



Maroc
youcode school
Académie Africaine de Formation Industrielle, Youcode, Safi

Soumis à

YouCode – École de Formation en Développement Web et Mobile

En réponse partielle aux exigences du projet fil rouge
de Développement Web dans le cadre du programme de formation

Présenté par

EDDERKAOUI OUSSAMA

FootConnect

Plateforme de gestion des matchs de football amateur

Année académique : 2024-2025

Encadré par : **OUCHAD HOUSSNI**

Contents

1	Contexte du Projet	2
2	Objectifs du Projet	2
3	Acteurs du Projet	2
4	Use Cases Détaillés	2
4.1	Administrateurs des Équipes	2
4.2	Arbitres	2
4.3	Administrateurs de la Fédération	3
4.4	Entraîneurs	3
5	Fonctionnalités Clés	3
6	Architecture et Technologies	4
7	Charte Graphique	4
8	Méthodologie	5
9	Planification du Projet	5
10	Livrables	6
11	Évaluations et Retours	6
12	Risques et Contraintes	6
13	Conclusion	6

1 Contexte du Projet

Le projet FootConnect est une plateforme web dédiée à la gestion des matchs de football amateur. L'objectif est de centraliser la gestion des matchs, des statistiques des joueurs, des blessures, et des sanctions, tout en améliorant la communication entre les différents acteurs du football (arbitres, administrateurs d'équipes, entraîneurs et fédération).

2 Objectifs du Projet

- **Centralisation des informations** : Une plateforme unique pour gérer les matchs, les joueurs, les sanctions et les blessures.
- **Suivi des performances** : Permettre aux entraîneurs et arbitres de suivre les statistiques des joueurs et des équipes.
- **Gestion des blessures et sanctions** : Automatiser la gestion des suspensions et sanctions.
- **Amélioration de la communication** : Notifications en temps réel pour la disponibilité des arbitres, gestion des blessures et informations partagées.

3 Acteurs du Projet

- **Administrateurs des équipes** : Gestion des joueurs et des matchs, suivi des blessures.
- **Arbitres** : Confirmer la disponibilité pour les matchs, saisir les événements et les blessures après chaque match.
- **Administrateurs de la fédération** : Valider les listes de joueurs, gérer les matchs et suivre les statistiques.
- **Entraîneurs** : Suivre les performances des joueurs, gérer les entraînements et sélectionner les joueurs pour les matchs.

4 Use Cases Détaillés

4.1 Administrateurs des Équipes

1. Gérer la Liste des Joueurs

- **Objectif** : Enregistrer et mettre à jour les joueurs de l'équipe.
- **Actions** : L'administrateur saisit les informations des joueurs (personnelles, rôle, état de santé, etc.) et soumet la liste à la fédération pour validation.

2. Valider la Liste des Joueurs

- **Objectif** : Soumettre la liste des joueurs à la fédération pour validation.
- **Actions** : L'administrateur soumet la liste, la fédération valide ou rejette la liste, et l'administrateur reçoit une notification.

3. Suivre les Blessures des Joueurs

- **Objectif** : Suivre les blessures et ajuster les matchs en conséquence.
- **Actions** : L'administrateur reçoit les informations sur les blessures après chaque match et met à jour les données.

4.2 Arbitres

1. Confirmer la Disponibilité pour un Match

- **Objectif** : Confirmer la disponibilité avant chaque match.
- **Actions** : L'arbitre reçoit une notification 7 jours avant, confirme sa disponibilité, et la fédération est informée.

2. Saisir les Sanctions et Événements du Match

- **Objectif** : Enregistrer les événements du match (buts, cartons, etc.).
- **Actions** : L'arbitre saisit les événements après le match et les soumet pour validation.

3. Saisir les Blessures des Joueurs

- **Objectif** : Enregistrer les blessures des joueurs après chaque match.
- **Actions** : L'arbitre saisit les blessures, y compris la gravité, et une notification est envoyée aux administrateurs des équipes concernées.

4.3 Administrateurs de la Fédération

1. Valider les Listes des Joueurs

- **Objectif** : Valider ou rejeter les listes des joueurs soumises par les administrateurs des équipes.
- **Actions** : La fédération vérifie et valide ou rejette chaque liste.

2. Suivi des Blessures et Sanctions

- **Objectif** : Suivre les sanctions et les blessures des joueurs.
- **Actions** : Accéder à un tableau de bord des blessures et sanctions pour un suivi global.

