




# PHP : Event Manager v1




Cahier des charges technique — Plateforme de gestion d'événements

## Cahier des charges technique — Plateforme de gestion d'événements

### Objectifs du projet




Ce projet a pour but de développer une application web modulaire et complète en PHP/MySQL, permettant la gestion intégrale d'événements, depuis leur création jusqu'à la facturation, en passant par la billetterie, la communication et le suivi en temps réel. Il est conçu pour être réalisé dans un contexte d'équipe agile, avec une répartition flexible des tâches, en favorisant la collaboration inter-étudiante et les itérations rapides. Le projet s'adresse à des étudiants en informatique spécialisés dans le développement web full-stack.   

### Compétences mobilisées

- Conception d'architecture logicielle MVC orientée objet avec PHP 8+
- Modélisation et manipulation avancée d'une base de données relationnelle (MySQL)
- Développement frontend responsive en HTML/CSS/JavaScript (framework CSS recommandé)
- Intégration de services tiers via des API REST (paiement, IA, QR code)
- Conteneurisation avec Docker et orchestration avec Docker Compose
- Maîtrise du cycle de vie Git/GitHub : gestion des branches, pull requests, intégration continue
- Mise en œuvre de la méthodologie agile (scrum, backlog, daily, sprint review)
- Test unitaire (PHPUnit), validation de formulaire, sécurité des entrées (sanitizing)   

### Spécifications fonctionnelles

### Utilisateurs et gestion des comptes

- Interface d'inscription avec confirmation par e-mail et protection anti-bot (captcha ou honeypot)
- Authentification sécurisée (hashage des mots de passe, sessions)
- Réinitialisation de mot de passe via token sécurisé
- Gestion de rôles : organisateur, visiteur, administrateur
- Tableau de bord personnel avec vue contextuelle (selon le rôle)
- Édition du profil utilisateur : prénom, nom, email, photo, mot de passe   

### Module événementiel

#### Présentation générale

Le module événementiel constitue le cœur de la plateforme. Il permet aux organisateurs de créer, gérer et publier des événements de manière structurée et intuitive. Chaque événement peut comporter de nombreux détails, associés à des entités complémentaires telles que les prestataires, les catégories, les intervenants ou les coûts. Il vise à offrir une expérience fluide pour l'édition et une présentation claire pour les visiteurs.

#### Création d'un événement

L'interface de création repose sur un formulaire guidé, organisé en plusieurs sections. Elle inclut :

- Un titre (unique) et une description riche à l'aide d'un éditeur WYSIWYG (mise en forme, images, liens...)
- Le téléchargement d'une ou plusieurs images d'illustration (avec redimensionnement)
- La sélection des horaires : date de début, de fin, fuseau horaire
- La localisation via saisie assistée (autocomplétion d'adresse + carte Google Maps)

## Associations complémentaires

Chaque événement peut être enrichi par :

- Une ou plusieurs **catégories** (concert, festival, atelier...)
- Des **étiquettes** personnalisées (ex. « plein air », « gratuit », « VIP »)
- La désignation d'**organiseurs secondaires** avec droits partagés
- L'ajout de **prestataires** liés à l'événement : leurs prestations et coûts sont visibles dans une vue dédiée
- L'encodage d'un **budget prévisionnel** (frais généraux, artistes, logistique...)

## Duplication

Il est possible de dupliquer un événement existant :

- Tous les champs sont préremplis dans un nouveau brouillon
- Les éléments sensibles (statuts, billets, ventes) ne sont pas répliqués

## Affichage public

Chaque événement possède :

- Une URL propre de type `evenements/slug-titre`
- Une page publique stylisée affichant le contenu, les horaires, la carte, les artistes
- Un bouton d'accès à la billetterie (le cas échéant)

## Gestion des statuts

Pour refléter le cycle de vie d'un événement, plusieurs statuts sont utilisés :

- **Brouillon** : non publié, visible uniquement par l'équipe organisatrice
- **Planifié** : visible, mais la billetterie n'est pas encore active
- **Publié** : accessible au public avec billetterie ouverte
- **Annulé** : signalé comme tel avec message explicite
- **Archivé** : consultable en lecture seule après l'événement

## Commentaires internes

Un système de **commentaires privés** est disponible pour faciliter la coordination de l'équipe. On peut y :

- Ajouter des notes ou questions internes
- Mentionner un autre membre de l'équipe (ex. @prenom)
- Associer un commentaire à une section (prestataire, planning, etc.)

