

Cahier des Charges

Application de Gestion d'Événements

Période de réalisation : 1er au 15 octobre 2025

Version : 1.0

Date : Octobre 2025

Sommaire

I : Présentation d'ensemble du projet

1. Présentation du contexte
2. Les objectifs de l'application
3. La cible adressée par le site
4. Objectifs quantitatifs
5. Périmètre du projet

II : Description graphique et ergonomique

1. Charte graphique

III : Description fonctionnement et technique

1. Description fonctionnelle du site
2. Description fonctionnelle du Back-office
3. Informations relatives aux contenus
4. Contraintes techniques

IV : Prestations attendues

1. Prestations attendues
2. Planning
3. Méthodologie de suivi
4. Propriété

V : Conclusion

I : Présentation d'ensemble du projet

Présentation du contexte

Le projet consiste à développer une application web dédiée à la gestion complète d'événements. L'objectif est de simplifier la création, la consultation, la modification et la suppression d'événements, tout en permettant aux utilisateurs de s'inscrire et de participer aux événements proposés.

Cette plateforme en ligne est conçue pour aider les organisateurs et les participants à gérer efficacement leurs événements. Les administrateurs peuvent créer et gérer des événements, tandis que les utilisateurs peuvent découvrir, consulter et s'inscrire aux événements qui les intéressent.

L'application intègre un système d'authentification robuste avec gestion des rôles (administrateur et utilisateur) pour garantir une expérience sécurisée et personnalisée.

Objectifs de l'application

- Créer une plateforme web conviviale pour gérer efficacement le cycle de vie des événements, de leur création à leur archivage.
- Fournir aux administrateurs des outils complets pour créer, modifier et supprimer des événements avec toutes les informations nécessaires (titre, description, date, heure, lieu, catégorie).
- Offrir une interface intuitive pour que les utilisateurs puissent consulter les événements disponibles, voir les détails complets et s'inscrire facilement.
- Permettre aux utilisateurs de gérer leurs inscriptions (inscription et désinscription) et de voir la liste des participants.
- Implémenter un système de rôles efficace distinguant les administrateurs (gestion complète) des utilisateurs standards (consultation et inscription).
- Assurer la sécurité des données et l'authentification des utilisateurs avec un système de connexion robuste.
- Fournir une expérience utilisateur fluide avec une interface responsive et accessible depuis tous les appareils.

Les cibles adressées par le site

L'application s'adresse à plusieurs catégories d'utilisateurs :

- **Les organisateurs d'événements** : Personnes ou organisations ayant besoin

de créer et gérer des événements (conférences, ateliers, formations, événements culturels, etc.). L'interface d'administration permet une gestion efficace.

- **Les participants potentiels** : Personnes cherchant à découvrir et participer à des événements. L'application offre une vue claire des événements à venir avec une inscription simplifiée.
- **Les associations et entreprises** : Structures organisant régulièrement des événements et nécessitant un outil centralisé pour gérer leurs activités et leur communauté.
- **Les étudiants et professionnels** : Personnes cherchant à participer à des événements de développement professionnel, de formation ou de networking.
- **Les passionnés d'événements culturels** : Individus souhaitant découvrir et participer à des événements variés (concerts, expositions, conférences, etc.).

Objectifs quantitatifs

Le projet vise à atteindre les objectifs quantitatifs suivants :

- Atteindre un minimum de 100 utilisateurs actifs d'ici le 15 octobre.
- Proposer une base de données comprenant au moins 20 événements variés pour les démonstrations et tests d'ici la fin du projet.
- Assurer un taux de disponibilité du site de 99% pendant la phase de développement pour garantir une expérience utilisateur fluide.
- Réduire le temps de chargement des pages à moins de 2 secondes pour une navigation rapide.
- Implémenter 100% des fonctionnalités CRUD pour les événements et les inscriptions.
- Garantir un système d'authentification sécurisé avec hashage des mots de passe et gestion appropriée des sessions.

Périmètre du projet

L'application est développée dans un cadre pédagogique avec les contraintes suivantes :

- Disponibilité en français uniquement.
- Responsive Design pour s'adapter aux différents appareils (desktop, tablette, mobile).
- Durée de développement limitée à 2 semaines (1er au 15 octobre 2025).
- Concentration sur les fonctionnalités essentielles (CRUD événements, authentification, inscriptions).