3. Gérer les Matches et Statistiques

- **Objectif** : Planifier et valider les matchs, générer des statistiques.
- **Actions** : Planifier les matchs, valider les résultats, et générer des rapports sur les performances.

4.4 Entraîneurs

1. Suivi des Performances des Joueurs

- **Objectif** : Suivre les performances des joueurs.
- **Actions** : Consulter les statistiques des joueurs et ajuster les choix tactiques.

2. Gérer les Entraînements et Sélection des Joueurs

- **Objectif** : Planifier les entraînements et sélectionner les joueurs pour les matchs.
- **Actions** : Planifier les entraînements et sélectionner les joueurs en fonction de leurs performances et état physique.

5 Fonctionnalités Clés

- **Gestion des utilisateurs** :
 - Création et gestion des rôles pour chaque acteur.
 - Gestion des accès et des permissions.
- **Planification des matchs** :
 - Interface calendrier pour organiser et valider les matchs.
 - Gestion des lieux et des horaires.
- **Saisie des événements du match** :
 - Les arbitres enregistrent les événements comme les buts et sanctions.
 - Interface intuitive pour la saisie.
- **Suivi des blessures** :
 - Les arbitres saisissent les blessures et des notifications sont envoyées aux administrateurs des équipes.
 - Gestion des informations sur la gravité et la durée de la blessure.

- **Statistiques des joueurs et équipes :**
 - Outils d'analyse pour suivre les performances.
 - Génération de rapports personnalisables.
- **Gestion des suspensions automatiques :**
 - Suspension automatique des joueurs ayant 4 cartons jaunes ou une carte rouge.
 - Notification aux administrateurs et entraîneurs.

6 Architecture et Technologies

- **Technologies Web :**
 - Frontend : HTML5, CSS3, JavaScript (Frameworks possibles : React, Angular, Vue.js). La charte graphique ci-dessous sera implémentée pour assurer une cohérence visuelle.
 - Backend : PHP (Laravel) avec Blade pour le templating.
 - Base de données : PostgreSQL.
- **API RESTful :** pour l'échange de données entre le frontend et le backend.
- **Hébergement :** Solution cloud (AWS, Google Cloud, Azure) ou serveur dédié, selon les besoins.
- **Sécurité :**
 - Authentification via JWT ou OAuth.
 - Cryptage des données sensibles.
 - Protection contre les attaques courantes (XSS, CSRF, injections SQL).

7 Charte Graphique

- **Palette de couleurs :**
 - Bleu Foncé (#1E3A8A) : Utilisé dans la barre latérale pour un contraste marqué et une lisibilité optimale.
 - Bleu Moyen (#2563EB) : Pour les boutons et les éléments interactifs afin de les rendre visibles et attractifs.
 - Gris Clair (#F3F4F6) : En arrière-plan pour créer une hiérarchie visuelle sans distraire l'utilisateur.
 - Gris Moyen (#D1D5DB) : Utilisé dans les cartes et les sections pour délimiter les zones de contenu.
 - Noir Charbon (#111827) : Pour le texte principal, garantissant une excellente lisibilité.
- **Typographies :**
 - Titre et en-têtes : Poppins, sans-serif – pour un aspect moderne et élégant.
 - Texte courant : Roboto, sans-serif – pour une lecture confortable et fluide.
- **Éléments visuels :**
 - **Boutons :**
 - Couleur : Bleu Moyen (#2563EB) avec du texte blanc pour le contraste.
 - Forme : Légèrement arrondie (border-radius: 8px) pour un aspect moderne.
 - Effet au survol : Ombre douce et légère augmentation de luminosité.
 - **Cartes d'événements de match :**
 - Fond gris clair pour différencier les sections.
 - Ombres subtiles pour un effet de profondeur.

- **Icônes :**
 - Utilisation d'icônes linéaires et modernes (Heroicons ou Font Awesome).
 - Couleur bleue pour garder une cohérence avec la palette.
- **Navigation latérale :**
 - Fond Bleu Foncé avec texte blanc pour une navigation claire et un bon contraste.
 - Indicateur de section active avec un surlignement plus clair.
- **Améliorations potentielles :**
 - Ajouter des animations douces pour les interactions (hover, clic).
 - Définir des espacements uniformes pour garder une structure aérée et cohérente.
 - Utiliser des images ou des illustrations légères pour renforcer l'aspect visuel sans surcharger l'interface.

