

Fiche de Documentation Technique

– Projet Congrès ANALIM

Informations Générales

Nom du projet : Congrès ANALIM

Type : Application web de gestion d'événement

Technologie principale : PHP (Architecture MVC)

Public cible : Administrateurs du congrès, équipes de facturation, services comptables

Objectif du projet :

Développer une application web permettant la gestion efficace d'un congrès professionnel, en particulier la gestion :

- des **organismes payeurs** (entreprises, administrations...),
- des **règlements des participants** (factures, paiements...),
- des sessions et des activités liées au congrès (à intégrer).

Architecture du Projet

Modèle architectural : MVC (Model - View - Controller)

Description des couches :

- **Model** : Contient les classes métiers en PHP orienté objet (OrganismePayeurs, Facture, Congressiste, etc.). Ces classes assurent également l'accès aux données via des méthodes telles que :
 - getOrganismesPayeursParNom()
 - getLesFacturesRegler()
 - ajouterReglement()
 - **View** : Ensemble des fichiers PHP responsables de l'interface utilisateur :
 - Formulaires (saisie/modification d'organismes, paiement de factures)
 - Listes dynamiques (factures à régler, organismes existants)
 - **Controller** : Reçoit les actions des utilisateurs depuis les vues, applique la logique métier via les modèles, puis redirige l'utilisateur vers la vue appropriée.
-

Technologies et Outils

Élément	Détail
Langages	PHP (POO), HTML5, CSS3
Base de données	MySQL
Environnement local	XAMPP (Apache, MySQL, PHP)
Éditeur de code	Visual Studio Code
Versioning	Git (hébergé sur GitHub)

Fonctionnalités Clés

◆ Partie 1 : Gestion des Organismes Payeurs

- Création, modification et suppression via formulaire dédié
- Saisie des informations : nom, adresse, type d'organisme, contacts
- Recherche par nom pour retrouver un organisme rapidement
- Association des organismes aux factures des congressistes

◆ Partie 2 : Règlements des Factures

- Affichage des factures non réglées d'un congressiste
- Paiement d'une facture : saisie du montant, date, mode de paiement
- Possibilité de lier un **organisme payeur** à la facture (si applicable)
- Historique des règlements

Fonctionnalités complémentaires (à prévoir) :

- Génération de reçus PDF
 - Notifications internes ou par e-mail (paiement reçu)
 - Tableau de bord global des paiements
-

Sécurité et Droits d'Accès

- Authentification de base (à prévoir pour version complète)

- Rôles utilisateur recommandés :
 - **Administrateur** : accès complet (CRUD, paiements)
 - **Comptable** : accès en lecture + saisie des paiements
 - **Utilisateur standard** : lecture seule
-

Tests

- Tests manuels sur environnement local (XAMPP)
- Cas de test :
 - Création et suppression d'un organisme
 - Paiement correct d'une facture
 - Gestion des erreurs (facture inexistante, montant incorrect)

Recommandation : intégrer des **tests automatisés PHPUnit** sur les classes métier (Model), notamment la gestion de la base de données.

Déploiement

Environnement de déploiement Local (WampServer ou XAMPP)

Méthode recommandée Copie du projet dans htdocs + importation BDD

Configuration requise PHP ≥ 7.4, MySQL ≥ 5.7

Évolutions futures :

- Hébergement en ligne (via OVH, Ionos, etc.)
 - Sécurisation SSL
 - Interface responsive (intégration Bootstrap recommandée)
-

Annexes

- Diagramme de classes des entités principales
- Cas d'utilisation des utilisateurs
- Scripts SQL de création des tables
- Schéma de navigation (arborescence des vues)