Concours Robots

Documentation technique

Mission 9 - Auto-inscriptions avec modération

Mettre en place les fonctionnalités permettant à un gestionnaire de gérer les auto-inscriptions des abonnés (à modérer)

Auteur : Yanis PROUST

Date : Octobre 2025

Version 2

Table des matières

Introduction

Cette documentation technique concerne la **Mission 9 – Auto-inscriptions avec modération** du projet *Concours Robots*. Elle a pour objectif de décrire de manière précise la mise en place d'une fonctionnalité permettant aux visiteurs de s'inscrire sur la plateforme et de devenir abonnés, sous réserve d'une validation par un gestionnaire.

Le document présente :

- le **contexte technique** (technologies et architecture utilisées),
- le **fonctionnement détaillé** de l'auto-inscription,
- les **règles d'intégrité référentielle** appliquées à la base de données,
- la description des opérations **CRUD** liées à cette fonctionnalité,
- ainsi que les **tests fonctionnels et techniques** nécessaires pour valider la solution.

L'objectif est de fournir une documentation claire et exploitable par tout développeur ou membre de l'équipe projet, afin d'assurer la compréhension, la maintenance et l'évolutivité de cette fonctionnalité.

Contexte général

Le **Concours Robots** est un événement annuel organisé entre plusieurs **collèges privés des Deux-Sèvres**, dans le cadre de l'enseignement de technologie en classe de troisième. L'édition 2025 s'est tenue le **5 avril 2025** à la salle de basket de Valette et a réuni **six établissements**. Chaque collège pouvait inscrire plusieurs équipes de **quatre collégiens**, identifiées par un code unique.

Le concours comporte différentes **épreuves** (normales, bonus et site internet de présentation) notées selon des **barèmes et coefficients spécifiques**. Le **score global** d'une équipe est obtenu par la somme pondérée des notes obtenues à chaque épreuve. Des **épreuves bonus** servent à départager les ex æquo.

Le déroulement repose sur une forte **collaboration entre les élèves** :

- des **jurys** composés de collégiens notent les épreuves d'autres établissements ;
- des **secrétaires** saisissent les notes dans le système ;
- un **gestionnaire** supervise la saisie, corrige les erreurs éventuelles et édite les résultats finaux.

À l'issue du concours, les **résultats** sont publiés par catégories (classement général, esthétique, site internet, meilleure équipe par collège) et envoyés aux enseignants sous forme de **fichiers tableurs**.

Pour faciliter la gestion globale, une **application web** est développée avec le **framework PHP Laravel**. Elle doit permettre :

- l'inscription des équipes par les enseignants avant le concours ;
- la saisie et la consultation des résultats pendant l'événement ;
- l'accès public aux classements après le concours.

L'application vise à offrir une solution **simple, sécurisée et responsive**, accessible depuis ordinateur, tablette ou smartphone, et adaptée aux différents **rôles utilisateurs** : visiteurs, abonnés, élèves, enseignants, jurys, secrétaires, gestionnaires et administrateurs.

Contexte technique

Le projet **Concours Robots** a été développé en environnement web avec une architecture classique **client-serveur**.

Les principales technologies utilisées sont :

- **Langage back-end** : PHP
- **Framework** : Laravel *(structure MVC — Model, View, Controller)*
- **Base de données** : MariaDB
- **Front-end** : HTML5, CSS3, JavaScript
- **Serveur** : Apache

Architecture générale

- **Modèle (Model)** : gère la base de données *(tables users, utilisateurs, roles et leurs relations)*.
- **Vue (View)** : pages HTML/CSS du site *(formulaires, listes de comptes, messages d'erreur/validation)*.
- **Contrôleur (Controller)** : logique métier *(création de compte, validation par un gestionnaire, suppression en cascade)*.

Ce découpage MVC permet :

- une **séparation claire** entre la logique applicative et l'interface utilisateur,
- une **maintenance facilitée**,
- la **réutilisabilité du code** *(par exemple : les contrôleurs de gestion des comptes sont réutilisés pour la modération)*.