8 Méthodologie

- **Méthodologie Agile :**
 - Utilisation de sprints de 2 à 4 semaines avec des révisions régulières.
 - Revue continue des besoins et des fonctionnalités avec les utilisateurs (entraîneurs, arbitres, administrateurs).
- **Développement itératif :**
 - Prototypes fonctionnels livrés à chaque fin de sprint pour recueillir des retours utilisateurs.
- **Tests utilisateurs :**
 - Tests effectués avec chaque groupe d'utilisateurs (entraîneurs, arbitres, administrateurs) tout au long du développement.

9 Planification du Projet

- **Phase 1 - Analyse et spécifications :** du 15/01/2025 au 31/01/2025
 - Collecte des besoins, définition des fonctionnalités et rédaction des spécifications techniques.
 - Définition détaillée de la charte graphique et réalisation de maquettes pour l'interface utilisateur (wireframes et mockups).
- **Phase 2 - Conception et prototypage :** du 01/02/2025 au 21/02/2025
 - Conception de l'architecture du système et de la base de données (PostgreSQL).
 - Réalisation de maquettes interactives de l'interface utilisateur.
- **Phase 3 - Développement et implémentation :** du 01/03/2025 au 15/04/2025
 - Développement des fonctionnalités principales : gestion des utilisateurs, gestion des équipes, planification des matchs.
 - Développement de l'API RESTful.
 - Utilisation de Blade pour la création des vues et gestion de la logique côté serveur avec Laravel. Application de la charte graphique.
- **Phase 4 - Tests et déploiement :** du 15/04/2025 au 21/04/2025
 - Tests unitaires, tests d'intégration et tests utilisateurs.
 - Déploiement sur un environnement de production.
- **Phase 5 - Finalisation et documentation :** du 21/04/2025 au 01/05/2025
 - Rédaction de la documentation utilisateur et technique.

10 Livrables

- **Application web fonctionnelle** avec toutes les fonctionnalités essentielles.
- **Documentation utilisateur** détaillée de la plateforme pour chaque acteur (entraîneurs, arbitres, administrateurs).
- **Documentation technique** détaillée de l'architecture, du code source et des technologies utilisées.
- **Rapport final de projet** : Présentation du processus de développement, des choix techniques, des difficultés rencontrées, et des améliorations possibles.

11 Évaluations et Retours

- **Tests utilisateurs** impliquant les acteurs du projet pour valider les fonctionnalités.
- **Démonstration finale** devant un panel d'enseignants et d'experts.
- **Questionnaire de satisfaction** pour recueillir les retours des utilisateurs afin d'améliorer la plateforme.

12 Risques et Contraintes

- **Risques techniques** :
 - **Difficulté d'intégration de certaines fonctionnalités** :
 - Solution : Planifier des phases de prototypage et de tests approfondis pour les fonctionnalités complexes.
 - Solution : Utiliser les fonctionnalités et les paquets de Laravel (ex: Eloquent pour l'accès à la base de données, etc) pour simplifier l'intégration.
 - **Gestion de grandes quantités de données** :
 - Solution : Optimiser la base de données (PostgreSQL) et les requêtes.
 - Solution : Utiliser des techniques de pagination et de mise en cache.
 - **Difficultés liées à Blade (templates)** :
 - Solution: Bien comprendre la syntaxe de Blade, et organiser les templates avec des layouts, sections et includes.
- **Risques de retard** :
 - **Ajout de fonctionnalités imprévues** :
 - Solution : Définir un scope clair et gérer les changements de périmètre avec une méthode agile.
 - Solution : Prioriser les fonctionnalités et les reporter si nécessaire.
 - **Problèmes techniques** :
 - Solution : Prévoir une marge de manœuvre dans le planning.
 - Solution : Mettre en place une gestion des incidents efficace.
- **Contraintes de temps** :
 - Respect des délais pour la présentation et la remise des livrables.
 - Solution : Planifier des jalons et des revues régulières.
 - Solution : Gérer les priorités et s'assurer de la disponibilité des ressources.
 - Solution : Utiliser des outils de gestion de projet (ex: Trello, Jira) pour suivre l'avancement.

13 Conclusion

Le projet FootConnect fournit une solution complète pour la gestion des matchs de football amateur. Il permet de suivre les performances des joueurs, de gérer les blessures et les sanctions, et de faciliter

la communication entre tous les acteurs du système (arbitres, entraîneurs, administrateurs d'équipes, et fédération). Ce projet répond à des besoins spécifiques et offre une interface intuitive, avec une esthétique soignée, pour améliorer l'organisation et l'efficacité des compétitions de football amateur.