Ce module assure ainsi une gestion complète, modulaire et collaborative de chaque événement. 🗨️ 📅 🏠

- Création d'événement avec saisie guidée : titre, description riche (éditeur WYSIWYG), image, horaires, lieu (avec intégration Google Maps)
- Association de catégories, étiquettes, organisateurs secondaires
- Possibilité de dupliquer un événement existant
- Affichage public avec URL propre
- Statuts multiples : brouillon, planifié, publié, annulé, archivé
- Système de commentaires internes (notes d'organisation) 🗨️ 📅 🏠

## Billetterie en ligne

## Présentation générale

La billetterie constitue une fonctionnalité centrale de la plateforme pour les événements publics. Elle permet aux visiteurs d'acheter des billets via une interface intuitive, tout en offrant aux organisateurs un contrôle total sur les ventes. Chaque événement peut proposer plusieurs types de billets, avec gestion des quotas et options tarifaires.



## Création des types de billets

Pour chaque événement, les organisateurs peuvent :

- Définir des **types de billets** : normal, VIP, gratuit, accès backstage, etc.
- Spécifier un **prix**, une **quantité disponible**, une **durée de mise en vente**
- Indiquer des **conditions d'utilisation** (limite d'âge, accès restreint)

## Interface d'achat

L'interface de vente est publique et permet :

- De **visualiser les types de billets disponibles**, leurs prix, leur disponibilité
- D'ajouter un ou plusieurs billets au **panier**
- De procéder à un **paiement simulé** via une API REST fictive

## Génération de billet avec QR code

Après validation du paiement, le système :

- Génère un **fichier PDF personnalisé** pour chaque billet
- Attribue un **QR code unique** lié au billet dans la base de données
- Envoie automatiquement le billet par **e-mail** au participant

## Suivi des ventes


L'organisateur peut consulter en temps réel :


- Le **nombre de billets vendus** par type
- Le **revenu généré** (simulé)
- Le **statut de chaque commande** (payé, en attente, échec)

## Gestion des remboursements

Un module de remboursement manuel permet :

- L'annulation d'un billet et sa désactivation
- La génération d'un **avoir** ou d'une **note de crédit** (facultatif)

Ce module assure une gestion fluide et sécurisée de la vente de billets, tout en restant connecté aux autres briques de la plateforme (événements, scan, statistiques). 

- Définition de types de billets : gratuit, tarif plein, étudiant, VIP, accès backstage, etc.
- Gestion fine des quotas, dates de mise en vente, prix et conditions
- Interface publique intuitive pour la réservation, avec panier et paiement simulé
- Génération automatique d'un PDF pour chaque billet, intégrant un QR code unique
- Envoi automatisé des billets par e-mail après validation du paiement
- Système de gestion des remboursements (manuel ou simulé) 

## Application mobile pour le scan des billets



### Présentation générale

Cette application mobile sous forme de **Progressive Web App (PWA)** permet aux organisateurs et agents d'accueil de **scanner les billets** présentés par les visiteurs lors d'un événement. Elle offre une solution légère, fonctionnelle et connectée en temps réel à la plateforme principale. 📱 📷 ✅

