Toc196823130)

[4.3. Sécurité et navigation 45](#_Toc196823131)

[4.4. Interface utilisateur 45](#_Toc196823132)

[4.5. Points d’attention 46](#_Toc196823133)

[5. Contraintes techniques 46](#_Toc196823134)

[6. Architecture technique (modules/dossiers) 46](#_Toc196823135)

[7. Livrables attendus 47](#_Toc196823136)

[8. Déploiement et maintenance 47](#_Toc196823137)

[9. Sécurité et RGPD 48](#_Toc196823138)

[10. Tests et validation 48](#_Toc196823139)

[Diagramme de gantt 49](#_Toc196823140)

[LDAP - Telora 50](#_Toc196823141)

[UML 50](#_Toc196823142)

[1. Diagramme de contexte 50](#_Toc196823143)

[2. Diagramme de package 50](#_Toc196823144)

[3. Diagramme use\_case 51](#_Toc196823145)

[4. Cartographie technique 52](#_Toc196823146)

[5. Diagramme de classe 52](#_Toc196823147)

[LDAP – Telora 53](#_Toc196823148)

[Merise – Modélisation BDD 53](#_Toc196823149)

[1. Dictionnaire de données 53](#_Toc196823150)

[2. MCD (Modelé conceptuel de données) 54](#_Toc196823151)

[54](#_Toc196823152)

[3. MLD (Modèle logique de données) 54](#_Toc196823153)

[4. MPD (Modèle physique de données) 55](#_Toc196823154)

[1ClickAllEat 56](#_Toc196823155)

[Extraits de code commentés 56](#_Toc196823156)

[1. Authentification et gestion des rôles 56](#_Toc196823157)

[56](#_Toc196823158)

[Explication détaillée : 56](#_Toc196823159)

[session\_start() 56](#_Toc196823160)

[démarre la session PHP, permettant d’accéder aux variables de session pour suivre l’état de connexion de l’utilisateur. 56](#_Toc196823161)

[La condition 56](#_Toc196823162)

[if (!isset($\_SESSION['role'])) 56](#_Toc196823163)

[vérifie si l’utilisateur est authentifié (un rôle doit être défini en session après connexion). 56](#_Toc196823164)

[Si l’utilisateur n’est pas connecté, il est redirigé vers la page de login, ce qui protège toutes les pages sensibles contre l’accès non autorisé. 56](#_Toc196823165)

[Les variables de session comme 56](#_Toc196823166)

[role 56](#_Toc196823167)

[et 56](#_Toc196823168)

[partner\_id 57](#_Toc196823169)

[sont utilisées pour adapter dynamiquement l’interface et les droits d’accès. Par exemple, un partenaire ne pourra voir que ses propres clients, un client que ses propres contacts, etc. 57](#_Toc196823170)

[Ce mécanisme centralise la gestion des droits et garantit la séparation stricte des accès selon le rôle, comme exigé dans le cahier des charges. 57](#_Toc196823171)

[2. Récupération et affichage paginé des contacts 57](#_Toc196823172)

[**Explication détaillée :** 57](#_Toc196823173)

[Cette séquence gère la pagination pour l’annuaire des contacts d’un client. 57](#_Toc196823174)

[Elle récupère la page courante et le nombre d’éléments à afficher par page depuis l’URL (GET), avec des valeurs par défaut sécurisées. 57](#_Toc196823175)

[**countAnnuaireByClient($clientsId)** compte le nombre total de contacts pour le client, ce qui permet de calculer le nombre total de pages. 57](#_Toc196823176)

[L’offset ( 57](#_Toc196823177)

[$offset 57](#_Toc196823178)

[) détermine à partir de quel contact commencer l’affichage pour la page courante. 57](#_Toc196823179)

[**getAnnuaireByClientPaginated($clientsId, $offset, $perPage)** récupère uniquement les contacts à afficher pour cette page, optimisant les performances même pour de gros volumes de données. 57](#_Toc196823180)

[Ce mécanisme améliore l’ergonomie et la rapidité de navigation dans l’annuaire. 58](#_Toc196823181)

[3. Modification d’un contact 58](#_Toc196823182)

[**Explication détaillée :** 58](#_Toc196823183)

[Cette méthode met à jour toutes les informations d’un contact dans la base de données. 58](#_Toc196823184)

[Elle utilise une requête SQL préparée, ce qui protège contre les injections SQL et garantit la sécurité des données. 58](#_Toc196823185)

[Chaque champ du contact (prénom, nom, email, etc.) est lié à un paramètre de la requête pour éviter toute altération malveillante. 58](#_Toc196823186)

[En cas d’erreur lors de la mise à jour (par exemple, si la base est inaccessible), une exception claire est levée, facilitant le debug et la robustesse de l’application. 58](#_Toc196823187)

[Cette méthode est appelée lors de la modification d’un contact depuis l’interface utilisateur. 59](#_Toc196823188)

[4. Import de contacts depuis un fichier CSV 59](#_Toc196823189)

[**Explication détaillée :** 59](#_Toc196823190)

[Cette fonction permet d’importer en masse des contacts pour un client à partir d’un fichier CSV. 59](#_Toc196823191)

[Elle lit chaque ligne du fichier, nettoie les données (suppression des caractères spéciaux, normalisation des champs) et insère chaque contact dans la base. 59](#_Toc196823192)

[Elle gère les erreurs de format (colonnes manquantes, emails invalides, etc.) et évite les doublons, assurant la qualité des données importées. 59](#_Toc196823193)

[Ce mécanisme facilite la migration de données ou l’intégration de nouveaux clients disposant déjà d’un annuaire. 59](#_Toc196823194)

[5. Suppression d’un client 60](#_Toc196823195)

[**Explication détaillée :** 60](#_Toc196823196)

[Cette méthode supprime un client de la base de données à partir de son identifiant. 60](#_Toc196823197)

[L’utilisation d’une requête préparée garantit la sécurité de l’opération. 60](#_Toc196823198)

[Elle permet de gérer la suppression d’un client depuis l’interface d’administration ou de gestion partenaire, en respectant les droits d’accès. 60](#_Toc196823199)

[6. Connexion utilisateur 61](#_Toc196823200)

[**Explication détaillée :** 61](#_Toc196823201)

[Cette méthode gère la connexion d’un utilisateur à l’application. 61](#_Toc196823202)

[Elle recherche l’utilisateur dans la base à partir du login, puis vérifie le mot de passe avec 61](#_Toc196823203)

[password\_verify 61](#_Toc196823204)

[, qui prend en charge le hachage sécurisé. 61](#_Toc196823205)

[Si la connexion réussit, elle initialise la session et stocke l’identifiant et le rôle de l’utilisateur, permettant de personnaliser l’accès à l’application. 61](#_Toc196823206)

[La fonction 62](#_Toc196823207)

[redirectBasedOnRole 62](#_Toc196823208)

[permet de rediriger l’utilisateur vers l’espace qui lui correspond (admin, partenaire, client). 62](#_Toc196823209)

[En cas d’échec, la méthode retourne 62](#_Toc196823210)

[false 62](#_Toc196823211)

[ce qui permet d’afficher un message d’erreur à l’utilisateur. 62](#_Toc196823212)

[LDAP – Telora 63](#_Toc196823213)

[Plan de test qualité 63](#_Toc196823214)

[1. Objectif 63](#_Toc196823215)

[2. Périmètre du test 63](#_Toc196823216)

[3. Méthodologie 63](#_Toc196823217)

[4. Tableau du plan de test 64](#_Toc196823218)

[5. Exécution, reporting et validation 64](#_Toc196823219)

[6. Conclusion 64](#_Toc196823220)

[Telora - LDAP 65](#_Toc196823221)

[Guide Utilisateur 65](#_Toc196823222)

[Fonctionnalités principales : 65](#_Toc196823223)

[Écran 2 : Ajout d'un Nouveau Partenaire 66](#_Toc196823224)

[Écran 3 : Page d'Accueil Telora avec Section Partenaires 67](#_Toc196823225)

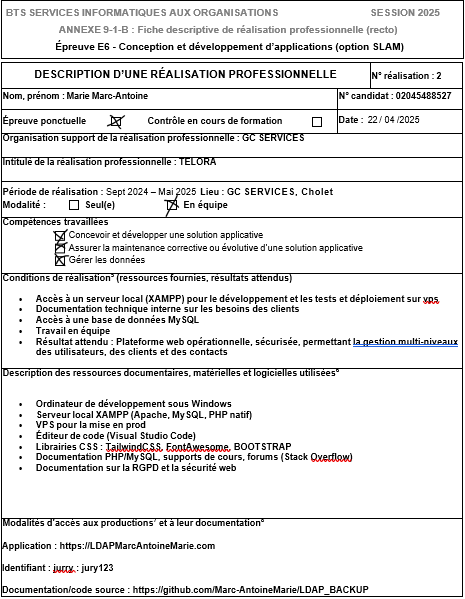
[Écran 1 : Liste des Clients d'un Partenaire 68](#_Toc196823226)

[Écran 2 : Modifier un Client 69](#_Toc196823227)

[Écran 3 : Ajouter un Nouveau Client 70](#_Toc196823228)

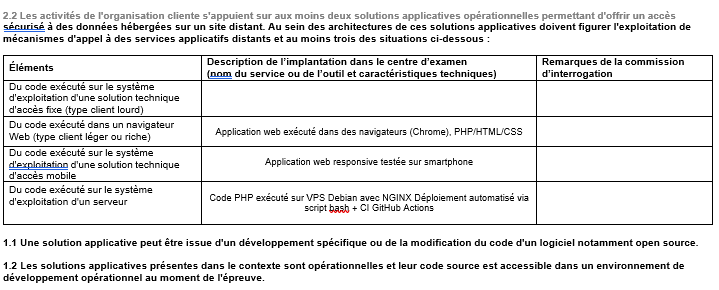
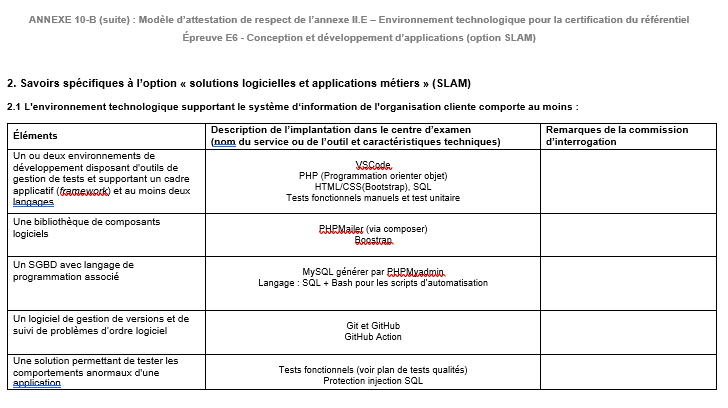
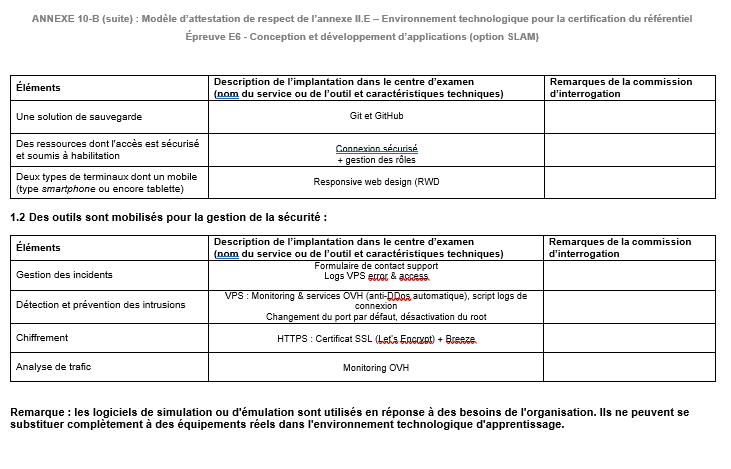
# Annexe 9 – 1 B







# Annexe 10 – B



# 1ClickAllEat

## Cahier des charges

**1. Contexte**

La startup 1ClickAllEat nous mandate pour développer une application de réservation de restaurant au préalable. Cette solution vise à simplifier et rendre plus accessibles les restaurants traditionnels qui sont concurrencés par les fast-foods.

**2. Objectif Principal**

Proposer une solution optimisée pour plusieurs restaurants qui proposent des menus différents. L'application permettra d'optimiser les échanges entre les différents acteurs du restaurant (administration, service, cuisine, clients).

**3. Public Cible**

Restaurateurs et leur personnel

**4. Expression des Besoins**

L'application 1ClickAllEat doit offrir une gestion complète et optimisée des restaurants, permettant de suivre en temps réel les menus, les commandes et les réservations. L'ensemble des opérations sera parfaitement synchronisé avec les horaires d'ouverture, assurant une expérience fluide et sur-mesure pour chaque client.

La plateforme fournira aux restaurateurs les outils nécessaires à l'ajustement des offres, à la gestion des réservations et au suivi des commandes, accessibles en ligne pour un pilotage efficace des établissements à tout moment.

