

# **Cahier des charges - Plateforme de gestion d'un tournoi de golf**

**Cours : Projet intégrateur**

**Programme : Programmation informatique**

**Session : Quatrième session**

**Professeur : Abdarahmane Ben mimoun**

---

**Nom de l'équipe : Équipe 6**

**Membres de l'équipe :**

- Chanez meziane
  - Rayane Tarki
  - hanane Khenteur
- 

**Thème du projet : \*\*Gestion d'un tournoi de golf\*\***

## **1. Présentation du projet**

### **1.1 Contexte**

Le collège La Cité organise chaque année un tournoi de golf destiné aux employés, aux retraités ainsi qu'aux partenaires commanditaires. Actuellement, plusieurs étapes de l'organisation (inscriptions, paiements, gestion des équipes et des commandites) sont réalisées de façon manuelle, ce qui entraîne des erreurs, une perte de temps et un manque de visibilité globale pour les organisateurs.

Dans un contexte académique de projet intégrateur, ce projet vise à concevoir une plateforme web complète permettant d'automatiser et de centraliser la gestion du tournoi de golf. Cette solution pourrait, à terme, être utilisée réellement par le collège pour simplifier l'organisation de l'événement.

### **1.2 Objectifs**

L'objectif principal est de développer une application web sécurisée permettant de gérer un tournoi de golf de manière efficace et professionnelle.

Objectifs spécifiques :

- Permettre aux participants de s'inscrire en ligne facilement
- Permettre le paiement en ligne sécurisé (ou simulé en contexte académique)

- Faciliter la création et la gestion des équipes de 4 joueurs
- Gérer les commandites et les partenaires commanditaires
- Offrir aux administrateurs un contrôle complet du tournoi

Critères de performance :

- Interface claire et facile à utiliser
- Fiabilité du système et absence d'erreurs critiques
- Sécurité des données personnelles
- Code structuré et maintenable

### **1.3 Périmètre du projet**

Inclus dans le projet :

- Développement d'une application web
- Gestion des utilisateurs et des rôles
- Gestion des inscriptions, paiements et équipes
- Gestion des commandites
- Base de données relationnelle
- Tests fonctionnels

Exclus du projet :

- Intégration avec de vrais systèmes bancaires
- Application mobile native
- Fonctionnalités avancées de statistiques ou d'analytique

## **2. Description fonctionnelle**

### **2.1 Besoins et exigences métiers**

Problématique : Les organisateurs du tournoi ont besoin d'un outil centralisé pour gérer efficacement les inscriptions, les paiements, les équipes et les commanditaires.

Utilisateurs cibles :

- Administrateurs (organisateurs du tournoi)
- Employés et retraités du collège
- Partenaires commanditaires

Scénarios d'utilisation :

- Un participant s'inscrit au tournoi, effectue le paiement et rejoint une équipe
- Un commanditaire choisit une commandite et inscrit des joueurs
- Un administrateur crée un tournoi, configure les paramètres et gère les équipes

#### **2.1.1 Règles d'affaires**

Les règles d'affaires définissent les contraintes et le fonctionnement du système afin d'assurer une gestion cohérente du tournoi.

### **Règles générales :**

- Un tournoi doit être créé par un administrateur avant toute inscription.
- Les inscriptions ne sont possibles que lorsque le tournoi est ouvert.
- Un utilisateur doit être authentifié pour accéder aux fonctionnalités protégées.

### **Règles liées aux équipes :**

- Une équipe de golf est composée de 4 joueurs maximum.
- Une équipe ne peut pas être rejointe si elle est complète.
- La création d'une équipe génère automatiquement un code secret unique.
- Un joueur ne peut appartenir qu'à une seule équipe.

### **Règles liées aux inscriptions :**

- Une inscription est valide uniquement après paiement (simulation).
- Les participants employés/retraités doivent s'inscrire individuellement.
- Les commanditaires peuvent inscrire plusieurs joueurs selon la commandite choisie.

### **Règles liées aux commandites :**

- Une commandite peut être monétaire ou inclure des places/équipes.
  - Le nombre de commandites par type est limité par le tournoi.
  - Les joueurs inscrits via une commandite doivent être associés au commanditaire.
- 2.2 Fonctionnalités principales
- Authentification sécurisée des utilisateurs
  - Gestion des rôles (administrateur, participant, commanditaire)
  - Création, modification et suppression d'un tournoi
  - Ouverture et fermeture des inscriptions
  - Crédit d'équipes avec code secret
  - Rejoindre une équipe existante
  - Gestion des commandites et des partenaires
  - Paiement en ligne (simulation)

## **2.3 Interface utilisateur**

L'interface sera intuitive, moderne et responsive.

Pages principales :

- Page d'accueil du tournoi
- Formulaire d'inscription participant
- Formulaire d'inscription commanditaire
- Tableau de bord administrateur

## 2.4 Conditions d'utilisation

- Accessible via navigateur web
- Compatible PC et mobile
- Utilisable dans un environnement académique et organisationnel

## 2.5 – User Stories

### **\*\* Administrateur**

1. **En tant qu'administrateur**, je veux créer un tournoi de golf afin de gérer les inscriptions et les équipes.
2. **En tant qu'administrateur**, je veux ouvrir et fermer les inscriptions afin de contrôler l'accès au tournoi.
3. **En tant qu'administrateur**, je veux gérer les équipes (ajouter, déplacer ou supprimer des joueurs) afin d'assurer des équipes complètes de 4 joueurs.
4. **En tant qu'administrateur**, je veux créer et configurer des types de commandites afin de gérer les partenaires du tournoi.
5. **En tant qu'administrateur**, je veux consulter les inscriptions afin d'avoir une vue globale du tournoi.

### **\*\* Employé / Retraité (Participant)**

1. **En tant que participant**, je veux m'inscrire au tournoi en ligne afin de réserver ma place.
2. **En tant que participant**, je veux payer les frais d'inscription afin de confirmer ma participation.
3. **En tant que participant**, je veux créer une équipe avec un code secret afin d'inviter d'autres joueurs.
4. **En tant que participant**, je veux rejoindre une équipe existante avec un code afin de jouer avec mes collègues.

### **\*\* Commanditaire**

1. **En tant que commanditaire**, je veux choisir un type de commandite afin de soutenir le tournoi.
2. **En tant que commanditaire**, je veux inscrire les joueurs inclus dans ma commandite afin de gérer ma participation.
3. **En tant que commanditaire**, je veux payer ma commandite afin de valider mon inscription.

## 2.6 Messages d'erreur, notifications et confirmations

Afin de guider l'utilisateur, le système affichera des messages clairs et explicites.

**Messages de confirmation :**

- Inscription réussie au tournoi.
- Paiement effectué avec succès.
- