**Cahier Des Charges**

**1. Description du projet**

L'application proposée a pour objectif de permettre aux utilisateurs de consulter les matchs amateurs à venir et de réserver des places.

Les responsables de stade auront la possibilité de publier des événements de matchs en renseignant les attributs nécessaires.

L'application permettra également aux administrateurs de gérer les utilisateurs, les stades, les catégories, les places, les réservations et les factures.

**2. Fonctionnalités par rôle**

1. **Fonctionnalités utilisateurs :**

* Consultation des matchs : Un utilisateur anonyme pourra consulter la liste des matchs à venir ainsi que les places disponibles.
* Inscription et authentification : Un utilisateur pourra créer un compte, s'authentifier, et se déconnecter.
* Réservation de places : Un utilisateur authentifié pourra réserver des places pour les matchs à venir, effectuer le règlement et recevoir une confirmation par email ou notification.

1. **Fonctionnalités responsables de stade :**

* Inscription et authentification : Un responsable de stade pourra s’inscrire après validation par un administrateur, s'authentifier et se déconnecter.
* Gestion des stades : Un responsable de stade pourra ajouter et gérer les stades dont il est responsable.
* Création d'événements (matchs) : Un responsable de stade pourra créer de nouveaux matchs en renseignant les informations suivantes :
  + Date et heure du match
  + Adresse du stade
  + Nom du stade
  + Nombre de places disponibles
  + Type de places disponibles
  + Tarifs pour chaque type de place

1. **Fonctionnalités administrateur :**

* Authentification et gestion des sessions : Un administrateur pourra s'authentifier, se déconnecter et gérer les sessions des utilisateurs et responsables de stade.
* Gestion complète de l’application : L'administrateur aura un contrôle total sur toutes les entités suivantes :
  + Utilisateurs
  + Stades
  + Catégories de places
  + Places
  + Réservations
  + Matches
  + Paniers
  + Factures

**3. Technologies utilisées**

* **Schéma de base de données :** UML / Schéma relationnel
* **Back-end API REST :**
  + Technologie : .NET Core
  + Base de données : SQL Server / MySQL
  + ORM : Entity Framework
  + Authentification : JSON Web Tokens (JWT) (optionnel)
  + Gestion des rôles : Identity
* **Stockage des factures** : SharePoint (factures au format PDF)
* **Application web (MVC) :** ASP.NET Core
* **Interface utilisateur** :
  + Bootstrap
  + Angular (optionnel)
* **Notifications et mails :** Power Automate
* **Suivi et analyse des logs :** Power BI
* **Gestion du code source :** GitHub
* **Intégration continue et déploiement :** Azure DevOps
* **Cloud :** Microsoft Azure