**5. Fonctionnalités**

**5.1. Fonctionnalités MVP (Minimum Viable Product)**

* **Gestion des restaurants**
  + Création et paramétrage des profils de restaurants
  + Configuration des horaires d'ouverture
  + Gestion des tables et capacités
* **Authentification des utilisateurs**
  + Système d'inscription/connexion sécurisé
  + Récupération de mot de passe
  + Différents niveaux d'accès selon les rôles
* **Accès spécifique restaurant**
  + Interface dédiée aux restaurateurs
  + **Tableau de bord de gestion**
  + Statistiques de base sur l'activité
* **Accès spécifique client**
  + Interface utilisateur intuitive
  + Recherche et filtrage des restaurants
  + Historique des commandes/réservations
* **Gestion de la carte/des menus**
  + Classification des articles au menu par catégorie
  + Ajout/modification/suppression d'articles
  + Gestion des prix et disponibilités
* **Gestion des commandes/réservations**
  + Création et suivi des réservations
  + Gestion des statuts des commandes
  + Notifications aux clients et restaurateurs

**5.2. Fonctionnalités MLP (Minimum Lovable Product)**

* **Génération de QR code**
  + Lien d'accès à la commande pour chaque carte de restaurant
  + QR code dynamique et personnalisable
* **Responsive**
  + Adaptation à tous les formats d'écran
  + Expérience utilisateur optimisée sur mobile, tablette et desktop
* **Paramétrage de la charte graphique**
  + Personnalisation des couleurs et logo du restaurant
  + Adaptation visuelle de la carte selon l'identité du restaurant
* **Paiement en ligne via API Stripe**
  + Intégration complète avec le package Laravel Cashier
  + Gestion sécurisée des transactions
  + Suivi des paiements et remboursements

**6. Contraintes Techniques**

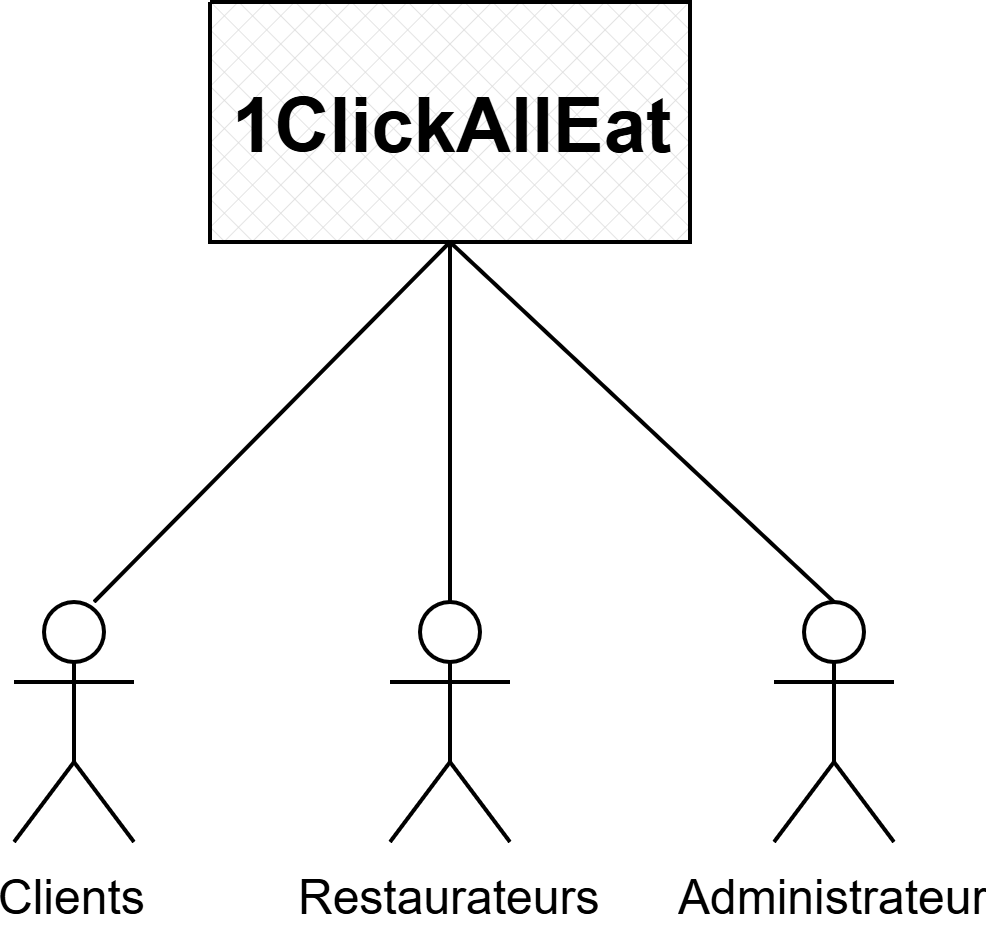
* **Framework de développement**
  + Laravel (Back-end)
  + Interface utilisateur moderne et intuitive
* **Déploiement**
  + Sur un nom de domaine dédié
  + Hébergement sécurisé et performant
* **Intégration continue**
  + Automatisation des migrations de base de données
  + Gestion du vidage des caches de Laravel
  + Optimisation des fichiers pour l'environnement de production
* **Template**
  + Design responsive et moderne
  + Adaptable à l'identité visuelle des différents restaurants
* **Sécurité**
  + Certificat SSL (HTTPS) pour sécuriser les communications
  + Protection des données personnelles
* **Tests unitaires**
  + Mise en place de tests automatisés
  + Code qui teste le code