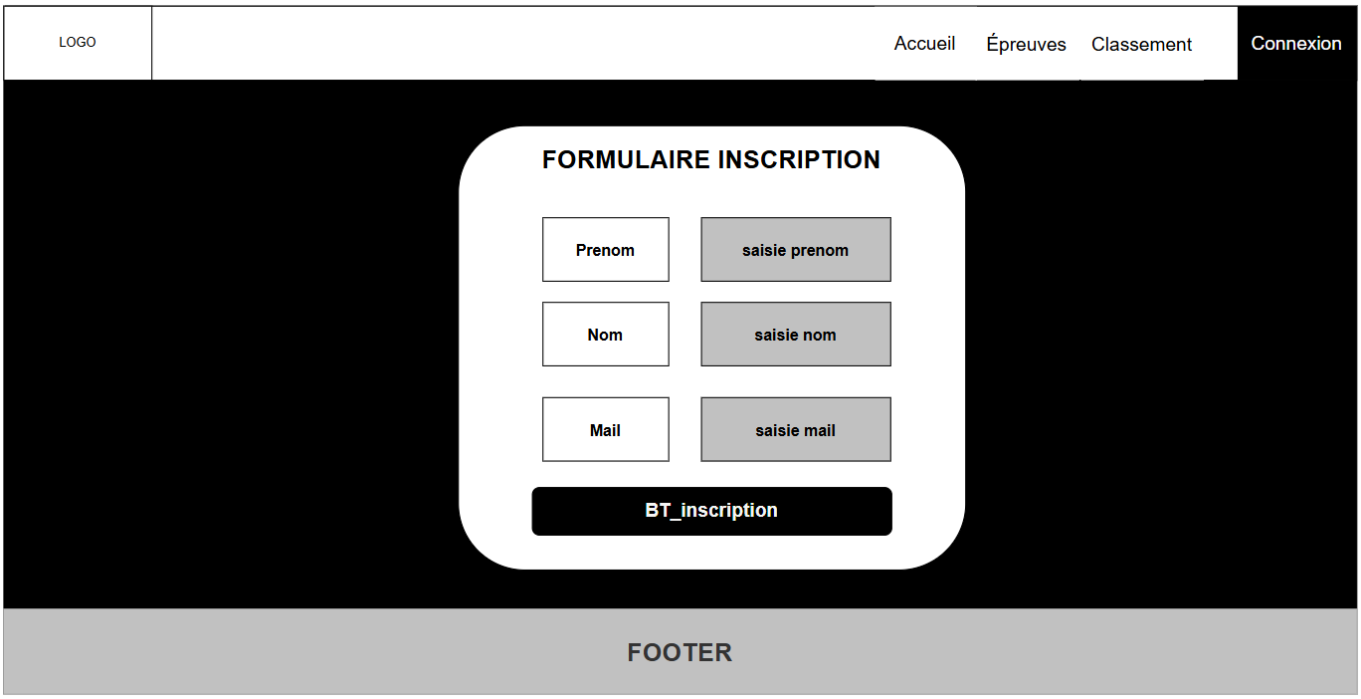
Pour rappel

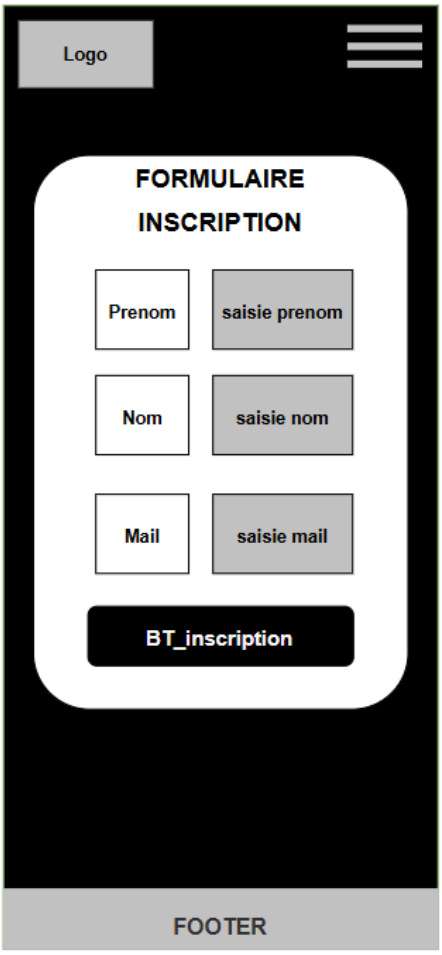
- Un **visiteur** est un internaute non authentifié, sans droit.
- Un **abonné** est un utilisateur avec droit de consultation uniquement.

Les **auto-inscriptions** permettent donc à un visiteur de demander l'obtention du rôle **abonné**.