II : Description graphique et ergonomique

Charte graphique

Couleurs

Palette de couleurs professionnelle et moderne :

- Couleur principale : #2C3E50 (bleu foncé)
- Couleur secondaire : #3498DB (bleu clair)
- Couleur d'accentuation : #E74C3C (rouge/orange)
- Couleur de fond : #ECF0F1 (gris très clair)
- Couleur de texte : #2C3E50 (bleu foncé)

Police

- Police principale : Roboto ou similaire (sans-serif, moderne, lisible)

Inspirations

- Interface épurée et professionnelle
- Design moderne type Material Design
- Navigation claire et intuitive
- Mise en page aérée avec espaces blancs

Logo

- Logo à définir avec Figma, intégrant le concept d'événements et de communauté.

Ergonomie

- Navigation principale fixe en haut de page
- Boutons d'action clairement visibles
- Formulaires simples et intuitifs
- Messages de confirmation et d'erreur visibles
- Fil d'Ariane pour faciliter la navigation
- Design responsive pour tous les écrans

III : Description fonctionnement et technique

Description fonctionnelle du site

Spécifications générales

Page d'accueil :

Présente la liste des événements à venir et donne accès à toutes les fonctionnalités principales selon le rôle de l'utilisateur.

Aide à la navigation :

Utilisation d'un fil d'Ariane et d'une navigation claire pour faciliter le parcours utilisateur.

Sécurité des données :

Protection des informations personnelles des utilisateurs via hashage des mots de passe, protection CSRF et sessions sécurisées.

Interaction utilisateur :

Conception pour une expérience fluide et intuitive, avec une architecture MVC claire et une séparation des responsabilités.

Accessibilité :

Accessible depuis n'importe quel appareil avec une connexion Internet.

Spécifications fonctionnelles

Création de compte :

Les utilisateurs peuvent créer un compte avec un email et un mot de passe.
Attribution automatique du rôle ROLE_USER.

Authentification :

Connexion et déconnexion sécurisées, avec persistance de la session utilisateur et protection des routes selon les rôles.

Liste des événements (Public) :

Consultation par tous (connectés ou non) des événements à venir avec titre, date et lieu.

Détail d'un événement (Public) :

Affichage complet : titre, description, date/heure, lieu, catégorie, liste des participants inscrits.

Création d'événements (Admin uniquement) :

Formulaire pour créer des événements avec titre, description, date/heure, lieu, catégorie.

Modification d'événements (Admin uniquement) :

Formulaire pré-rempli pour modifier les événements existants.

Suppression d'événements (Admin uniquement) :

Suppression avec confirmation, en cascade (supprime les inscriptions associées).

Inscription à un événement (Utilisateurs connectés) :

Inscription par les utilisateurs authentifiés, avec vérification des doublons.

Désinscription (Utilisateurs connectés) :

Annulation de la participation à un événement.

Affichage des participants :

Nombre ou liste des inscrits sur la page de détail de l'événement.

Recherche et filtrage (Bonus optionnel) :

Recherche par mots-clés ou filtres par date, lieu ou catégorie.

Description fonctionnelle du Back-office

Le back-office est intégré à l'application via des routes protégées par le rôle ROLE_ADMIN.

Gestion des événements :

Interface pour visualiser tous les événements (passés et à venir), avec accès rapide à la création, modification et suppression.

Gestion des inscriptions :

Vue d'ensemble des inscriptions par événement, avec détails des participants.

Tableau de bord (optionnel) :

Statistiques sur le nombre d'événements, inscriptions et popularité.

Gestion des utilisateurs (optionnel) :

Liste des utilisateurs, modification des rôles, désactivation des comptes.

Informations relatives aux contenus

Événements :

Contenu créé par les administrateurs via l'interface :

- Titre (obligatoire, max 255 caractères)
- Description (obligatoire, texte long)
- Date et heure (obligatoire)
- Lieu (obligatoire)
- Catégorie (optionnelle)
- Createur (référence à l'administrateur)

Comptes utilisateurs :

Créés via le formulaire d'inscription :

- Email (unique, obligatoire)
- Mot de passe (hashé, obligatoire)
- Rôles (automatique : ROLE_USER par défaut)

Inscriptions :

Créées automatiquement lors de l'inscription à un événement :

- Référence à l'utilisateur
- Référence à l'événement
- Date d'inscription
- Contrainte d'unicité (un utilisateur par événement)

Données de test :