# Diagramme de Gantt :

# 1ClickAllEat

## UML

### 1. Diagramme de contexte



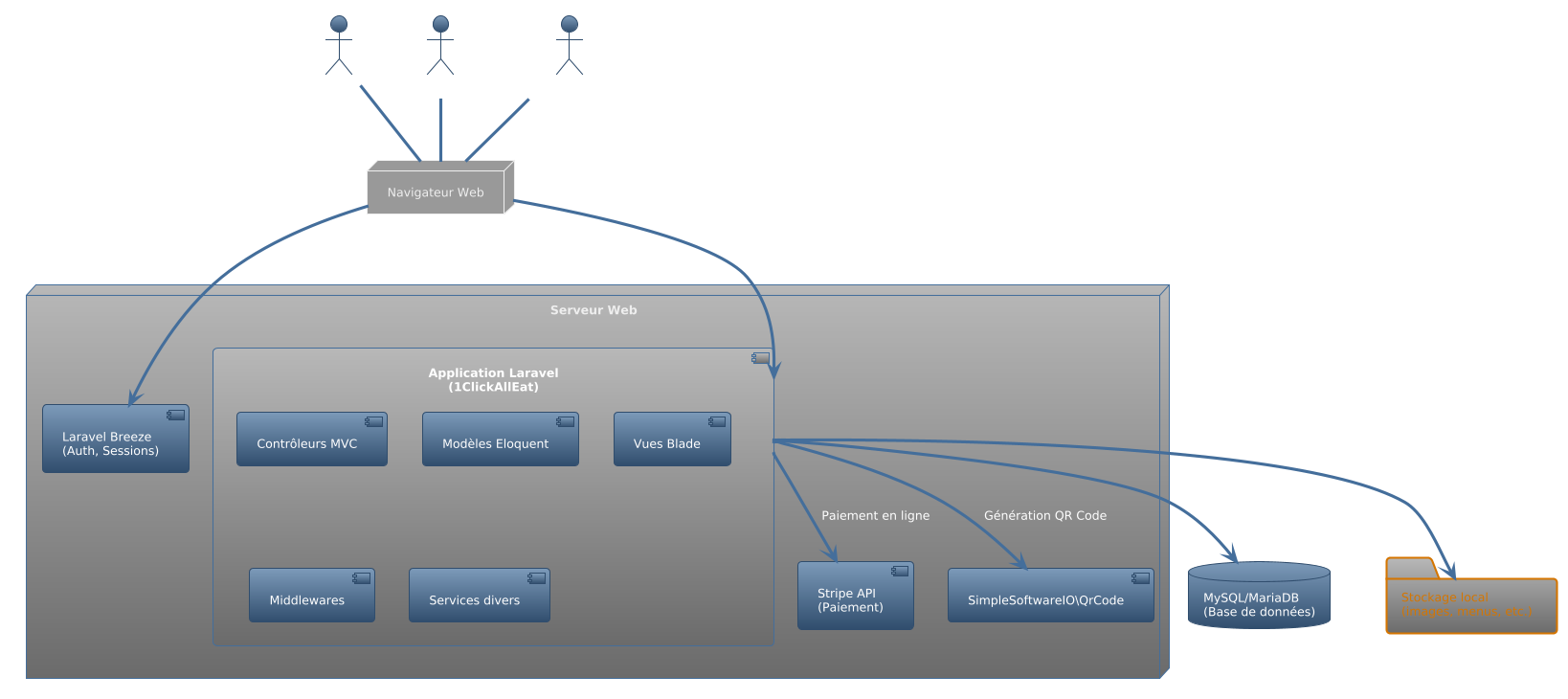
### 2. Diagramme de packages

## 

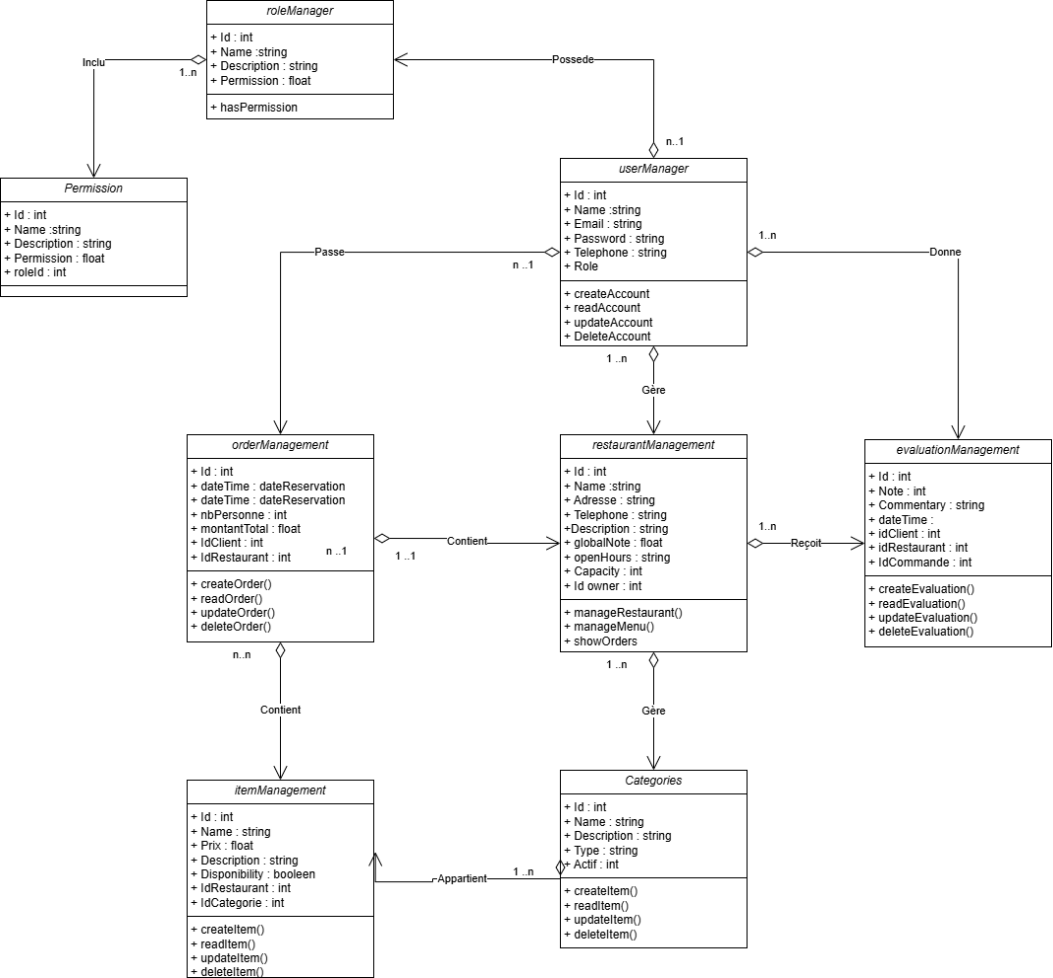
## 

### 3. Diagramme Use\_Case

### 4. Cartographie technique



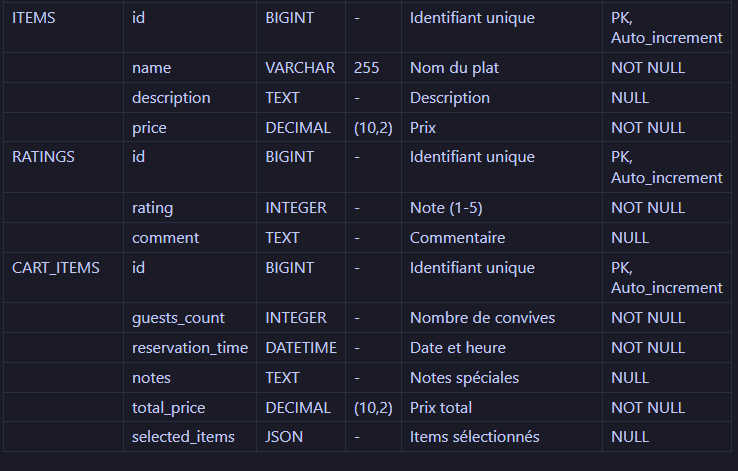
### 5. Diagramme de classe



# 1ClickAllEat

## Merise – Modélisation BDD

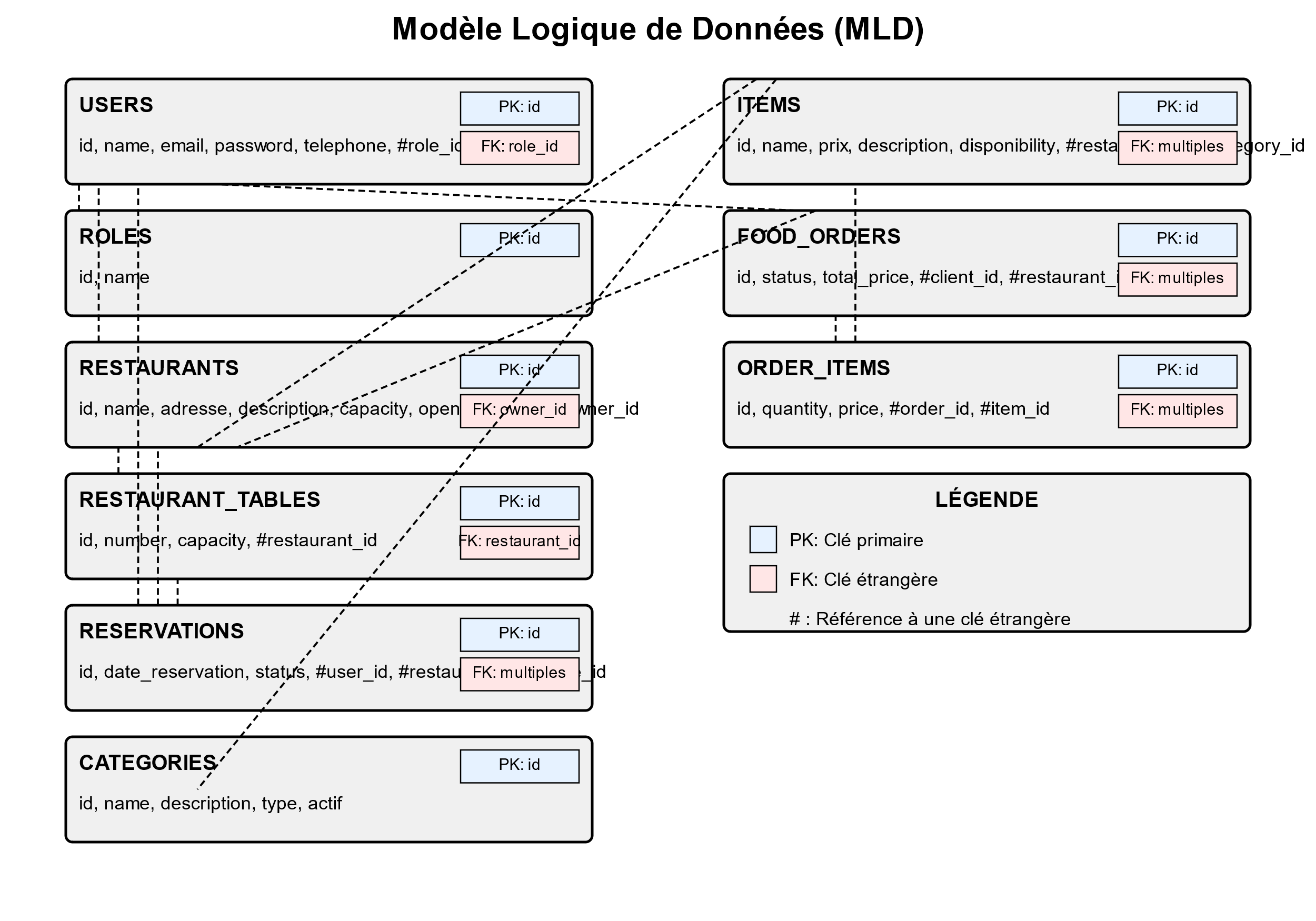
### 1. Dictionnaire de données



### 2. MCD (Modelé Conceptuel de données)

### 

### 3. MLD (Modèle Logique de données)



### 4. MPD (Modèle Physique de données)

### 

# 1ClickAllEat

## Extraits de code commentés

### 1. Réservation d’une table

**Explication détaillée**  
Cette méthode permet à un utilisateur connecté de réserver une table :

* Elle valide que la table existe et que la date/heure sont correctes.
* Elle vérifie que l’utilisateur est bien connecté.
* Elle contrôle que la table n’est pas déjà réservée à ce créneau.
* Si tout est OK, elle crée la réservation et génère automatiquement une commande vide associée, facilitant le workflow côté client et restaurateur.

### 2. Affichage différencié des commandes et réservations (client vs restaurateur)

**Explication détaillée**  
Cette méthode affiche les commandes et réservations dans l’espace utilisateur :

* Si l’utilisateur est un **client** : il voit ses propres commandes et réservations.
* Si c’est un **restaurateur** : il voit toutes les commandes passées dans son restaurant.
* Cela permet une expérience personnalisée et sécurisée selon le rôle, en limitant l’accès aux bonnes informations.

### 3. Passage d’une commande (création d’une commande à partir du panier)

**Explication détaillée**

* Cette méthode gère la validation du panier et la création d’une commande.
* Elle vérifie que le panier contient bien des plats, puis crée la commande avec le total calculé.
* Chaque plat sélectionné est ajouté à la commande via la relation items().
* Cela garantit la cohérence des données et permet un suivi précis des commandes côté cuisine.

### 4. Annulation d’une réservation

**Explication détaillée**

* Cette méthode permet à un utilisateur d’annuler sa propre réservation.
* Elle vérifie l’autorisation (propriétaire uniquement).
* Elle change le statut en “cancelled” et sauvegarde la modification.
* Cela protège la logique métier et évite les suppressions accidentelles ou malveillantes.

### 5. Ajout d’un plat au panier (extrait simplifié)

php

CopyInsert

// Extrait de CartController.php

public function add(Request $request, $itemId)

{

$item = Item::findOrFail($itemId);

$cart = session()->get('cart', []);

$cart[$itemId] = [

'name' => $item->name,

'quantity' => ($cart[$itemId]['quantity'] ?? 0) + 1,

'price' => $item->effective\_price,

];

session(['cart' => $cart]);

return back()->with('success', 'Plat ajouté au panier.');

}

**Explication détaillée**

* Cette méthode ajoute un plat au panier stocké en session.
* Si le plat existe déjà, on incrémente la quantité.
* Le panier est persistant tant que la session de l’utilisateur reste active.
* Cela permet une expérience utilisateur fluide et rapide, sans base de données pour le panier temporaire.

# 1ClickAllEat

## Plan de test qualité

### Objectif

Fournir un cadre structuré pour valider que l’application 1ClickAllEat répond à toutes les exigences fonctionnelles (MVP, MLP), techniques et de qualité : gestion des restaurants, authentification, accès différencié, gestion des menus/commandes, responsive, personnalisation graphique, sécurité, CI/CD, etc.

Périmètre du test

### Fonctionnalités MVP

* Gestion des restaurants (création, modification, suppression)
* Authentification et gestion des utilisateurs (clients, restaurateurs)
* Accès différencié (dashboard restaurateur, espace client)
* Gestion des menus/cartes (catégorisation, affichage)
* Gestion des commandes (création, suivi, annulation)

### Fonctionnalités MLP

* Génération et affichage du QR Code pour accès rapide
* Design responsive (mobile, tablette, desktop)
* Personnalisation graphique (charte, couleurs, logos)
* Paiement en ligne (API Stripe)

### Contraintes techniques

* Laravel (PHP 8+)
* Déploiement HTTPS (certificat SSL)
* Intégration continue (migrations, cache, optimisation)
* Tests unitaires/fonctionnels automatisés

### Méthodologie

* Tests manuels et automatisés : PHPUnit,
* Tests fonctionnels : Vérification de chaque exigence du cahier des charges
* Tests d’intégration : Vérification de la communication entre modules
* Tests de performance : Simulation de charge, temps de réponse
* Tests de sécurité : SSL, authentification, gestion des rôles
* Tests de compatibilité/responsive : Desktop, tablette, mobile, principaux navigateurs
* Environnement de préproduction : Identique à la production

### Tableau du plan de test

| Périmètre testé | Date | Description du test | Résultat attendu | Résultat obtenu | Réussite/Échec | Testé par |
| --- | --- | --- | --- | --- | --- | --- |
| Authentification utilisateur | 2025-04-10 | Connexion/déconnexion (client, restaurateur, admin) | Accès sécurisé, redirection correcte, session isolée | À renseigner | À renseigner | QA |
| Gestion des restaurants | 2025-04-11 | CRUD restaurant via dashboard restaurateur | Opérations sans erreur, affichage immédiat des changements | À renseigner | À renseigner | QA |
| Accès restaurateur | 2025-04-12 | Dashboard, gestion menus, commandes, réservations | Accès réservé, outils visibles uniquement pour restaurateur | À renseigner | À renseigner | QA |
| Accès client | 2025-04-12 | Consultation menus, réservation, commande | Accès restreint, affichage personnalisé selon le restaurant | À renseigner | À renseigner | QA |
| Gestion menus/cartes | 2025-04-13 | CRUD catégories/articles, affichage dynamique | Modifications visibles, catégorisation correcte | À renseigner | À renseigner | QA |
| Gestion des commandes | 2025-04-14 | Passer/annuler commande, suivi en temps réel | Commande enregistrée, notification cuisine, historique mis à jour | À renseigner | À renseigner | QA |
| Génération QR Code | 2025-04-15 | Génération, scan, redirection vers menu/commande | QR code lisible, redirection correcte | À renseigner | À renseigner | QA |
| Responsive design | 2025-04-16 | Tests sur desktop, tablette, mobile, Chrome, Firefox, Safari | Interface cohérente, aucune perte de fonctionnalité | À renseigner | À renseigner | QA |
| Personnalisation graphique | 2025-04-17 | Modification couleurs, logos, polices via interface | Rendu conforme à la charte définie | À renseigner | À renseigner | QA |
| Paiement en ligne (Stripe) | 2025-04-18 | Transaction test, validation, génération reçu | Paiement validé, reçu généré, sécurité respectée | À renseigner | À renseigner | QA |
| CI/CD et déploiement | 2025-04-19 | Migrations, cache, optimisation lors d’un déploiement | Scripts CI déclenchés, déploiement sans interruption, rollback possible | À renseigner | À renseigner | DevOps |
| Sécurité SSL/API | 2025-04-20 | Vérification HTTPS, sécurité Stripe, rôles | Aucune faille détectée, accès sécurisé, logs d’accès complets | À renseigner | À renseigner | Sécurité |

### Exécution, reporting et validation

* Les anomalies sont tracées dans l’outil de suivi (GitHub Issues…).
* Un rapport de test est généré à chaque fin de cycle.
* La mise en production n’est validée que si tous les tests critiques sont passés et les anomalies bloquantes corrigées.

### Conclusion

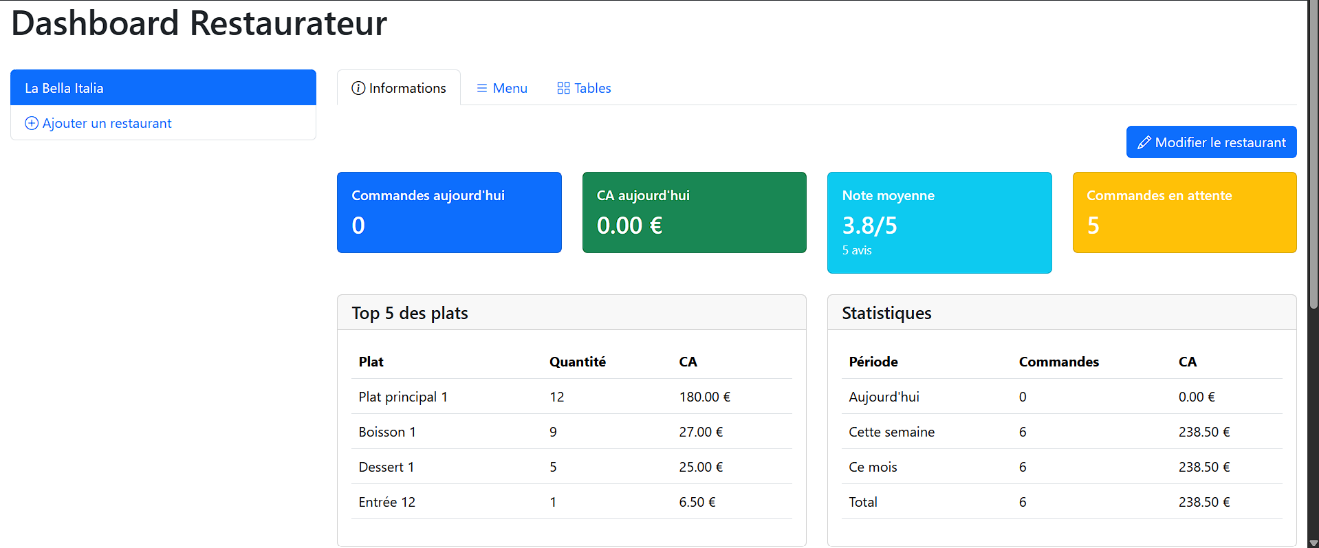
Ce plan de test garantit la conformité de l’application 1ClickAllEat avec les attentes fonctionnelles et techniques du projet. Il sera mis à jour à chaque évolution majeure du produit ou retour utilisateur.

