

# Cahier des charges technique

15/10/2025

MATEJKA Milan  
MAURICE Simon

# Projet : Jardins de Cocagne

L'application vise à digitaliser la gestion des opérations principales d'un Jardin, notamment l'administration des parcours d'insertion, des abonnements aux paniers, et la logistique des livraisons.



*Réseau Cocagne*



## Sommaire :

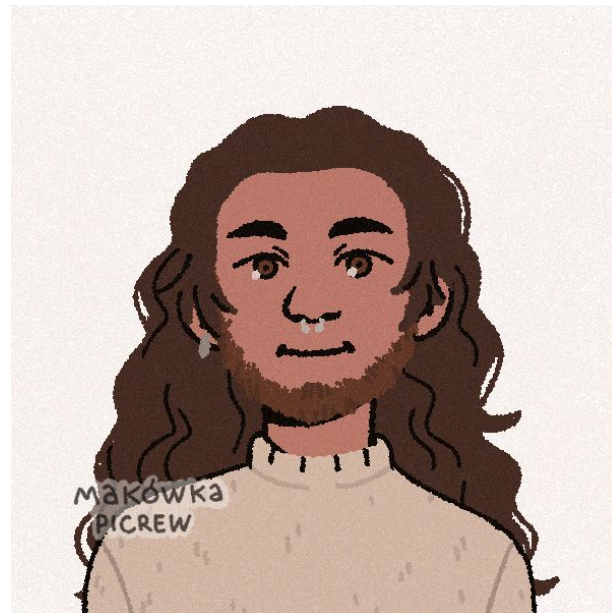
- Présentation de l'équipe
- Back-End
- Front-End
- Base de données
- Environnement de développement
- Représentation d'un jardin de cocagne
- Hébergement du projet
- Solutions pour les tests

# Présentation de l'équipe



Milan MATEJKA

Développeur Front/Back-End



Simon MAURICE

Développeur Front/Back-End

# Back-End

---





Symfony

## Symfony/PHP

Pour ce projet on va utiliser symfony qui nous donne de nombreux avantages :

Symfony est un framework reconnu pour sa maturité et sa documentation complète, assurant un développement solide et maintenable.

Symfony intègre des outils pour gérer les rôles et garantir la conformité RGPD dans la gestion des données

Associé à PHP 8.2, Symfony a des performances élevées, essentielles pour le Back-end et le Front-end.

# Doctrine ORM

Doctrine permet de représenter les relations sous forme de classes PHP simples.

L'utilisation de l'ORM garantit que toutes les interactions avec la base de données sont sécurisées, évitant les risques d'injections SQL et protégeant les données sensibles.

Doctrine gère automatiquement les migrations (les changements de structure de la base de données). Si nous devons ajouter un champ, le processus est automatisé et sécurisé.

L'équipe de développement utilise le langage PHP au lieu du SQL brut pour la plupart des requêtes, ce qui accélère la mise en place des modules Back-end





# Front-End

---



**HTML**



**CSS**





# HTML



# CSS



## HTML5/CSS

Utilisation des balises sémantiques de HTML5 pour construire une interface claire et accessible

Garantie d'un design entièrement responsive. L'application sera utilisable de manière optimale sur tous les supports

Utilisation des standards CSS3 pour améliorer l'expérience utilisateur et la navigation, rendant la gestion des données pour l'équipe

# Twig

Twig permet de séparer clairement le Back-end(gérée par Symfony et PHP) du Front-end (HTML/CSS). Cela rend le Back-end et le Front-end plus faciles à maintenir et à faire évoluer indépendamment.

Twig est indispensable pour afficher dynamiquement les données extraites de la base MySQL via Doctrine.

Grâce à l'héritage de templates et aux macros (composants réutilisables), nous pouvons créer un layout standard (en-tête, pied de page, navigation) pour tous les modules de l'application.



# Leaflet



Leaflet est une bibliothèque JavaScript légère et open source. Elle garantit un chargement rapide de la carte des Dépôts, même sur les appareils mobiles.

La carte permettra à l'équipe de : visualiser précisément l'emplacement de tous les Dépôts, mettre en place des tournées de livraison.

Les Adhérents pourront consulter une carte dans leur espace personnel pour localiser rapidement leur dépôt de retrait.

L'intégration de Leaflet est compatible avec Twig. Les coordonnées GPS des Dépôts, stockées via Doctrine/MySQL, seront injectées dynamiquement dans la carte.

# Base de donnée

---



# MySQL

MySQL est la référence pour les applications web. Il garantit la cohérence et l'intégrité des données , essentielle pour la gestion précise des Adhérent et des Livraisons.

La version 8.0 offre des améliorations de performance significatives. Elle est capable de supporter l'augmentation progressive du nombre d'Adhérents et des historiques de Livraisons sans dégradation majeure des performances.

MySQL s'intègre parfaitement avec PHP et Doctrine ORM.