Fixtures pour démonstrations :

- 1-2 comptes administrateur
- 3-5 comptes utilisateur test
- 10-20 événements variés
- Inscriptions de démonstration

Contraintes Techniques

Stack technologique :

- Framework : Symfony 5, 6 ou 7
- Architecture : MVC stricte avec entités Doctrine, contrôleurs et templates Twig
- Base de données : MySQL ou MariaDB
- ORM : Doctrine
- Formulaires : Symfony Form Component
- Authentification : Symfony Security Bundle
- Gestion de version : Git avec hébergement sur GitHub

Base de données :

Tables :

Table user :

- id (int, PK, auto-increment)
- email (string(180), unique, not null)
- password (string(255), hashed, not null)
- roles (json, not null, default ["ROLE_USER"])

- created_at (datetime)
- created_at (datetime)

Table event :

- id (int, PK, auto-increment)
- title (string(255), not null)
- description (text, not null)
- date_time (datetime, not null)
- adress (string(255), not null)
- postal_code (int, not null)
- city (string(255), not null)
- country (string(255), not null)
- category (string(100), nullable)
- created_by_id (FK → users.id)
- created_at (datetime)
- updated_at (datetime)

Table registration :

- id (int, PK, auto-increment)
- user_id (FK → users.id, not null)
- event_id (FK → events.id, not null)
- registered_at (datetime, not null)
- UNIQUE constraint (user_id, event_id)

Authentification et sécurité :

- Authentification avec Form Login Authenticator
- Hashage des mots de passe (bcrypt/argon2)
- Protection CSRF sur les formulaires
- Protection XSS via Twig (échappement automatique)
- Gestion des rôles avec @IsGranted ou access_control
- Sessions sécurisées avec cookies HttpOnly

Gestion de version :

- Repository GitHub dédié
- Commits réguliers (minimum quotidiens)
- Messages de commit clairs en français ou anglais
- README.md complet avec instructions d'installation
- Fichier .gitignore approprié

Outils de conception :

- Figma pour les maquettes UI/UX
- Maquettes obligatoires pour les pages principales
- Design responsive

Gestion de projet :

- Trello avec méthode Kanban
 - Listes : À faire, En cours, Terminé
 - Mise à jour quotidienne
-

IV : Prestations attendues

Prestations attendues

Développement de l'application web :

- Mise en place de l'architecture Symfony (MVC)
- Création des entités Doctrine et migrations
- Développement des contrôleurs et actions
- Création des templates Twig
- Implémentation du système d'authentification
- Développement des fonctionnalités CRUD
- Intégration du CSS et responsive design

Conception de l'interface utilisateur :

- Création des maquettes Figma :
 - Page d'accueil / Liste des événements
 - Page de détail d'un événement
 - Formulaire de création/édition d'événement
 - Pages d'inscription et de connexion
 - Interface responsive (desktop et mobile)

Base de données et gestion des données :

- Modélisation de la base de données (diagramme de classes)
- Création des entités Doctrine avec relations
- Génération et exécution des migrations
- Création de fixtures pour données de test (optionnel)

Fonctionnalités principales :

- Système d'authentification (register/login/logout)
- CRUD complet pour les événements (admin)
- Système d'inscriptions aux événements
- Affichage de la liste des participants
- Gestion des rôles et autorisations
- Messages flash pour confirmations/erreurs
- Recherche/filtrage (optionnel, bonus)

Tests et qualité :

- Tests manuels de toutes les fonctionnalités
- Vérification de la sécurité (accès non autorisés, injections, etc.)
- Tests du responsive design
- Validation des formulaires
- Gestion des erreurs

Documentation :

- README.md complet sur GitHub
- Instructions d'installation détaillées
- Comptes de test fournis
- Documentation des choix techniques (optionnel)
- Commentaires dans le code si nécessaire

Exigences en matière de qualité

Fiabilité :

- Application stable, sans erreurs critiques
- Toutes les fonctionnalités demandées fonctionnent correctement

Performance :

- Temps de chargement < 2 secondes
- Requêtes à la base de données optimisées

Sécurité :

- Hashage approprié des mots de passe
- Protection contre CSRF, XSS, injection SQL
- Gestion sécurisée des sessions
- Contrôle d'accès strict selon les rôles

Accessibilité :

- Structure HTML sémantique
- Contraste de couleurs approprié
- Navigation au clavier possible
- Messages d'erreur clairs

