

# Cahier des charges techniques - SAE S5 Jardins de Cocagne

ALGUL Sefer - GRANDGIRARD Louis  
Parcours FI + Poursuite d'études

15/10/2025



UNIVERSITÉ  
DE LORRAINE








Saint-Dié-des-Vosges

# Sommaire

1. Contexte du projet
2. Objectifs techniques
3. Technologies retenues
4. Justification des choix
5. Interface utilisateur
6. API et Web services
7. Environnement collaboratif
8. Solution d'hébergement
9. Sécurité et qualité
10. Solutions pour les tests
11. Présentation de l'équipe
12. Conclusion

# 1. Contexte du projet

-  SAE 5 : Application pour les Jardins de Cocagne
-  Objectifs : gérer les livraisons, les paniers, les adhérents
-  Approche : développement incrémental, qualité logicielle, API
-  Qui est le client : Jardins de Cocagne
-  La finalité : optimisation logistique, simplicité d'usage

## 2. Objectifs techniques



# Symfony

A quoi sert ?	Avantages
Développer des applications web robustes et organisées	Structure claire et modulaire (facilite la maintenance et la compréhension du code)
Structurer le code selon le modèle MVC	Sécurité renforcée grâce à la gestion native de l'authentification, rôles et accès
Gérer la logique métier, la base de données et les interfaces de manière séparée	Performances et fiabilités
Sécuriser les applications (authentification, rôles, permissions)	Interopérabilité élevée avec d'autres outils (Twig, Doctrine ...)
Réutiliser des composants existant pour faciliter le développement	Réutilisation de composants
Garantir la maintenabilité et l'évolutivité du projet dans le temps.	Support long terme garantissant la stabilité des versions principales



# Bootstrap

A quoi sert ?	Avantages
Concevoir rapidement des interfaces web modernes et responsives	Gain de temps grâce à une bibliothèque complète de composants réutilisables
Standardiser le design grâce à des composants prêts à l'emploi	Responsive design intégré
Adapter automatiquement les pages à tous types d'écrans	Compatibilité multi-navigateurs
Accélérer le développement front-end sans devoir écrire tout le CSS à la main	Facile à personnaliser
Assurer la cohérence visuelle sur l'ensemble du site	Intégration simple avec d'autres technologies
Intégrer facilement JavaScript pour les éléments interactifs	Large communauté offrant de nombreux thèmes et exemples



# MySQL

A quoi sert ?	Avantages
Stocker, organiser et gérer les données d'une application web	Performance et fiabilité reconnues, même pour des bases volumineuses
Exécuter des requêtes SQL pour insérer, modifier, supprimer ou consulter des informations	Compatibilité étroite avec PHP et les frameworks comme Symfony
Assurer la cohérence et la sécurité des données grâce à un modèle relationnel	Gratuit et open source, tout en étant très stable
Etablir la connexion entre la couche applicative (Symfony) et la base de données	Sécurité configurable, avec gestion des utilisateurs et des droits d'accès
gérer de gros volumes de données tout en maintenant de bonnes performances	Outils graphiques disponibles



A quoi sert ?	Avantages
Générer les pages HTML d'une application Symfony de manière simple et structurée	Syntaxe claire et lisible
Séparer la logique métier du code d'affichage	Sécurisé par défaut
Insérer dynamiquement des données PHP dans les pages web	Compatible avec Symfony et ses composants internes
Organiser les vues grâce à un système de templates	Facile à maintenir et à étendre, même sur de gros projets
Faciliter la maintenance et la mise à jour de l'interface utilisateur	Performant, grâce à la compilation des templates



### 3. Technologies retenues

Couche	Langage principal	Framework / outils	Librairie / API	Alternatives envisagées	Solution retenue
Front-end	HTML / CSS / JS	Bootstrap	Leaflet.js; FullCalendar.js	React, Angular	Bootstrap + Leaflet.js
Back-end	PHP	Symfony		Laravel, SpringBoot	Symfony
Base de données	SQL		MySQL Workbench	PostgreSQL, SQLite	MySQL
Template	PHP / HTML	Twig		Laravel	Twig

## 4. Justificatif des choix

Nom	Raison
Symfony	Framework PHP – MVC – Sécurisé – Modulaire - Maintenable
MySQL	Base de données relationnelle – Performante – Fiable – Compatible PHP
Leaflet.js	Carte interactives – Open source – Léger - Gratuit
FullCalendar.js	Calendrier dynamique – Gestion des plannings – Intégration simple
Bootstrap	Interface responsive – Rapide – Moderne – Facile à utiliser