La base sera configurée en UTF-8mb4 pour supporter tous les caractères spéciaux. Les données personnelles des adhérents seront stockées en respectant les contraintes de sécurité et les exigences RGPD.



# PhpMyAdmin



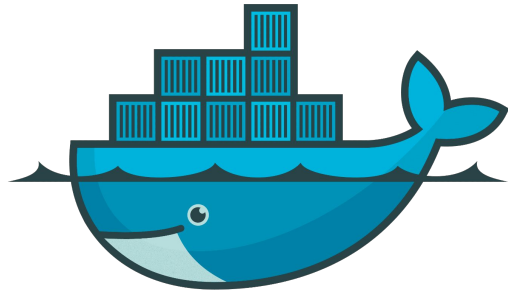
phpMyAdmin est une interface web qui permet à l'équipe de développement de visualiser, gérer et modifier la structure de la base de données MySQL sans avoir à utiliser de commandes en ligne complexes.

En phase de développement, il permet de vérifier rapidement si les données sont correctement insérées par Symfony et Doctrine.

phpMyAdmin est conteneurisé directement dans le `docker-compose.yml`. Il est disponible localement dès le démarrage de l'environnement de développement, offrant un accès immédiat pour l'équipe technique.

# Environnement de développement

---



docker





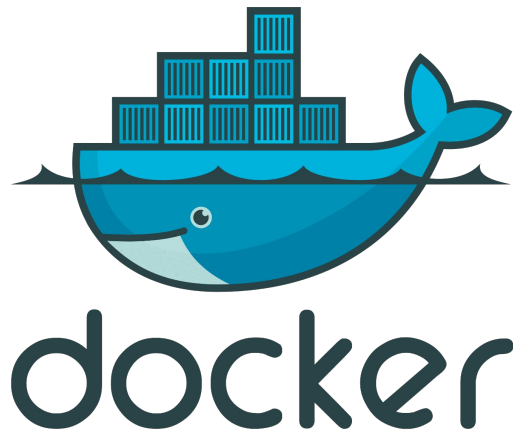
# Docker

Docker encapsule l'application et ses dépendances (PHP 8.2, MySQL 8.0) dans des conteneurs isolés. Cela garantit que l'environnement de développement local est identique à l'environnement de production.

Le fichier docker-compose.yml permet de démarrer l'intégralité du projet avec une seule commande. C'est crucial pour la configuration de l'environnement de développement..

Chaque composant fonctionne dans son propre conteneur. Si un service rencontre un problème, il n'affecte pas l'ensemble de l'application, ce qui facilite grandement le débogage.

Docker assure que les versions strictes des technologies (ex. : MySQL 8.0, PHP 8.2).



# NGINX



Nginx est reconnu pour sa capacité à gérer un grand nombre de connexions simultanées de manière efficace. Il est idéal pour servir rapidement les pages du Front-end consultées par tous les adhérents.

Nginx sert de façade publique et transfère les requêtes dynamiques au conteneur php-fpm. Il assure la bonne coordination entre les composants de l'architecture.

Nginx sera configuré pour mettre en œuvre les en-têtes de sécurité HTTP, agissant comme une première couche de protection avant même que le code Symfony ne s'exécute.

Nginx gère directement la distribution des fichiers statiques (CSS, JavaScript, images) sans solliciter PHP, ce qui permet des temps de chargement des pages HTML/CSS très rapides.



## Représentation d'un jardin de cocagne

Unity permet de créer un environnement interactive du jardin et des parcelles. Les adhérents et les partenaires pourront explorer virtuellement l'environnement où sont cultivés les produits

.Le projet Unity sera compilé en WebGL, ce qui permet de l'intégrer comme un composant sur la page web gérée par Symfony/Twig.. Il fonctionne sur la majorité des navigateurs.

Possibilité d'ajouter des éléments interactifs dans la simulation dans le future.

# Hébergement du projet

Platform.sh est optimisé pour les applications PHP/Symfony utilisant des conteneurs.

Le déploiement est activé par Git. Chaque fois que le code est mis à jour, Platform.sh prend le relais pour créer et déployer automatiquement la nouvelle version.

Pour chaque nouvelle fonctionnalité (ex. : la carte Leaflet ou le module Unity), un environnement de test temporaire est créé.

La plateforme gère la haute disponibilité, les mises à jour de sécurité au niveau du système et les sauvegardes automatiques de la base de données MySQL.



# Solutions pour les tests

---

PHPUnit



# PHPUnit / Panther



PHPUnit :

Assurer que les fonctions de Symfony sont précises et isolées. Le but est de couvrir au minimum 70% des fonctions.

Simuler les connexions et les actions de chaque rôle pour confirmer que les formulaires et les accès sécurisés fonctionnent comme prévu.

Panther :

Tester l'application en entière dans un navigateur pour valider les interactions complexes, y compris le bon fonctionnement de la carte Leaflet et de la simulation Unity