Conformité :

- Conventions de code Symfony et PSR
- Bonnes pratiques de développement web
- Architecture MVC
- Principes de sécurité de base

Documentation :

- README clair avec instructions d'installation
- Commentaires pertinents dans le code
- Comptes de test fournis

Planning

Le projet sera réalisé en 10 jours de développement effectifs :

Étape 1 : Lancement et Conception (Jours 1-2)

Jour 1 (1 octobre) :

- Prise en main du sujet et clarification des exigences
- Création du tableau Trello avec toutes les tâches
- Création du projet Figma
- Début des maquettes (page d'accueil, liste événements)
- Réflexion sur l'architecture de la base de données

Jour 2 (2 octobre) :

- Finalisation des maquettes Figma (toutes les pages principales)
- Validation des maquettes
- Affinement du planning Trello
- Début de la conception de la base de données (diagramme)

Étape 2 : Setup et Base de Données (Jour 3)

Jour 3 (3 octobre) :

- Initialisation du projet Symfony
- Configuration de la base de données (.env)
- Création du repository GitHub et premier commit
- Création des entités (User, Event, Registration)
- Définition des relations entre entités
- Génération et exécution des migrations
- Vérification de la structure en base de données

Étape 3 : Authentification (Jour 4)

Jour 4 (6 octobre) :

- Implémentation de l'entité User (UserInterface)
- Création du système d'inscription (make:registration-form)
- Création du système de connexion (make:auth)
- Configuration de security.yaml
- Personnalisation des templates login/register
- Tests de l'authentification
- Premiers commits réguliers

Étape 4 : CRUD Événements (Jours 5-6)

Jour 5 (7 octobre) :

- Création du contrôleur EventController
- Implémentation de la liste des événements (index)
- Implémentation du détail d'un événement (show)
- Création du formulaire EventType
- Implémentation de la création d'événement (new) - Admin
- Tests et ajustements

Jour 6 (8 octobre) :

- Implémentation de la modification d'événement (edit) - Admin
- Implémentation de la suppression d'événement (delete) - Admin
- Configuration des restrictions d'accès (IsGranted)
- Création des templates Twig pour toutes les vues
- Tests complets du CRUD

Étape 5 : Système d'Inscriptions (Jour 7)

Jour 7 (9 octobre) :

- Implémentation de l'inscription à un événement (register)

- Implémentation de la désinscription (unregister)
- Vérification des doublons
- Affichage des participants sur la page de détail
- Gestion des messages flash
- Tests du système d'inscriptions

Étape 6 : Frontend et Finitions (Jours 8-9)

Jour 8 (10 octobre) :

- Intégration CSS sur toutes les pages
- Mise en conformité avec les maquettes Figma
- Implémentation du responsive design
- Amélioration de l'UX (messages, navigation)
- Personnalisation des messages de validation
- Tests sur différents navigateurs et résolutions

Jour 9 (13 octobre) :

- Tests complets de l'application
- Correction des bugs identifiés
- Création de fixtures pour les données de test
- Finalisation du README.md
- Derniers ajustements
- Bonus si temps disponible (recherche/filtres, amélioration du design)

Étape 7 : Préparation Soutenance (Jours 10-11)

Jour 10 (14 octobre) :

- Tests finaux exhaustifs
- Vérification de tous les livrables
- Mise à jour finale du Trello
- Export/captures du Figma
- Rédaction/finalisation du README
- Préparation des supports de présentation
- Répétition de la démonstration

Jour 11 (15 octobre) :

- **SOUTENANCE FINALE**
- Présentation de 10 minutes
- Démonstration en direct
- Questions/réponses (5 minutes)

Équipe de développement

Projet individuel réalisé par un étudiant développeur.

Temps de travail : 10 jours de développement effectifs sur 2 semaines.

Budget

Projet pédagogique sans budget financier associé. Investissement en temps :

Temps de développement estimé :

- Conception et maquettage : 2 jours
 - Setup et base de données : 1 jour
 - Authentification : 1 jour
 - CRUD événements : 2 jours
 - Système d'inscriptions : 1 jour
 - Frontend et intégration : 2 jours
 - Tests et préparation soutenance : 1 jour
- Total : 10 jours de développement**

Outils utilisés (gratuits) :

- Symfony CLI
- Composer
- PHP / MySQL
- VS Code ou IDE similaire
- Git / GitHub
- Figma (version gratuite)
- Trello (version gratuite)

Méthodologie de suivi

Suivi du projet via la méthode Kanban avec Trello.