## Fonctionnalités clés

- Interface responsive adaptée aux smartphones
- Authentification sécurisée des agents de scan
- Scan via **caméra du navigateur mobile** (API HTML5)
- Détection automatique du **QR code** présent sur les billets PDF
- **Vérification immédiate** de la validité du billet :
  - Billet valide → confirmation visuelle et sonore
  - Billet déjà scanné → message d'alerte
  - Billet inexistant ou annulé → rejet explicite

## Sécurité et performance

- Accès restreint via **token temporaire ou session authentifiée**
- Temps de réponse optimisé via appel API REST au serveur
- Journalisation des scans avec timestamp, agent, et statut
- Mode hors-ligne (optionnel) avec synchronisation différée en cas de coupure réseau

## Suivi en temps réel

- Affichage en direct du **nombre de billets scannés**
- Statistiques d'entrée par tranche horaire ou point d'accès
- Historique filtrable par événement, date ou agent

## Intégration avec les autres modules

- Connexion directe à la base de données de la billetterie
- Synchronisation avec les statuts de vente, d'annulation ou de remboursement
- Les billets scannés deviennent automatiquement **inactifs** pour éviter les fraudes

Ce module garantit une gestion fluide, sécurisée et moderne des entrées, en phase avec les standards du secteur événementiel. 📱 📷 ✅

- Web app de type PWA optimisée pour les appareils mobiles
- Authentification spécifique aux agents de scan
- Accès sécurisé via token JWT ou session temporaire
- Utilisation de l'API caméra HTML5 pour scanner les QR codes
- Vérification du billet en temps réel via une API REST
- Historique des scans, statistiques de passage, messages d'erreur explicites 📱 📷 ✅

## Gestion des factures 📄 📁 📌

- Création automatique de factures PDF conformes aux exigences légales
- Gestion des numérotations automatiques, dates, TVA, mentions obligatoires
- Téléchargement possible depuis l'espace utilisateur
- Registre de facturation consultable pour les administrateurs
- Possibilité d'ajout manuel de factures pour d'autres prestations (location de salle, etc.) 📄 📁 📌

## Paiement simulé 🏧 📱 🧱

- API REST fictive reproduisant les comportements d'une API réelle : `/checkout`, `/status`, `/webhook`
- Réponses aléatoires ou pilotées pour tester différents scénarios (succès, échec, en attente)
- Interface backend pour consulter les logs de transactions
- Possibilité de rejouer une transaction ou de simuler un remboursement
- Intégration de messages utilisateurs clairs pour les différents cas d'erreur 🏧 📱 🧱

## Gestion du line-up (programmation artistique) 🎵 🤔 📅

- Base de données d'intervenants (artistes, conférenciers, DJ, etc.)
- Affectation manuelle ou assistée à des créneaux horaires (avec conflit management)
- Visualisation graphique de la programmation sous forme de timeline interactive
- Éditeur de bio, genre musical, réseaux sociaux
- IA pour générer un line-up équilibré selon des critères définis (durée, diversité, répartition) 🎵 🤔 📅

## Calendrier interactif 📅 ↻ 📁

- Module central pour la planification des événements, tâches, prestations
- Vue mensuelle, hebdomadaire et journalière, avec filtres par type d'activité
- Ajout rapide via clic ou glisser-déposer
- Couleurs personnalisables selon la nature des éléments
- Possibilité de synchronisation avec des calendriers externes (Google, Outlook) 📅 ↻ 📁

## Prestataires et prestations 🤝 📄 🛠️

- Création de fiches prestataires avec coordonnées, domaines d'activité, statut
- Attribution à un événement avec description de la prestation, coûts, horaires
- Suivi de l'état de la prestation (à confirmer, confirmé, réalisé, annulé)
- Upload de documents (contrats, devis, fiches techniques)
- Intégration dans le calendrier et les factures 🤝 📄 🛠️

## Statistiques et rapports 📊 📈 📄

- Dashboard en temps réel : billets vendus, taux de remplissage, revenus estimés
- Graphiques interactifs (par jour, par type de billet, par canal de vente)
- Export des données au format CSV et PDF
- Statistiques post-événement (fréquentation réelle, analyse du scan, feedback utilisateurs) 📊 📈 📄