# 1ClickAllEat

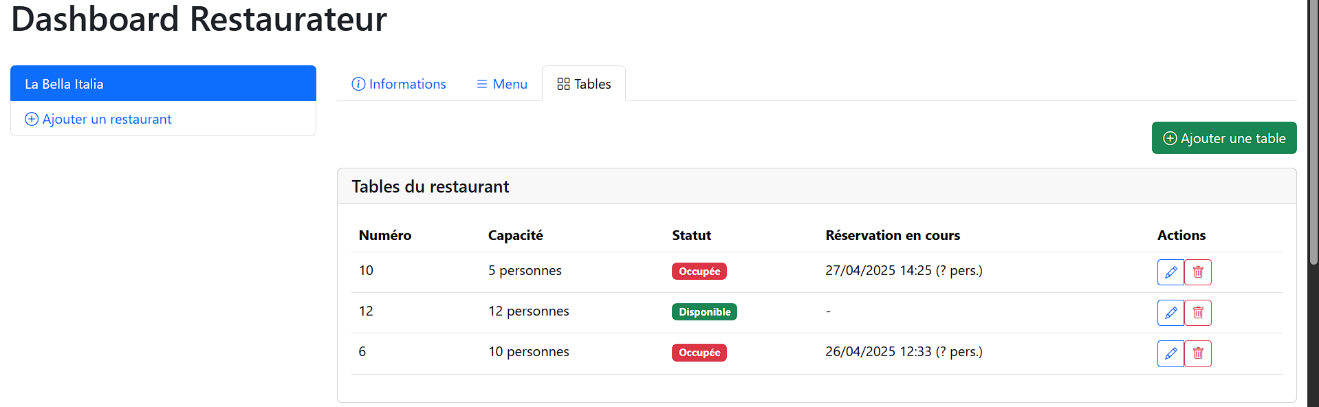
## Guide utilisateur

### Section 1: Interface Restaurateur

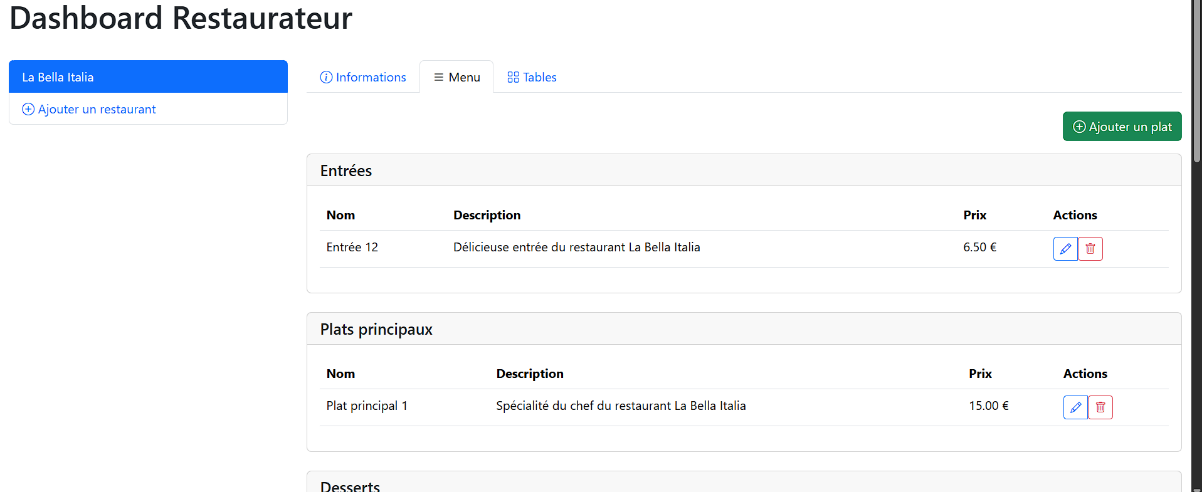
### Dashboard principal



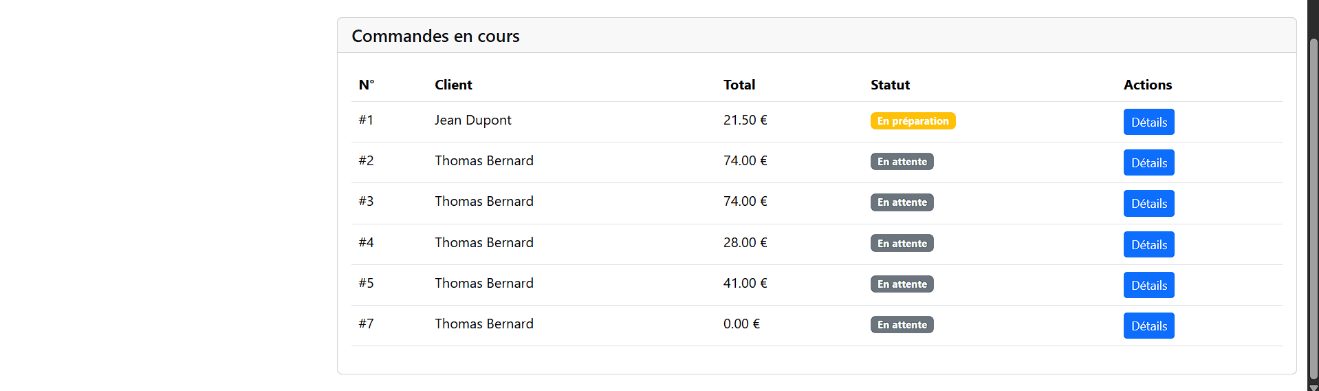
### Gestion des tables



### Gestion du menu



### Gestion des commandes

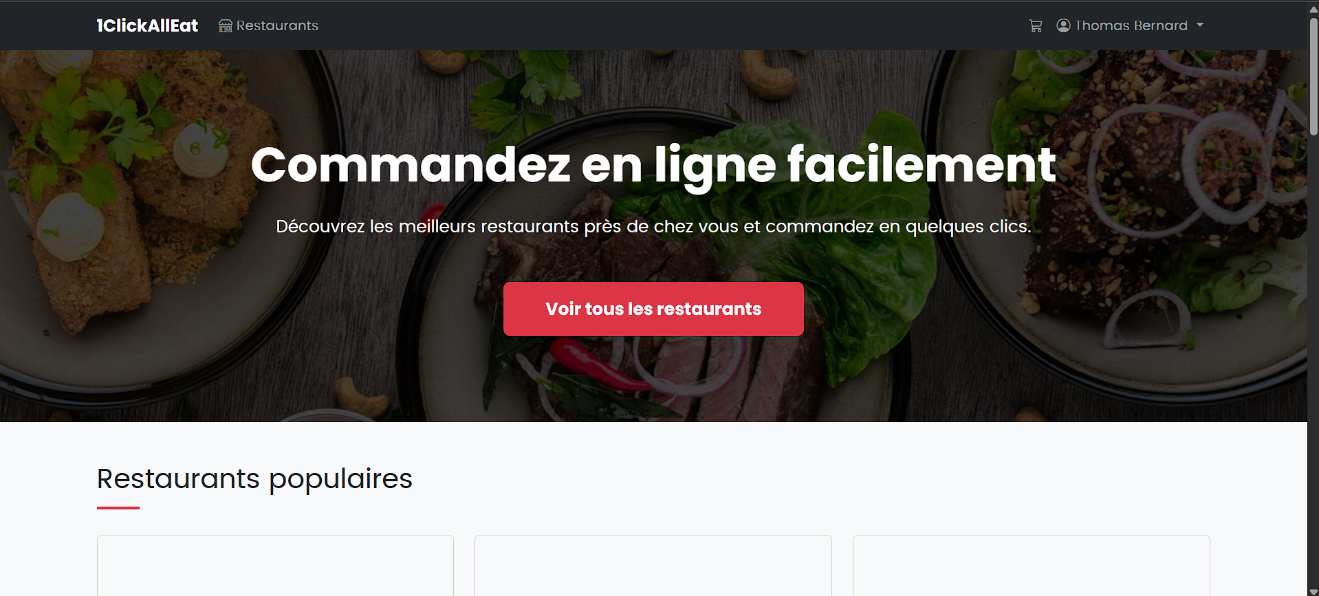


### Paramètres du restaurant



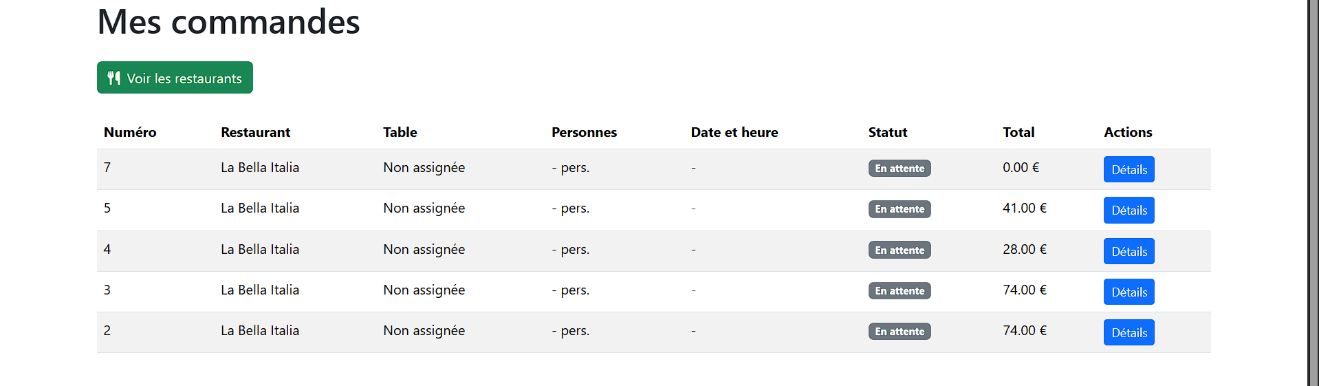
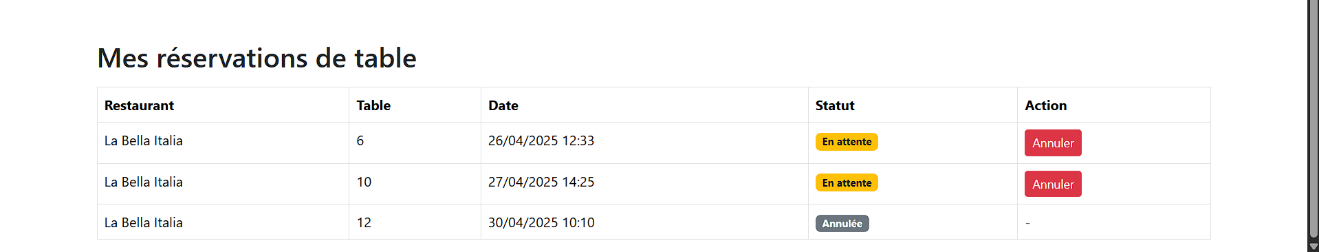
### Section 2: Interface Client

### Parcourir les restaurants

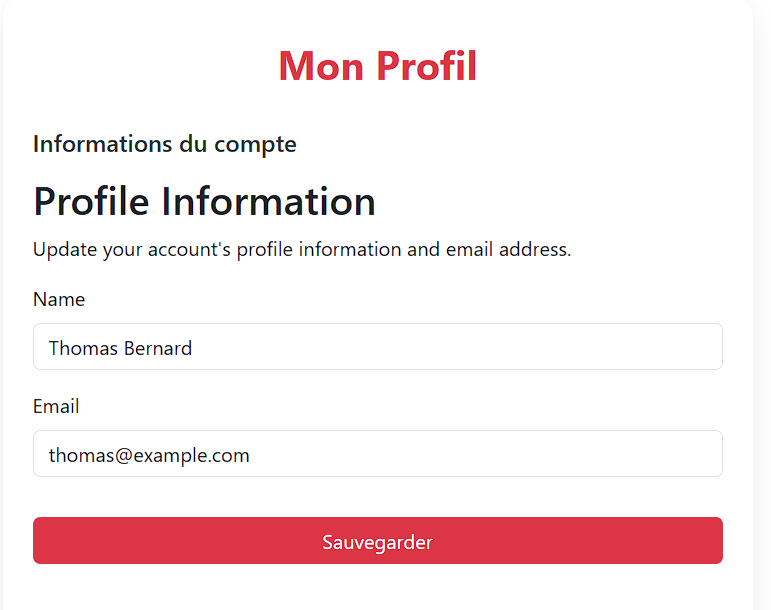


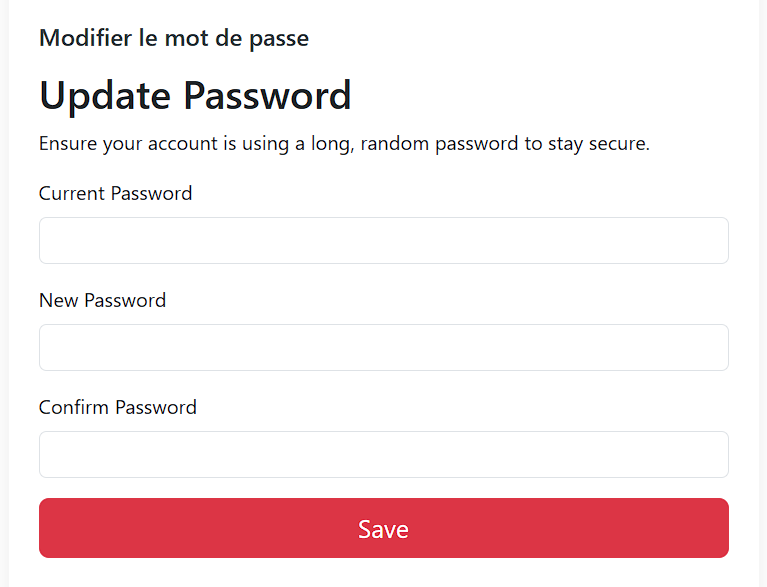
### Commander en ligne

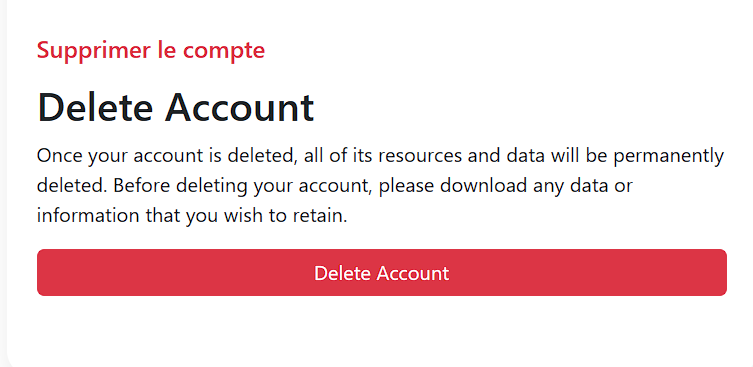
### Suivi des commandes et réservations



### Profil utilisateur







### Section 1: Interface Restaurateur

### 1. Dashboard principal

Le Dashboard Restaurateur est votre centre de contrôle principal pour gérer votre établissement sur la plateforme 1ClickAllEat. Il affiche:

* Le nom de votre restaurant (ex: La Bella Italia)
* Navigation principale: Informations, Menu, Tables
* Statistiques essentielles:
  + Commandes aujourd'hui (nombre)
  + Chiffre d'affaires (CA) du jour
  + Note moyenne et nombre d'avis (ex: 3.8/5 basé sur 5 avis)
  + Commandes en attente (nombre)
* Top 5 des plats les plus vendus avec quantités et CA générés
* Statistiques de performance par période (aujourd'hui, semaine, mois, total)

Le bouton "Modifier le restaurant" en haut à droite vous permet d'accéder rapidement aux paramètres de votre établissement.

### 2. Gestion des tables

L'onglet "Tables" vous permet de visualiser et gérer l'ensemble des tables de votre restaurant:

* Liste complète avec: Numéro, Capacité, Statut (Disponible/Occupée)
* Affichage des réservations en cours pour les tables occupées
* Pour chaque table, vous pouvez:
  + Modifier les informations (icône crayon)
  + Supprimer la table (icône poubelle)
* Bouton "Ajouter une table" pour créer une nouvelle table avec:
  + Numéro de table
  + Capacité (nombre de personnes)

Cette vue vous aide à optimiser l'occupation de votre restaurant et à suivre les réservations en temps réel.

### 3. Gestion du menu

L'onglet "Menu" vous permet de créer et gérer l'ensemble de votre carte:

* Organisation par catégories: Entrées, Plats principaux, Desserts, etc.
* Pour chaque plat, vous visualisez:
  + Nom du plat
  + Description
  + Prix
  + Actions (modifier/supprimer)
* Bouton "Ajouter un plat" pour enrichir votre menu avec:
  + Nom du plat
  + Description détaillée
  + Prix en euros
  + Catégorie (à sélectionner dans le menu déroulant)

Gardez votre menu à jour pour offrir la meilleure expérience à vos clients.

### 4. Gestion des commandes

La section "Commandes en cours" vous permet de suivre toutes les commandes reçues:

* Numéro de commande
* Nom du client
* Montant total
* Statut (En préparation, En attente)
* Action "Détails" pour consulter le détail complet

Cette vue vous aide à traiter efficacement les commandes et à suivre leur progression.