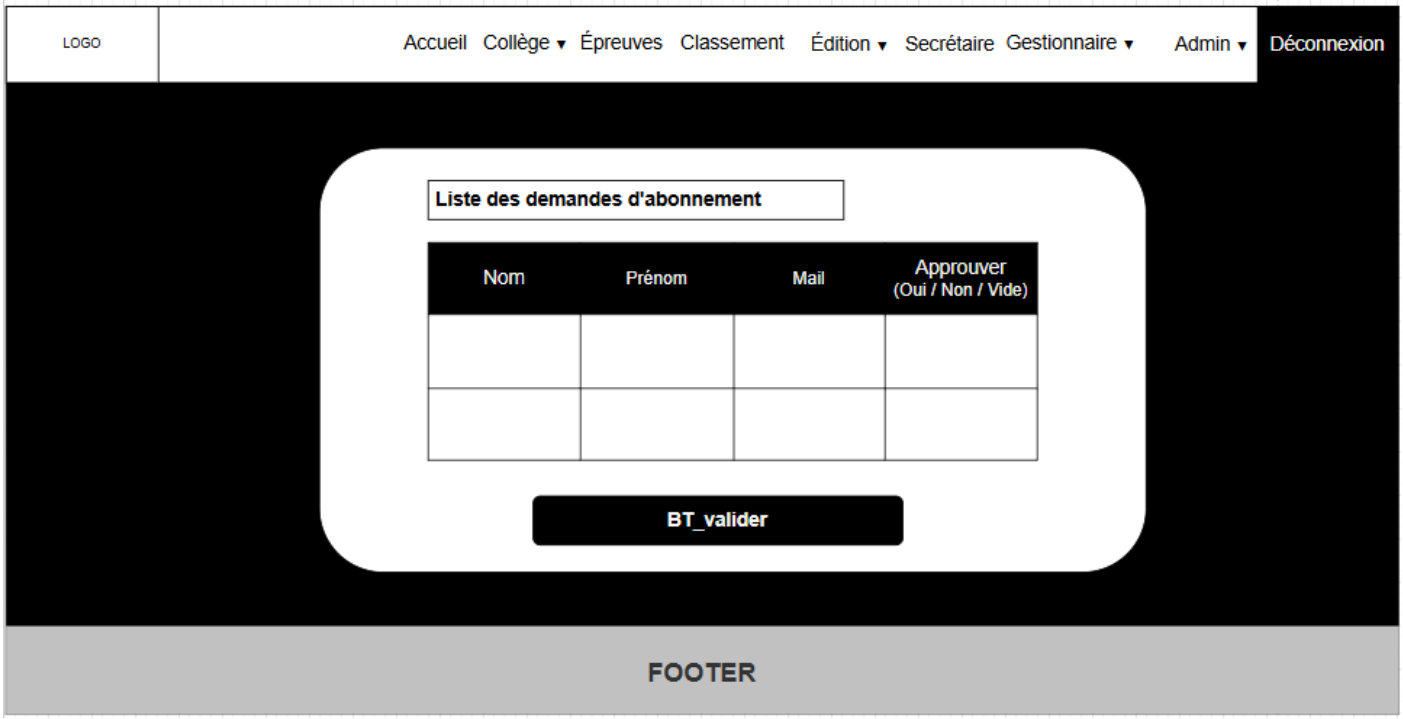
Maquettes

Maquette de la page inscription.php :





Maquette de la page abonnements.php :



Accès aux fonctionnalités de la mission

1. Formulaire d'inscription (inscription.php)

- Depuis la page d'accueil, l'utilisateur clique sur le bouton **Inscription**.
- Le formulaire d'inscription s'affiche et permet la saisie de l'**email**.
- Après validation du formulaire, la logique suivante est appliquée (AutoInscriptionController::store()) :

1. **Email inexistant en base**
   → Création d'un compte dans mcd_users (mot de passe temporaire généré automatiquement)
   → Envoi d'un email de vérification
   → L'utilisateur n'existe pas encore dans mcd_utilisateurs
2. **Email existant + email non vérifié**
   → Renvoi d'un email de vérification
   → Aucune nouvelle création
3. **Email existant + email vérifié + profil non créé**
   (l'utilisateur a confirmé son email mais n'a jamais complété son inscription)
   → Redirection vers la procédure de **mot de passe oublié**
4. **Email existant + email vérifié + profil déjà complet**
   → L'utilisateur est **redirigé vers la page de connexion**

À ce stade, si le compte est nouvellement créé :

✓ une entrée existe **seulement dans mcd_users**
✓ email_verified_at est encore **null**

2. Validation de l'email

• L'utilisateur clique sur le lien de vérification reçu par email.
• Laravel met alors à jour automatiquement le champ **email_verified_at** dans mcd_users.
• L'utilisateur est immédiatement authentifié et redirigé vers la page **/inscription**.