## Intégration d'une API IA 🤖 🧠 💬




- Générateur de description d'événement à partir de mots-clés
- Proposition automatique de titres d'événements ou hashtags promotionnels
- Recommandations d'intervenants/artistes basées sur le type d'événement
- Assistant conversationnel pour aider à la planification (tâches, messages automatiques, idées)
- Moteur d'analyse sémantique des avis post-événement 🤖 🧠 💬

## Technologies et architecture 🖥️ 🛠️ ⚙️

Domaine	Technologies retenues
Back-end	PHP 8+ (architecture MVC, PSR-4)
Base de données	MySQL 8.0
Frontend	HTML5, CSS3, JS (Tailwind ou Bootstrap)
Mobile	PWA, API Caméra HTML5
Environnement	Docker, Docker Compose, phpMyAdmin
Contrôle de version	Git + GitHub (Git flow, CI/CD avec GitHub Actions)
PDF/QR Code	mPDF, DomPDF, android/qr-code
API IA	OpenAI, simulateur local
Paiement simulé	API REST locale

## Structure projet recommandée 📁 📁 🧱




- Architecture MVC stricte, avec injection de dépendances si pertinent

- Arborescence claire : `controllers/`, `models/`, `views/`, `services/`, `api/`, `tests/`
- Fichiers de configuration et variables `.env` externalisées
- Utilisation de Composer pour autoloading et gestion des dépendances
- Scripts d'installation (`setup.sh`, `init.sql`, `seeders`)
- Mise en place d'un pipeline CI/CD simplifié pour tests automatiques   

## Interfaces attendues (exemples non exhaustifs)

Interface	Accès utilisateur
Authentification (login/signup)	Tous
Tableau de bord	Organisateur/Admin
Liste des événements	Organisateur
Création / édition d'événement	Organisateur
Vue publique événement	Visiteur
Billetterie	Visiteur
Panier et paiement simulé	Visiteur
Espace personnel + billets PDF	Visiteur
Interface de scan mobile	Organisateur
Génération et consultation des factures	Organisateur/Admin
Planning du line-up	Organisateur
Calendrier global	Organisateur/Admin
Liste des prestataires	Organisateur
Gestion des prestations	Organisateur
Rapports et statistiques	Organisateur/Admin
Tableau de configuration	Administrateur




## Bonnes pratiques recommandées

- Respect strict des conventions PSR (PSR-1, PSR-4, PSR-12)
- Architecture modulaire, découplée, testable
- Branches `feature/`, `bugfix/`, `release/`, avec PR systématiques
- Mise en place d'un linter (PHP-CS-Fixer) et analyse statique (PHPStan)
- Utilisation de PHPUnit avec couverture minimale visée (>80%)
- Documentation dans le code (PHPDoc) + Wiki GitHub ou `docs/`
- Intégration de jeux de données de développement (`faker`, `seeders`)   

## Extensions et évolutions possibles



- Interface multi-langue (FR/NL/EN) avec système de traductions
- Accès restreint par groupe d'utilisateurs ou codes d'invitation
- Gestion des bénévoles (planning, affectations, badges)
- Génération de plans de salle et numérotation des sièges
- Système de liste d'attente et relance automatique

- Notifications SMS ou WhatsApp via API Twilio ou simulateur
- Export vers des ERP ou CRM partenaires (optionnel)
- Compatibilité NFC / badge RFID pour contrôle d'accès en mode R&D   

---

Ce projet constitue un cas d'étude complet et réaliste de gestion d'application web événementielle. Il permet de mobiliser l'ensemble des compétences attendues d'un développeur full-stack moderne, depuis la modélisation jusqu'au déploiement, en passant par la sécurité, la scalabilité et l'expérience utilisateur. Il peut servir de base à des déploiements professionnels ou à la présentation dans un portfolio. 