### 5. Statistiques et performances

Le dashboard principal affiche des indicateurs clés de performance:

* Top 5 des plats les plus populaires avec:
  + Nom du plat
  + Quantité vendue
  + Chiffre d'affaires généré
* Récapitulatif financier par période:
  + Aujourd'hui (0 commandes, 0.00 €)
  + Cette semaine (6 commandes, 238.50 €)
  + Ce mois (6 commandes, 238.50 €)
  + Total (6 commandes, 238.50 €)

Ces données vous permettent d'analyser les tendances et d'optimiser votre offre.

### 6. Paramètres du restaurant

Le formulaire "Modifier le restaurant" vous permet de mettre à jour les informations essentielles:

* Nom du restaurant
* Adresse complète
* Description détaillée
* Horaires d'ouverture (format suggéré: Lun-Ven 9h-22h, Sam-Dim 10h-23h)
* Capacité totale (nombre de places)

Gardez ces informations à jour pour que les clients puissent trouver facilement votre établissement.

### Section 2: Interface Client

### 1. Parcourir les restaurants

La page d'accueil client présente:

* Bannière principale "Commandez en ligne facilement"
* Courte description: "Découvrez les meilleurs restaurants près de chez vous et commandez en quelques clics"
* Bouton "Voir tous les restaurants" pour accéder au catalogue complet
* Section "Restaurants populaires" présentant les établissements les plus appréciés
* Filtres de recherche avec:
  + Nom du restaurant
  + Note minimale
  + Adresse/Ville (Ex: Paris)
  + Tri (A-Z)

Cette interface permet aux clients de découvrir rapidement des restaurants qui correspondent à leurs préférences.

### 2. Réserver une table

La section "Mes réservations de table" permet aux clients de:

* Visualiser leurs réservations existantes avec:
  + Nom du restaurant
  + Numéro de table
  + Date et heure
  + Statut (En attente, Annulée)
  + Action possible (Annuler)
* Réserver de nouvelles tables dans les restaurants participants

Les réservations sont synchronisées avec le dashboard restaurateur, permettant une gestion efficace des places.

### 3. Commander en ligne

Les clients peuvent passer commande directement depuis la plateforme:

* Parcourir le menu du restaurant organisé par catégories
* Ajouter des plats à leur panier
* ²Définir la quantité pour chaque article
* Voir le total de leur commande en temps réel
* Valider et payer en ligne

### 4. Suivi des commandes et réservations

La section "Mes commandes" permet aux clients de:

* Voir l'historique complet de leurs commandes
* Consulter le statut de chaque commande (En attente, En préparation)
* Accéder aux détails complets (articles, prix, restaurant)
* Voir le montant total de chaque commande
* Les commandes sont numérotées pour faciliter le suivi

Cette transparence améliore l'expérience client et réduit les demandes d'information.

### 5. Profil utilisateur

Le menu utilisateur (accessible depuis l'avatar en haut à droite) permet d'accéder à:

* "Mon profil": informations personnelles
* "Mes commandes": historique des commandes
* "Déconnexion": quitter la session en cours

Les clients peuvent ainsi gérer facilement leur compte et accéder à leur historique.

# LDAP – Telora

## Cahier des charges détaillé

### 1. Contexte et Présentation générale

Telora est une application web de gestion de contacts, d’annuaires téléphoniques et d’utilisateurs, pensée pour un fonctionnement en marque blanche. Elle vise à permettre à différents types d’utilisateurs (Admins, Partenaires, Clients finaux) de gérer leurs propres données, via une interface moderne, responsive et sécurisée.

Le projet est développé en PHP/MySQL côté serveur, avec une interface HTML/CSS/JS enrichie par TailwindCSS et FontAwesome.

### 2. Objectifs principaux

- Centraliser la gestion des contacts et des utilisateurs pour des clients variés (partenaires, entreprises, utilisateurs finaux).

- Offrir une séparation stricte des accès selon le rôle.

- Proposer une expérience utilisateur fluide, moderne et sécurisée.

- Permettre l’import/export de contacts au format CSV, avec robustesse face aux erreurs de format.

### 3. Public cible

- Administrateurs système (super-admins)

- Partenaires (entreprises clientes de Telora)

- Clients finaux (utilisateurs des partenaires)

### 4. Expression des besoins et fonctionnalités

#### 4.1. Gestion multi-niveaux et rôles

- Admins : gestion globale des partenaires, clients, utilisateurs, et contacts.

- Partenaires : gestion de leurs propres clients finaux et de leurs contacts.

- Clients finaux : gestion de leurs propres contacts et informations.

### 4.2. Gestion des entités

- CRUD complet (création, lecture, modification, suppression) pour :

- Clients

- Utilisateurs

- Contacts de l’annuaire

- Import/export CSV des contacts (encodage UTF-8, format compatible)

- Recherche, filtrage, pagination des contacts

### 4.3. Sécurité et navigation

- Séparation stricte des accès selon le rôle (aucune fuite de données entre clients)

- Gestion du contexte de navigation via la session PHP

- Redirection automatique vers la page de connexion en cas d’accès non autorisé

- Logs d’erreur activés pour le suivi des bugs (sans fuite d’informations sensibles)

- Jamais de modification des variables de session à partir de l’URL côté client

- Vérification stricte des droits d’accès à chaque page sensible

### 4.4. Interface utilisateur

- Interface responsive, moderne, centrée, avec header harmonisé sur toutes les pages

- Utilisation de containers pour centrer et limiter la largeur du contenu (max 1200px)

- Design cohérent entre tous les formulaires et tableaux

- Utilisation de boutons modernes, feedback utilisateur (succès/erreur), modals pour les actions sensibles

- Intégration de TailwindCSS et FontAwesome

### 4.5. Points d’attention

- Maintien du contexte utilisateur lors de la navigation (pas de fuite de session)

- Cohérence visuelle et UX sur toutes les pages (header, boutons, tableaux, formulaires)

- Robustesse de la gestion des imports/exports CSV (gestion des erreurs de format)

- Tests sur différents rôles et scénarios d’accès

### 5. Contraintes techniques

- Technologies : PHP (backend), MySQL (base de données), HTML/CSS/JS (frontend), TailwindCSS, FontAwesome

- Organisation : séparation claire entre la logique métier (PHP, dossier classes/api), les vues (HTML, dossiers admin, annuaire, clientlist, utilisateurdetail…), et le style (CSS, partials)

- Sécurité : sessions PHP, contrôle d’accès, logs d’erreur, protection contre la modification des sessions côté client

- Ergonomie : centrage du contenu, largeur limitée, responsive design

- Tests : activation des logs, rédaction d’un cahier de tests fonctionnels, tests manuels/automatisés selon disponibilité

### 6. Architecture technique (modules/dossiers)

- admin/ : gestion des administrateurs et des partenaires

- annuaire/ : gestion des contacts et de l’annuaire téléphonique

- clientlist/ et clientdetail/ : gestion des clients

- utilisateurdetail/ : gestion des utilisateurs

- api/ : endpoints pour l’interface ou l’intégration externe

- classes/ : logique métier et objets PHP

- config/, database/ : configuration et scripts SQL

- login/ : authentification et gestion des sessions

- partials/ : fragments d’interface (header, footer…)

- utils/ : utilitaires divers

- docs/, Documentation\_LDAP/ : documentation technique et d’intégration

### 7. Livrables attendus

- Code source complet (PHP, CSS, JS, SQL)

- Documentation technique (structure, installation, sécurité)

- Cahier de tests fonctionnels (scénarios d’accès, sécurité, import/export)

- Fichiers d’exemple pour l’import/export CSV

### 8. Déploiement et maintenance

- Déploiement sur un serveur compatible PHP/MySQL, avec configuration sécurisée (HTTPS recommandé)

- Scripts d’installation et de migration de base de données fournis

- Procédures de sauvegarde et de restauration des données à documenter

### 9. Sécurité et RGPD

- Protection des données personnelles (sessions, accès, logs limités)

- Conformité RGPD à assurer (mentions légales, droits des utilisateurs)

- Certificat SSL recommandé pour toutes les communications

### 10. Tests et validation

- Rédaction d’un cahier de tests couvrant tous les rôles et scénarios critiques

- Tests d’import/export CSV avec gestion des erreurs

- Validation finale par démonstration auprès du tuteur

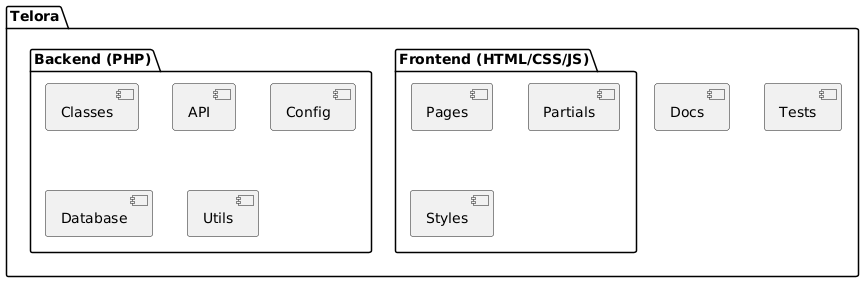
# Diagramme de gantt

# LDAP - Telora

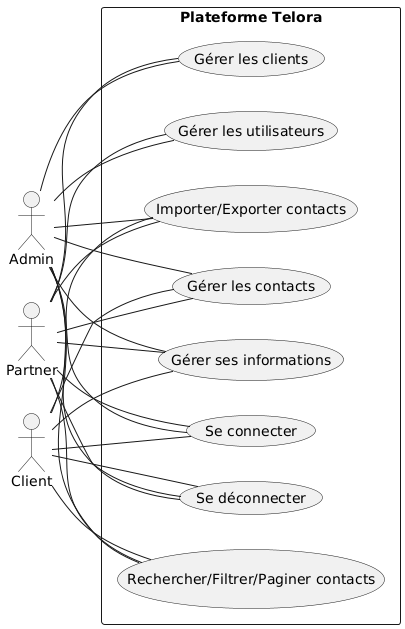
## UML

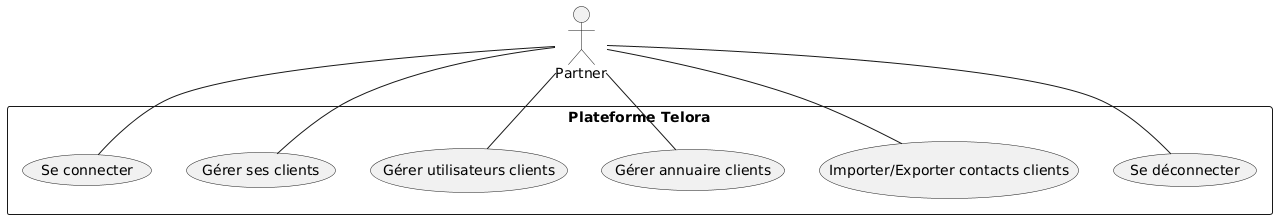
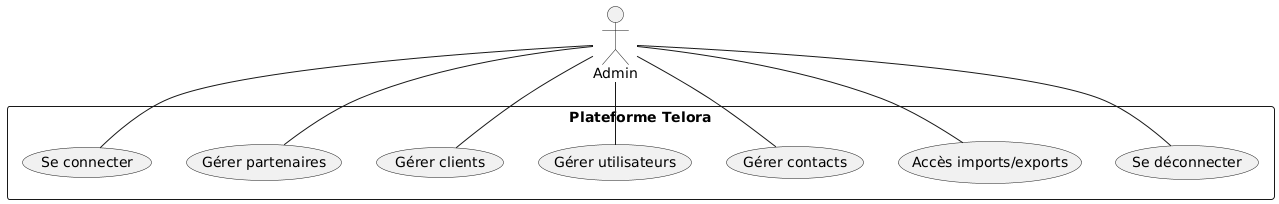
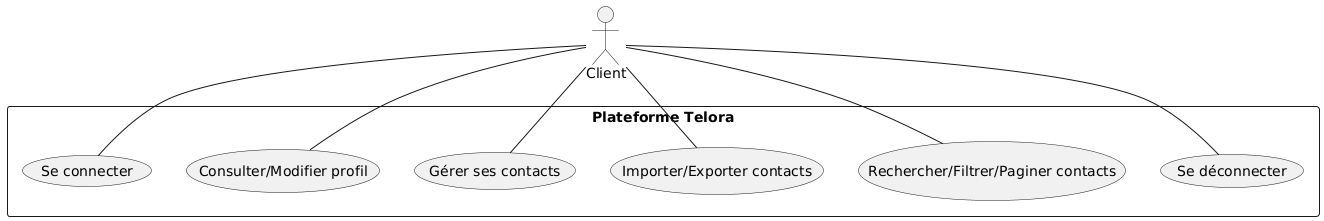
### 1. Diagramme de contexte

### 2. Diagramme de package

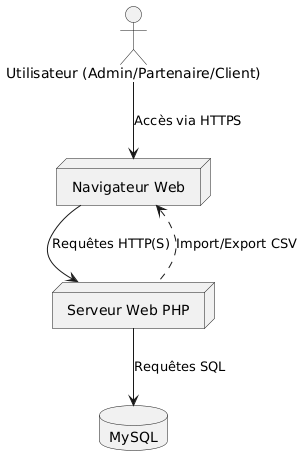


### 3. Diagramme use\_case

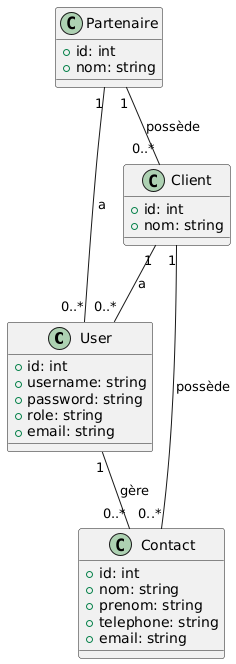




### 4. Cartographie technique



### 5. Diagramme de classe



# LDAP – Telora

## Merise – Modélisation BDD

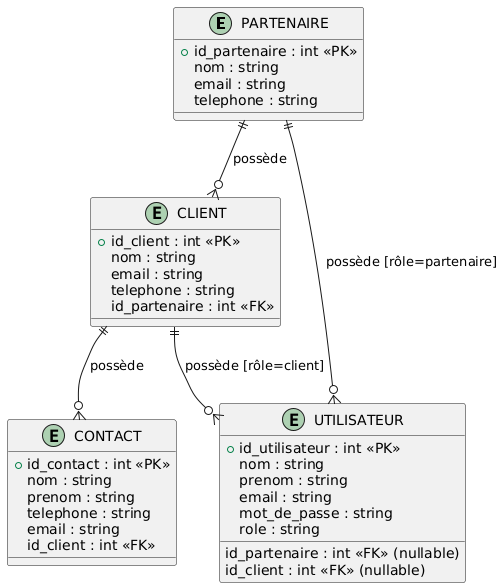
### 1. Dictionnaire de données

### 2. MCD (Modelé conceptuel de données)

### 

### 

### 3. MLD (Modèle logique de données)



### 4. MPD (Modèle physique de données)

# 1ClickAllEat

## Extraits de code commentés

### 1. Authentification et gestion des rôles

# 

# Explication détaillée :

# session\_start()

# démarre la session PHP, permettant d’accéder aux variables de session pour suivre l’état de connexion de l’utilisateur.