Principes de la méthode Kanban

Tableau Trello avec 4 listes principales :

- Backlog / À faire
- En cours
- En test / Revue
- Terminé

Découpage en tâches :

Chaque fonctionnalité décomposée en tâches atomiques. Exemples :

- "Créer l'entité Event"
- "Implémenter la route index pour lister les événements"
- "Créer le template event/index.html.twig"
- "Ajouter le CSS pour la liste des événements"
- "Tester l'affichage de la liste des événements"

Workflow :

1. Planification initiale : tâches dans "À faire"
2. Travail : déplacer dans "En cours"
3. Codée : déplacer dans "En test / Revue"
4. Validée : déplacer dans "Terminé"

Mise à jour quotidienne :

Mise à jour quotidienne du tableau Trello pour refléter l'avancement.

Labels Trello utilisés :

- Priorité Haute (rouge)
- Priorité Moyenne (orange)
- Priorité Basse (jaune)
- Bug (rouge foncé)
- Documentation (bleu)
- Bonus / Optionnel (vert)

Commits Git réguliers

- Minimum 1 commit par jour, idéalement plusieurs
- Messages de commit clairs et descriptifs
- Commits atomiques (une fonctionnalité = un commit)

Exemples de messages :

- "feat: Add User entity and authentication system"
- "feat: Implement event CRUD operations"
- "fix: Correct registration duplicate check"
- "style: Add responsive CSS for event list"
- "docs: Update README with installation instructions"

Plan de gestion de projet

Initiation

- Définir les objectifs : application fonctionnelle de gestion d'événements
- Identifier les parties prenantes : étudiant développeur, enseignant évaluateur
- Évaluer les ressources : 10 jours, outils gratuits
- Développer le plan : planning jour par jour avec jalons

Planification

- Définir les tâches et délais (planning ci-dessus)
- Établir les critères de réussite : fonctionnalités obligatoires implémentées
- Plan de communication : mise à jour Trello, commits Git réguliers
- Plan de gestion des risques : prévoir du temps buffer, prioriser les fonctionnalités essentielles

Exécution

- Commencer le développement selon le planning
- Mettre en œuvre les bonnes pratiques : conventions de code, MVC, sécurité
- Surveiller les progrès via Trello
- S'adapter si nécessaire

Contrôle

- Comparer les résultats aux critères de réussite
- Identifier et résoudre rapidement les problèmes
- Tester régulièrement pour assurer la qualité
- Demander de l'aide à l'enseignant si besoin

Clôture

- Vérifier que tous les objectifs sont atteints
- Vérifier les livrables (GitHub, Trello, Figma, README)
- Préparer la soutenance finale
- Documenter les leçons apprises

Propriété

Projet développé dans un cadre pédagogique par l'étudiant.

- Tous les droits de propriété intellectuelle (code source, conceptions, contenus) appartiennent à l'étudiant développeur.
- Code source hébergé sur GitHub, public ou privé selon les préférences de l'étudiant.
- L'établissement scolaire peut utiliser le projet à des fins pédagogiques et d'évaluation uniquement.

V : Conclusion

Ce cahier des charges offre une vision détaillée des objectifs, fonctionnalités et contraintes du projet d'application de gestion d'événements.

L'application vise à fournir une plateforme fonctionnelle et intuitive pour la création, la consultation et la gestion d'événements. Les administrateurs pourront gérer efficacement leurs événements, tandis que les utilisateurs pourront découvrir, consulter et s'inscrire facilement.

Les fonctionnalités incluent une authentification sécurisée, un CRUD complet pour les événements, un système d'inscriptions, une gestion des rôles et une interface responsive.

L'étudiant développeur s'engage à respecter les délais (2 semaines), les exigences techniques (Symfony, MySQL, Git) et à fournir un travail de qualité professionnelle.

Ce projet représente une opportunité d'apprentissage pour consolider les compétences en développement web avec Symfony, en gestion de projet et en conception d'applications complètes.

Ce cahier des charges est un document de référence pour toute la durée du projet.

Tout changement important aux spécifications devra être validé par l'enseignant responsable.

Document établi le : 1er octobre 2025

Version : 1.0

Projet : Application de Gestion d'Événements - Symfony

Durée : 15 jours (1er au 15 octobre 2025)