3. Inscription complète

Sur la page /inscription, l'utilisateur complète les informations suivantes :

- nom

- prénom
- genre
- mot de passe (définitif)

Lors de la validation (AutoInscriptionController::inscription()) :

Mise à jour de mcd_users

• le mot de passe temporaire est remplacé par celui choisi
• le champ name est mis à jour

Création du profil métier dans mcd_utilisateurs

Une ligne est insérée avec :

Champ	Valeur
id	identifiant du user
nom	saisi
prénom	saisi
code_genre	saisi
code_statut	**A** (En attente de validation par un gestionnaire)

À ce stade, l'utilisateur existe dans les deux tables mais **n'a aucun rôle**.

4. Gestion des abonnements (modération)

Accès réservé aux gestionnaires (rôle GST)

Depuis le menu Gestionnaire → Abonnement, l'interface affiche la liste de **toutes les demandes d'abonnement**, c'est-à-dire les utilisateurs dont code_statut = A.

Pour chaque demande, le gestionnaire peut choisir :

• **Attendre (A)**
→ aucun changement, l'utilisateur reste en attente

• **Approuver (N → Normal)**
→ actions effectuées :

- code_statut devient **N**
- un rôle **ABO** est attribué au compte
- l'utilisateur est **rattaché au concours en cours** via mcd_engager

• **Bloquer (B)**
→ l'utilisateur n'obtient **aucun rôle** et son statut devient **B**

La validation globale s'effectue via le bouton **Valider**.

5. Suppression définitive des demandes

Une seconde page est disponible : **gestion/supprimer_auto_abo**.

Elle affiche la même liste de demandes d'abonnement, mais ici le gestionnaire peut **supprimer définitivement un utilisateur**, ce qui entraîne la suppression des données dans mcd_users et dans mcd_utilisateurs.

Fonctionnement (CRUD)

Créer (C)

• Le visiteur saisit son email
• Selon les cas, un compte est créé ou réutilisé dans mcd_users
• Le profil n'est créé que lorsque l'utilisateur remplit son formulaire final
• Le statut initial du profil est **A**

Lire (R)

• Le gestionnaire consulte les profils en attente (code_statut = A)

Mettre à jour (U)

• Validation → A → N
• Attribution du rôle **ABO**
• Rattachement automatique au concours actif
• Blocage → statut B (aucun rôle)

Supprimer (D)

• Le gestionnaire peut supprimer définitivement un utilisateur
→ suppression de mcd_users, mcd_utilisateurs, et relations

Données & intégrité référentielle

Tables concernées :

• **mcd_users** : données de connexion (email, mot de passe, email_verified_at…)
• **mcd_utilisateurs** : données métier (nom, prénom, genre, code_statut)
• **mcd_roles** : rôles attribuables (VIS, ABO, GST…)
• **mcd_engager** : liaison entre utilisateur et concours

Règles à garantir :

• **Un user sans email vérifié n'a jamais de profil**
• **Le profil n'existe qu'après la vérification de l'email**
• mcd_utilisateurs.id = mcd_users.id (1-à-1 strict)
• Suppression en cascade entre user et profil
• Statuts :

- A → attente de validation
- N → normal (approuvé)
- B → bloqué
  • Rôle ABO attribué uniquement après validation gestionnaire
  • Email unique dans mcd_users
  • Aucune incohérence possible :
  → jamais de rôle sans profil
  → jamais de profil sans user

Tests à prévoir

A. Tests d'acceptation

1. **Inscription via email**
   - nouveaux comptes
   - renvoi email
   - redirections selon les cas
2. **Vérification email**
   - email_verified_at mis à jour
   - redirection vers /inscription
3. **Complétion du profil**
   - mot de passe temporaire remplacé

- insertion correcte dans mcd_utilisateurs
- statut A

4. **Modération gestionnaire**
    - A → N (rôle ABO + rattachement concours)
    - A → B (bloqué)
    - A → A (attente)
5. **Suppression définitive**
    - suppression user + profil
    - suppression automatique des engagements (cascade)
6. **Accès sécurisé**
    - pages gestion réservées à GST

Conclusion

La nouvelle version du système d'auto-inscription met l'accent sur :

• une véritable **sécurisation par email**,
• une séparation claire entre **compte technique** (mcd_users) et **profil métier** (mcd_utilisateurs),
• un contrôle stricte via un **statut** (A, N, B) et des **rôles** (dont ABO),
• un processus de modération fiable et transparent,
• une intégrité référentielle totale grâce aux règles de cohérence et de suppression en cascade.

Le système est robuste, maintenable et parfaitement adapté aux contraintes du Concours Robots.