# La condition

# if (!isset($\_SESSION['role']))

# vérifie si l’utilisateur est authentifié (un rôle doit être défini en session après connexion).

# Si l’utilisateur n’est pas connecté, il est redirigé vers la page de login, ce qui protège toutes les pages sensibles contre l’accès non autorisé.

# Les variables de session comme

# role

# et

# partner\_id

# sont utilisées pour adapter dynamiquement l’interface et les droits d’accès. Par exemple, un partenaire ne pourra voir que ses propres clients, un client que ses propres contacts, etc.

# Ce mécanisme centralise la gestion des droits et garantit la séparation stricte des accès selon le rôle, comme exigé dans le cahier des charges.

### 2. Récupération et affichage paginé des contacts

# **Explication détaillée :**

# Cette séquence gère la pagination pour l’annuaire des contacts d’un client.

# Elle récupère la page courante et le nombre d’éléments à afficher par page depuis l’URL (GET), avec des valeurs par défaut sécurisées.

# **countAnnuaireByClient($clientsId)** compte le nombre total de contacts pour le client, ce qui permet de calculer le nombre total de pages.

# L’offset (

# $offset

# ) détermine à partir de quel contact commencer l’affichage pour la page courante.

# **getAnnuaireByClientPaginated($clientsId, $offset, $perPage)** récupère uniquement les contacts à afficher pour cette page, optimisant les performances même pour de gros volumes de données.

# Ce mécanisme améliore l’ergonomie et la rapidité de navigation dans l’annuaire.

### 3. Modification d’un contact

# **Explication détaillée :**

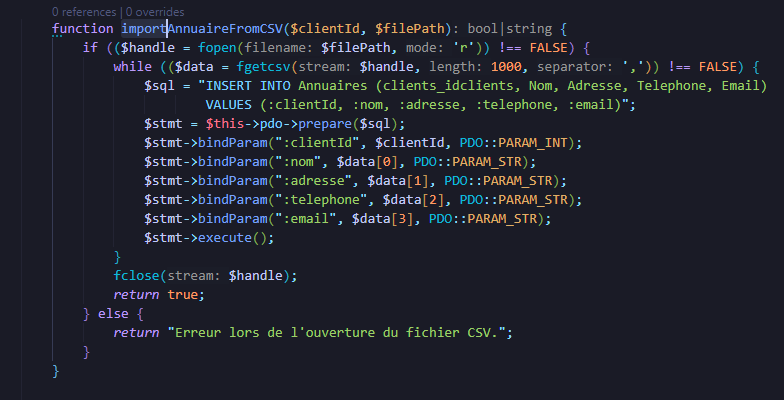
# Cette méthode met à jour toutes les informations d’un contact dans la base de données.

# Elle utilise une requête SQL préparée, ce qui protège contre les injections SQL et garantit la sécurité des données.

# Chaque champ du contact (prénom, nom, email, etc.) est lié à un paramètre de la requête pour éviter toute altération malveillante.

# En cas d’erreur lors de la mise à jour (par exemple, si la base est inaccessible), une exception claire est levée, facilitant le debug et la robustesse de l’application.

# Cette méthode est appelée lors de la modification d’un contact depuis l’interface utilisateur.

4. Import de contacts depuis un fichier CSV

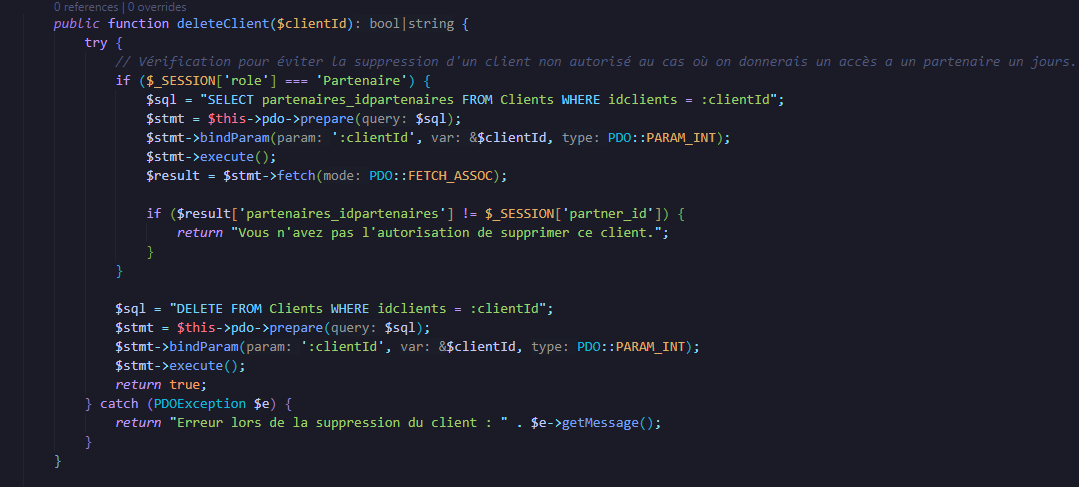
# **Explication détaillée :**

# Cette fonction permet d’importer en masse des contacts pour un client à partir d’un fichier CSV.

# Elle lit chaque ligne du fichier, nettoie les données (suppression des caractères spéciaux, normalisation des champs) et insère chaque contact dans la base.

# Elle gère les erreurs de format (colonnes manquantes, emails invalides, etc.) et évite les doublons, assurant la qualité des données importées.

# Ce mécanisme facilite la migration de données ou l’intégration de nouveaux clients disposant déjà d’un annuaire.

5. Suppression d’un client

# **Explication détaillée :**

# Cette méthode supprime un client de la base de données à partir de son identifiant.

# L’utilisation d’une requête préparée garantit la sécurité de l’opération.

# Elle permet de gérer la suppression d’un client depuis l’interface d’administration ou de gestion partenaire, en respectant les droits d’accès.

6. Connexion utilisateur

**Explication détaillée :**

Cette méthode gère la connexion d’un utilisateur à l’application.

Elle recherche l’utilisateur dans la base à partir du login, puis vérifie le mot de passe avec 

password\_verify

,qui prend en charge le hachage sécurisé.

Si la connexion réussit, elle initialise la session et stocke l’identifiant et le rôle de l’utilisateur, permettant de personnaliser l’accès à l’application.

La fonction 

redirectBasedOnRole

permet de rediriger l’utilisateur vers l’espace qui lui correspond (admin, partenaire, client).

En cas d’échec, la méthode retourne

# false

# ce qui permet d’afficher un message d’erreur à l’utilisateur.

# LDAP – Telora

## Plan de test qualité

1. Objectif

Valider que l’application actuelle répond aux besoins métiers (gestion utilisateurs, clients, partenaires, annuaire), assure la sécurité des données, la robustesse des traitements, et offre une expérience utilisateur cohérente sur tous supports.

2. Périmètre du test

**Fonctionnalités principales à tester**

* Authentification (login, logout, inscription)
* Gestion des utilisateurs (création, modification, suppression, affichage détail)
* Gestion des clients (CRUD, affichage liste/détail)
* Gestion des partenaires/restaurants (CRUD, accès admin)
* Gestion de l’annuaire (CRUD contacts, import/export CSV, recherche)
* Gestion des accès différenciés (admin, client)
* Sécurité (chiffrement, gestion des mots de passe)
* Responsive design (desktop, mobile)
* Import/export de données (CSV, LDAP)
* Scripts techniques (test TCP, import LDAP)

**Contraintes techniques**

* PHP custom, pas de framework
* Sécurité basique (chiffrement, gestion sessions)
* Tests techniques manuels/automatisés via scripts dédiés (pas de PHPUnit)
* Pas de CI/CD détecté

3. Méthodologie

* Tests manuels sur chaque fonctionnalité métier (CRUD, login, etc.)
* Tests d’intégration sur les interactions modules (ex : ajout client → visible dans annuaire)
* Tests de sécurité (mot de passe, session, accès différencié)
* Tests de performance sur les scripts d’import/export et requêtes lourdes
* Tests de compatibilité (navigateurs, responsive)
* Tests techniques via scripts dédiés (testTCP, LDAP\_test)

4. Tableau du plan de test

5. Exécution, reporting et validation

* Toutes les anomalies sont tracées dans un outil de suivi (ex : GitHub Issues).
* Un rapport de test est généré à chaque fin de cycle.
* La mise en production n’est validée que si tous les tests critiques sont passés et les anomalies bloquantes corrigées.

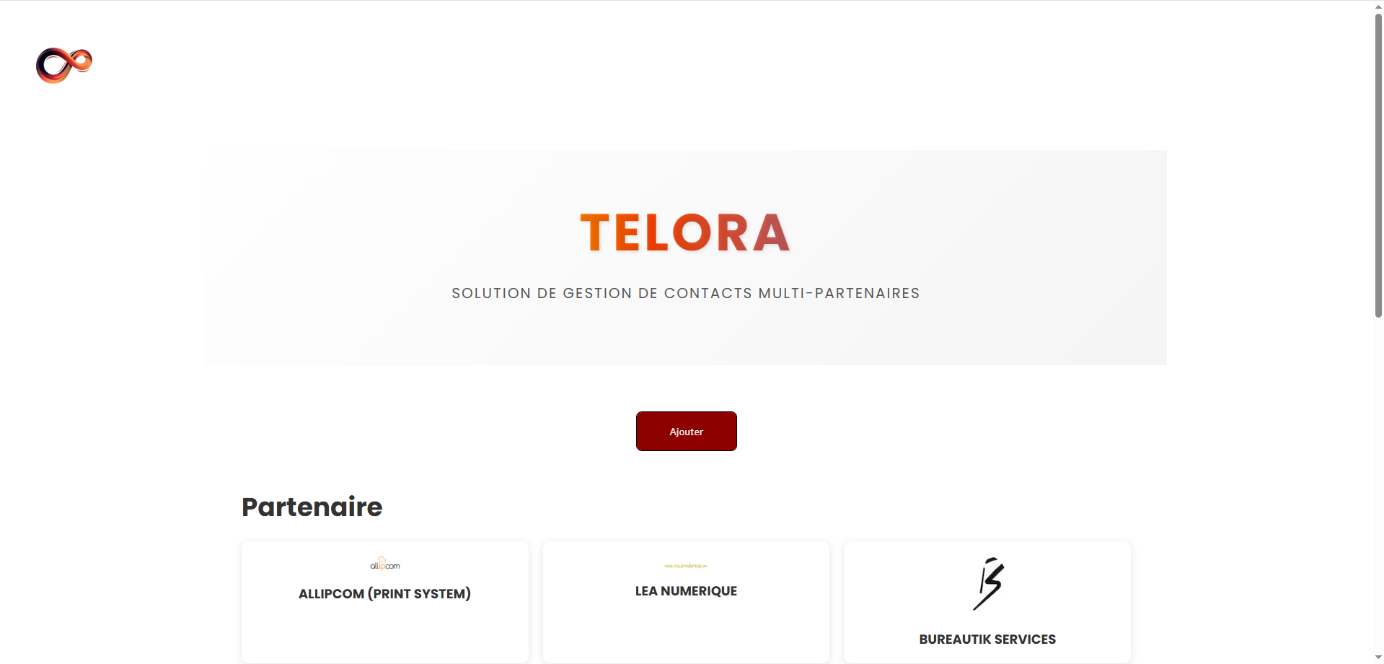
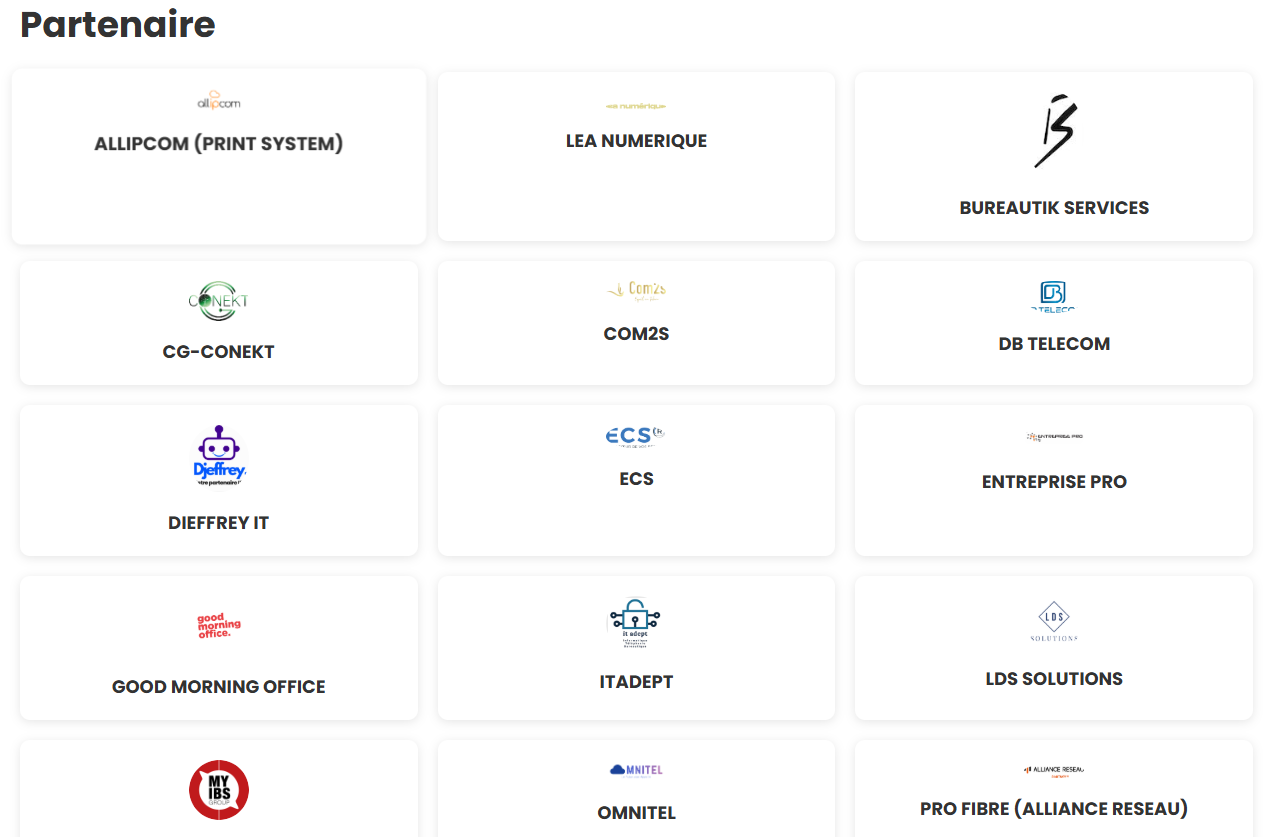
6. Conclusion

Ce plan de test qualité est adapté à la structure et aux fonctionnalités réelles de ton application. Il doit être révisé à chaque évolution majeure ou correction.