## 5. Interface utilisateur (admin)

- Interface simple via Twig + Bootstrap
- Modules de gestion des dépôts
- Modules de gestion tournées
- Modules de gestion paniers
- Modules de gestions adhérents

## 6. API et Web services

Objectifs	Exposer les données au format JSON pour permettre leur réutilisation.
Outil utilisé	Symfony
Fonctionnement	Routes d'accès sécurisées (GET, POST, PUT, DELETE) permettant à d'autres applications de consommer ou mettre à jour les données.
Avantages	Réutilisable – Sécurisée – Compatible – Facile à maintenir – Intégrée au projet Symfony

# 7. Environnement de développement collaboratif

## Plateforme de versionning et CI/CD

- GitLab :
- Héberge le dépôt
  - Permet la collaboration
  - Gère l'intégration continue



## Conteneurisation

- Docker :
- Fournit un environnement de développement
  - Facilite l'installation des services via des conteneurs
  - Evite les différences entre machines



## Environnement de développement

- VSCode :
- IDE utilisé pour éditer du code
  - Supporte les extensions Symfony et GitLab pour simplifier le travail



## Gestion de la base de données

- MySQL Workbench :
- Outil graphique
- Permet d'exécuter des requêtes SQL



## 8. Solution d'hébergement

Notre objectif est d'assurer un déploiement reproductible, stable et sécurisé de l'application Symfony.

Élément	Solution retenue	Justification
Méthode d'hébergement	Conteneurisation avec Docker Compose	Permet de reproduire facilement l'environnement de développement et de production
Serveur Web	Nginx	Serveur léger et performant pour le routage des requêtes Symfony
Base de données	MySQL	Base relationnelle robuste, compatible avec Doctrine
Orchestration	Docker Compose	Démarrage coordonné des conteneurs
Déploiement	GitLab CI/CD	Lancement automatique des tests puis déploiement du projet sur le serveur cible

# 9. Sécurité et qualité logicielle

## Sécurité dans l'application



- AUTHENTIFICATION DES UTILISATEURS
- GESTION DES RÔLES ET PERMISSIONS
  - PROTECTION DES MOTS DE PASSE
- FILTRAGE DES ACCÈS AUX DONNÉES SENSIBLES

## Sécurité du développement



- GESTION DES SECRETS
- VALIDATION DU CODE
- RESPECT DES BONNES PRATIQUES
- CORRECTION DES FAILLES

# 10. Solution pour les tests

Garantir la qualité du code, la stabilité des fonctionnalités et la fiabilité de l'application  
Symfony avant le déploiement.

Outil / Technologie	Rôle principal	Utilisation dans le projet
PHPUnit	Framework de tests unitaires et fonctionnels	Permet de tester les classes, services et entités Symfony de façon isolée
Symfony Test Client	Simulateur de requêtes HTTP	Vérifie le comportement des routes, contrôleurs et formulaires
Docker	Conteneur d'exécution isolé	Fournit un environnement de test identique à la production pour éviter les différences de configuration



# Type de tests prévus

Type de test	Description	Objectif
Unitaires	Test des fonctions et classes PHP	Vérifier que chaque composant fonctionne isolément
Intégration	est des interactions entre les modules	Assurer la cohérence des échanges
Fonctionnels	Simulation de navigation dans l'application via le client Symfony	Vérifier le bon comportement global
Test Utilisateurs	Phase de recette interne avec scénarios d'usage réels	Validation finale avant mise en ligne

# 11. Présentation de l'équipe

Élément	Description
Membres de l'équipe	ALGUL Sefer – GRANDGIRARD Louis Etudiants BUT Informatique 3
Organisation	Travail réalisé ensemble sur le même sujet, avec une coordination continue.
Communication	Échanges réguliers pour avancer ensemble et valider les décisions techniques.
Outils collaboratifs	Utilisation de GitLab pour le dépôt de code source et le suivi des modifications.
Répartition des tâches	Répartition flexible selon les besoins
Objectif commun	Produire une application fonctionnelle, claire et de qualité.

## 12. Conclusion



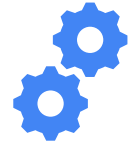
Résumé des choix  
techniques cohérents



Le cœur du projet est  
Symfony



L'intégration de la carte  
et du calendrier + base  
de données robuste



Solution évolutive et  
réutilisable