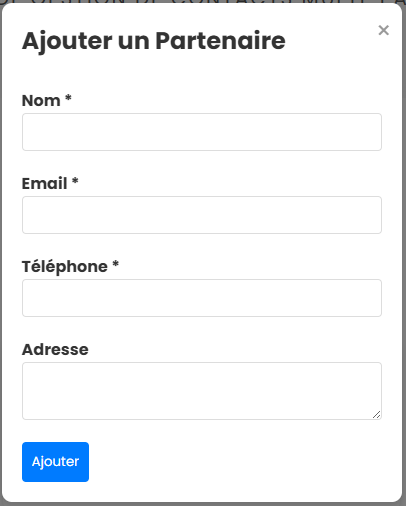
# Telora - LDAP

## Guide Utilisateur

Fonctionnalités principales :

* Visualisation rapide de tous les partenaires en un coup d'œil
* Organisation en grille pour une navigation facile
* ****Accès direct aux détails de chaque partenaire en cliquant sur sa carte

Écran 2 : Ajout d'un Nouveau Partenaire

****Cette fenêtre modale vous permet d'ajouter un nouveau partenaire à votre système.

**Champs à remplir :**

* Nom\* (obligatoire) : Nom commercial du partenaire
* Email\* (obligatoire) : Adresse email de contact principale
* Téléphone\* (obligatoire) : Numéro de téléphone du partenaire
* Adresse : Adresse physique du partenaire (facultatif)

**Actions :**

* Bouton "Ajouter" : Valide les informations et crée le nouveau partenaire
* Bouton "×" (en haut à droite) : Ferme la fenêtre sans enregistrer

**Note importante :** Les champs marqués d'un astérisque (\*) sont obligatoires et doivent être remplis pour pouvoir ajouter un partenaire.

Écran 3 : Page d'Accueil Telora avec Section Partenaires

Cet écran montre la page d'accueil de Telora qui intègre la section des partenaires.

**Éléments principaux :**

* En-tête avec logo Telora et infinity loop (en haut à gauche)
* Titre principal "TELORA" au centre
* Sous-titre descriptif : "SOLUTION DE GESTION DE CONTACTS MULTI-PARTENAIRES"
* Bouton "Ajouter" rouge qui permet d'accéder à la fenêtre d'ajout de partenaire
* Section "Partenaire" affichant les premiers partenaires de la liste (ALLIPCOM, LEA NUMERIQUE, BUREAUTIK SERVICES)

**Procédures Courantes**

**Pour ajouter un nouveau partenaire :**

1. Depuis l'écran principal ou la page d'accueil, cliquez sur le bouton "Ajouter"
2. Dans la fenêtre qui apparaît, remplissez les champs obligatoires (Nom, Email, Téléphone)
3. Complétez l'adresse si nécessaire
4. Cliquez sur le bouton bleu "Ajouter" pour valider

**Pour consulter tous les partenaires :**

1. Accédez à la section "Partenaire" depuis la page d'accueil
2. Parcourez la grille des partenaires pour voir l'ensemble des entreprises partenaires

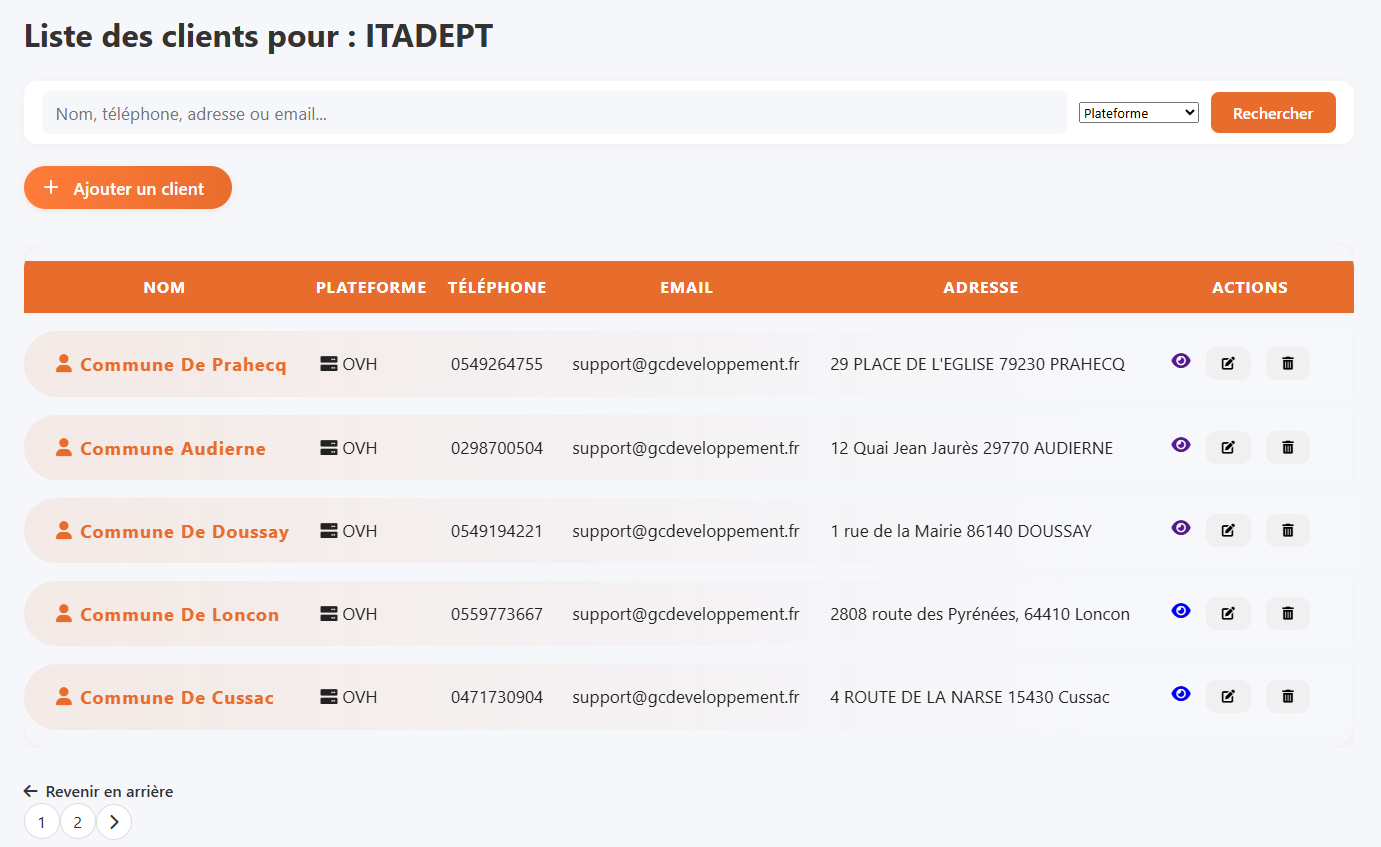
**Remarques importantes :**

* Seuls les administrateurs de la marque blanche ont accès à cette section de gestion des partenaires
* Le système utilise la technologie LDAP pour stocker et gérer les informations des partenaires
* Les modifications apportées aux partenaires sont immédiatement répercutées dans tout le système Telora

Ce guide vous aide à naviguer efficacement dans la section administrative des partenaires de Telora, facilitant ainsi la gestion de vos relations professionnelles.

**Vue Admin des Clients par Partenaire**

Cette section du guide utilisateur se concentre sur la gestion des clients associés à un partenaire spécifique. En complément de la gestion des partenaires, cette fonctionnalité permet d'administrer la liste des clients pour chaque partenaire dans le système Telora.

Écran 1 : Liste des Clients d'un Partenaire

Cet écran affiche la liste complète des clients associés au partenaire sélectionné (dans l'exemple : ITADEPT).

**Fonctionnalités principales :**

* Titre indiquant le partenaire sélectionné ("Liste des clients pour : ITADEPT")
* Barre de recherche permettant de filtrer par nom, téléphone, adresse ou email
* Menu déroulant pour filtrer par plateforme
* Bouton "Rechercher" orange pour lancer la recherche
* Bouton "Ajouter un client" pour créer un nouveau client
* Tableau des clients avec colonnes organisées (Nom, Plateforme, Téléphone, Email, Adresse, Actions)
* Pagination en bas de page (numéros 1, 2, etc.)
* Bouton "Revenir en arrière" pour retourner à la vue précédente

**Données du tableau :**

* Nom du client (ex: Commune De Prahecq, Commune Audierne, etc.)
* Plateforme utilisée (OVH pour tous les clients dans l'exemple)
* Numéro de téléphone du client
* Adresse email ([support@gcdeveloppement.fr](mailto:support@gcdeveloppement.fr) pour tous les clients dans l'exemple)
* Adresse physique du client
* Actions disponibles (icônes pour visualiser, modifier, supprimer)

Écran 2 : Modifier un Client

Cette fenêtre modale permet de modifier les informations d'un client existant.

**Champs disponibles :**

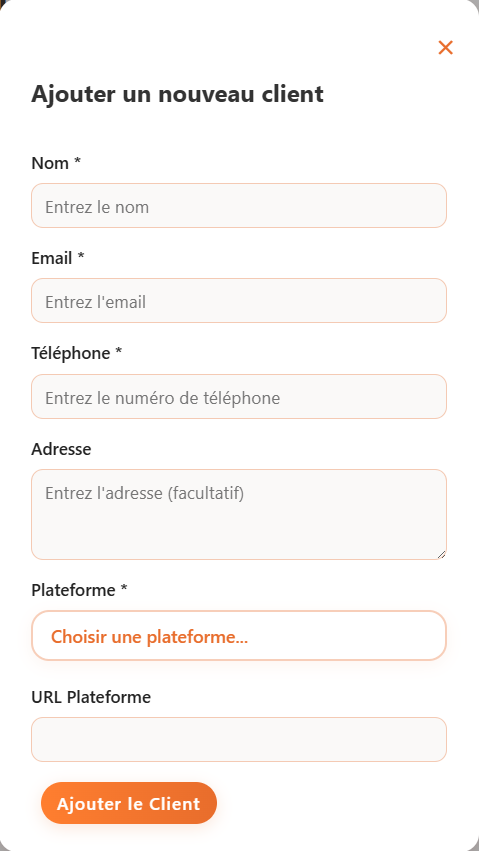
* Nom\* : Nom du client (exemple : "Commune de Prahecq")
* Email\* : Adresse email du client (exemple : "[support@gcdeveloppement.fr](mailto:support@gcdeveloppement.fr)")
* Téléphone\* : Numéro de téléphone du client
* Adresse : Adresse physique (exemple : "29 PLACE DE L'EGLISE 79230 PRAHECQ")
* Plateforme\* : Type de plateforme utilisée (exemple : "OVH")
* URL Plateforme : Adresse URL associée (exemple : "fr.proxysip.eu")

**Actions :**

* Bouton "Enregistrer les modifications" pour valider les changements
* Bouton "×" en haut à droite pour annuler et fermer la fenêtre

Écran 3 : Ajouter un Nouveau Client

Cette fenêtre modale permet d'ajouter un nouveau client au partenaire sélectionné.

**Champs à remplir :**

* Nom\* : Champ pour entrer le nom du nouveau client
* Email\* : Champ pour entrer l'adresse email
* Téléphone\* : Champ pour entrer le numéro de téléphone
* Adresse : Champ pour entrer l'adresse physique (facultatif)
* Plateforme\* : Menu déroulant pour sélectionner une plateforme
* URL Plateforme : Champ pour entrer l'URL associée à la plateforme

**Actions :**

* Bouton "Ajouter le Client" pour créer le nouveau client
* Bouton "×" en haut à droite pour annuler et fermer la fenêtre

**Note importante :** Les champs marqués d'un astérisque (\*) sont obligatoires.

**Procédures Courantes**

**Pour rechercher un client :**

1. Utilisez la barre de recherche en haut pour entrer des mots-clés (nom, téléphone, adresse, email)
2. Sélectionnez éventuellement une plateforme spécifique dans le menu déroulant
3. Cliquez sur le bouton "Rechercher" pour filtrer les résultats

**Pour ajouter un nouveau client :**

1. Cliquez sur le bouton orange "Ajouter un client"
2. Remplissez tous les champs obligatoires dans la fenêtre qui s'ouvre
3. Complétez les champs facultatifs si nécessaire
4. Cliquez sur "Ajouter le Client" pour valider

**Pour modifier un client existant :**

1. Localisez le client dans la liste
2. Cliquez sur l'icône de modification (crayon) dans la colonne "Actions"
3. Modifiez les champs souhaités dans la fenêtre qui s'ouvre
4. Cliquez sur "Enregistrer les modifications" pour valider

**Pour supprimer un client :**

1. Localisez le client dans la liste
2. Cliquez sur l'icône de suppression (poubelle) dans la colonne "Actions"
3. Confirmez la suppression si une confirmation est demandée

**Pour visualiser les détails d'un client :**

1. Localisez le client dans la liste
2. Cliquez sur l'icône de visualisation (œil) dans la colonne "Actions"

**Pour naviguer entre les pages de résultats :**

1. Utilisez les numéros de page en bas de l'écran (1, 2, etc.)
2. Utilisez la flèche droite pour avancer à la page suivante

**Remarques importantes :**

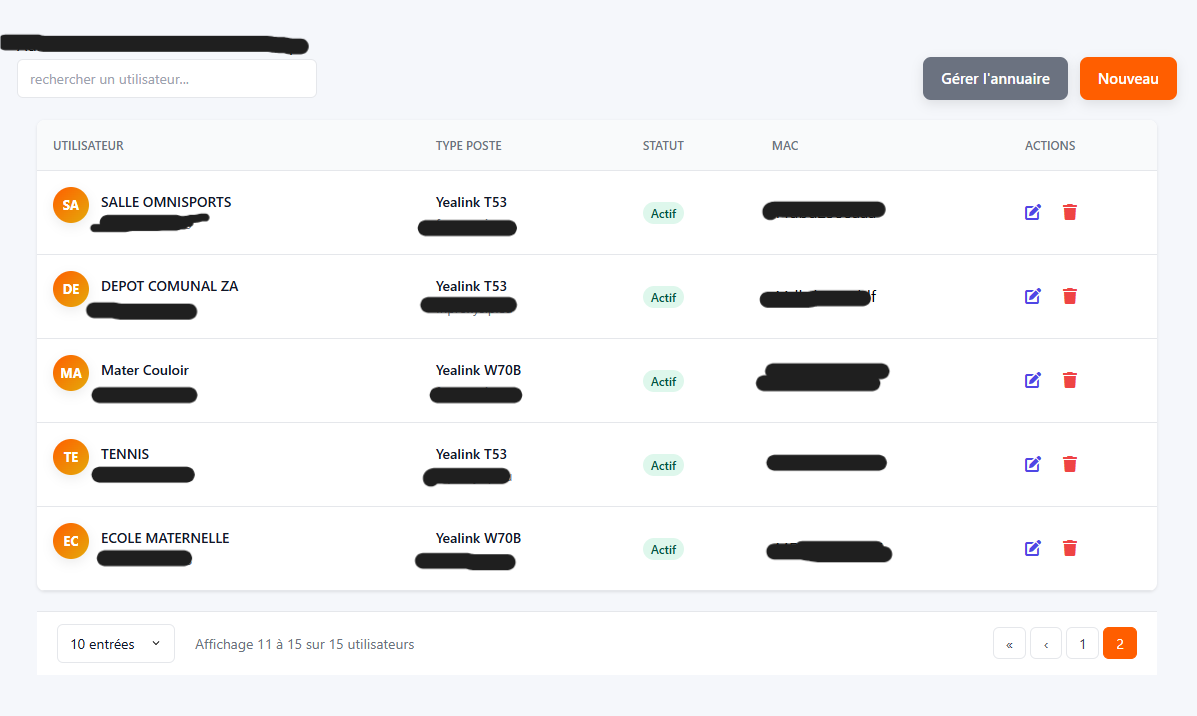
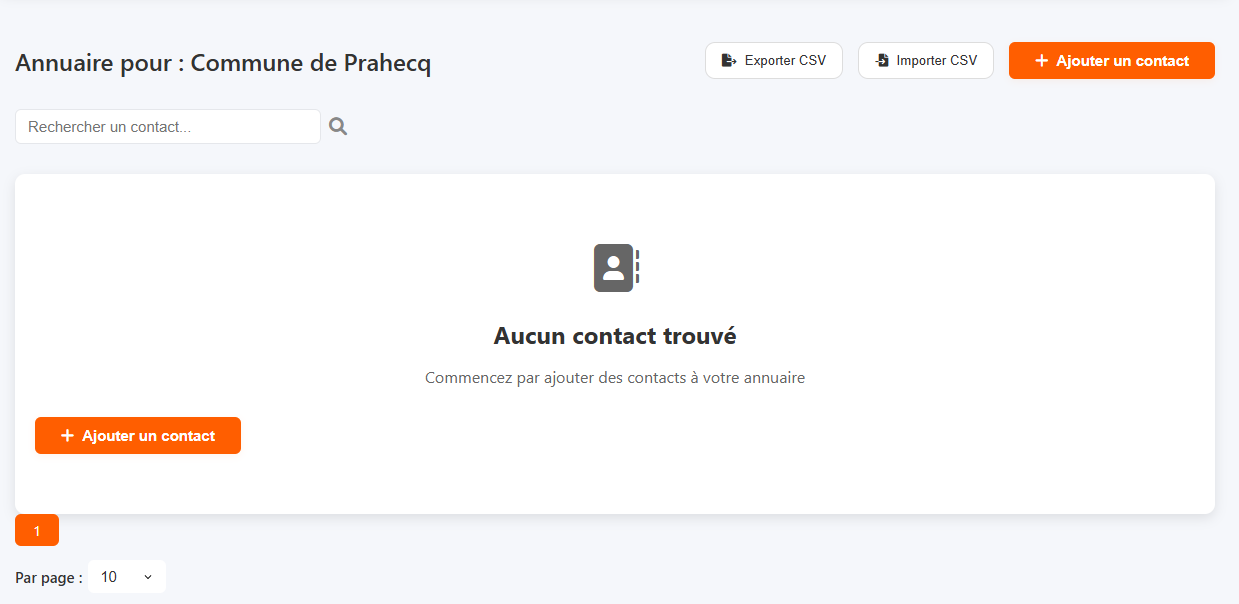
* Tous les clients affichés sont associés uniquement au partenaire sélectionné (ITADEPT dans l'exemple)
* Le système conserve une relation hiérarchique entre partenaires et clients
* La plateforme indique le service d'hébergement ou la technologie utilisée par le client
* L'URL Plateforme peut être utilisée pour accéder directement aux services du client

**Vue Principale des Utilisateurs**

**Fonctionnalités de l'écran principal**

L'écran principal des utilisateurs vous présente la liste de tous les utilisateurs enregistrés dans le système avec leurs informations clés.

**Éléments principaux :**

* Barre de recherche pour trouver rapidement un utilisateur
* Bouton "Gérer l'annuaire" en gris pour accéder à l'annuaire associé
* Bouton "Nouveau" en orange pour créer un nouvel utilisateur
* Tableau organisé avec les colonnes : Utilisateur, Type Poste, Statut, MAC, Actions

**Information affichée pour chaque utilisateur :**

* Avatar avec initiales (SA, DE, MA, TE, EC)
* Nom de l'utilisateur (ex. SALLE OMNISPORTS, DEPOT COMUNAL ZA)
* Type de poste téléphonique (Yealink T53, Yealink W70B)
* Statut (Actif pour tous les utilisateurs dans l'exemple)
* Adresse MAC du périphérique
* Actions disponibles (modification et suppression)

**Navigation :**

* Sélecteur de nombre d'entrées par page (10 entrées)
* Indication du nombre total d'utilisateurs affichés (Affichage 11 à 15 sur 15 utilisateurs)
* Pagination avec numéros de page (1, 2) et boutons de navigation

**Vue Annuaire d'un Client**

Après avoir cliqué sur "Gérer l'annuaire" pour un client spécifique, vous accédez à la vue de l'annuaire pour ce client (dans l'exemple : Commune de Prahecq).

**Fonctionnalités de l'écran annuaire**

**Éléments principaux :**

* Titre indiquant le client concerné ("Annuaire pour : Commune de Prahecq")
* Barre de recherche pour trouver rapidement un contact
* Boutons d'import/export (Exporter CSV, Importer CSV)
* Bouton "Ajouter un contact" en orange
* Zone d'affichage des contacts (vide dans l'exemple avec le message "Aucun contact trouvé")
* Indication pour commencer à ajouter des contacts
* Pagination en bas de page

**État initial :**

* Message "Aucun contact trouvé" avec icône de contact
* Suggestion "Commencez par ajouter des contacts à votre annuaire"
* Bouton "Ajouter un contact" également disponible dans la zone d'affichage

**Options de pagination :**

* Navigation par pages numérotées (1 dans l'exemple)
* Réglage du nombre de contacts par page (10 par défaut)

**Procédures Courantes**

**Pour rechercher un utilisateur :**

1. Sur l'écran principal, utilisez la barre de recherche en haut "rechercher un utilisateur..."
2. Tapez le nom ou une partie du nom de l'utilisateur que vous cherchez
3. Le système filtre automatiquement la liste pour afficher les résultats correspondants

**Pour créer un nouvel utilisateur :**

1. Cliquez sur le bouton orange "Nouveau" en haut à droite de l'écran principal
2. Remplissez les informations requises dans le formulaire qui s'affiche
3. Enregistrez les informations pour créer le nouvel utilisateur

**Pour modifier un utilisateur existant :**

1. Localisez l'utilisateur dans la liste
2. Cliquez sur l'icône de modification (crayon) dans la colonne "Actions"
3. Mettez à jour les informations nécessaires
4. Enregistrez les modifications

**Pour supprimer un utilisateur :**

1. Localisez l'utilisateur dans la liste
2. Cliquez sur l'icône de suppression (poubelle) dans la colonne "Actions"
3. Confirmez la suppression lorsque le système vous le demande

**Pour accéder à l'annuaire d'un client :**

1. Cliquez sur le bouton "Gérer l'annuaire" en haut à droite de l'écran principal
2. Vous serez dirigé vers l'annuaire du client sélectionné

**Pour ajouter un contact à l'annuaire :**

1. Dans la vue annuaire, cliquez sur le bouton orange "Ajouter un contact"
2. Remplissez les informations du contact dans le formulaire qui s'affiche
3. Enregistrez pour ajouter le contact à l'annuaire

**Pour importer des contacts depuis un fichier CSV :**

1. Dans la vue annuaire, cliquez sur le bouton "Importer CSV"
2. Sélectionnez le fichier CSV contenant vos contacts depuis votre ordinateur
3. Suivez les instructions pour finaliser l'importation

**Pour exporter l'annuaire au format CSV :**

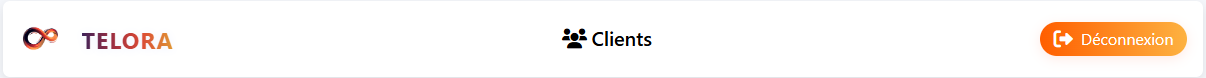
1. Dans la vue annuaire, cliquez sur le bouton "Exporter CSV"
2. Le fichier CSV contenant tous les contacts de l'annuaire sera téléchargé sur votre ordinateur

**Points Importants à Noter**

* **Utilisateurs vs Contacts** : Les utilisateurs sont les équipements téléphoniques configurés dans le système, tandis que les contacts sont les entrées dans l'annuaire pour les appels.
* **Annuaires multiples** : Un client peut avoir plusieurs annuaires, accessibles via le bouton "Gérer l'annuaire".
* **Statut des utilisateurs** : Le statut "Actif" indique que l'utilisateur est correctement configuré et opérationnel.
* **Types de postes** : Les types de postes (Yealink T53, Yealink W70B) indiquent le modèle de téléphone attribué à l'utilisateur.
* **Adresse MAC** : L'adresse MAC est un identifiant unique du périphérique physique associé à l'utilisateur.

**Barre de Navigation**

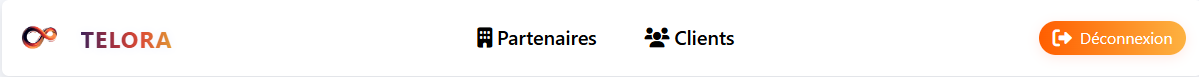
La barre de navigation de Telora est l'élément principal permettant de naviguer entre les différentes sections de l'application. L'interface s'adapte en fonction de votre rôle dans le système (administrateur, partenaire ou client).

**Barre de Navigation Client et Partenaire**

**Éléments de la barre de navigation client :**

* Logo Telora avec l'infinity loop à gauche
* Texte "TELORA" en violet
* Bouton "Clients" au centre avec icône représentant des personnes
* Bouton "Déconnexion" orange à droite avec icône de sortie

Cette interface simplifiée permet aux clients d'accéder uniquement à la gestion de leurs propres données, sans pouvoir consulter ou modifier les informations relatives aux partenaires.

**Barre de Navigation Administrateur**

**Éléments de la barre de navigation administrateur/partenaire :**

* Logo Telora avec l'infinity loop à gauche
* Texte "TELORA" en violet
* Bouton "Partenaires" avec icône de bâtiment
* Bouton "Clients" avec icône représentant des personnes
* Bouton "Déconnexion" orange à droite avec icône de sortie

Cette interface étendue permet aux administrateurs et partenaires d'accéder à la fois à la gestion des clients et des partenaires, offrant ainsi une vision plus complète du système.

**Différences Entre les Accès**

**Accès Client**

* Interface simplifiée
* Accès limité à la section "Clients" uniquement
* Peut consulter et gérer uniquement ses propres données
* Ne peut pas voir ni modifier les informations des partenaires

**Accès Partenaire**

* Interface plus complète
* Accès aux sections "Clients" et "Partenaires"
* Peut consulter et gérer ses propres informations et celles des clients qui lui sont associés
* Possibilité de naviguer entre la vue partenaire et la vue client

**Accès Administrateur**

* Interface complète
* Accès total aux sections "Clients" et "Partenaires"
* Peut consulter et gérer toutes les informations dans le système
* Droits étendus pour ajouter, modifier et supprimer des partenaires et clients

**Navigation dans l'Application**

**Pour accéder à la section Clients :**

1. Cliquez sur le bouton "Clients" dans la barre de navigation
2. Vous serez dirigé vers la liste des clients associés à votre compte

**Pour accéder à la section Partenaires (administrateurs et partenaires uniquement) :**

1. Cliquez sur le bouton "Partenaires" dans la barre de navigation
2. Vous serez dirigé vers la liste des partenaires disponibles

**Pour se déconnecter :**

1. Cliquez sur le bouton orange "Déconnexion" à droite de la barre de navigation
2. Vous serez déconnecté et redirigé vers la page de connexion

**Points Importants à Noter**

* La barre de navigation est toujours visible en haut de l'écran, quelle que soit la section dans laquelle vous vous trouvez
* Les fonctionnalités disponibles dans chaque section dépendent de votre niveau d'accès
* Le bouton actuellement actif (Clients ou Partenaires) est généralement mis en évidence pour indiquer dans quelle section vous vous trouvez
* L'interface s'adapte automatiquement en fonction de votre rôle dans le système
* Le bouton "Déconnexion" est toujours accessible pour quitter l'application de manière sécurisée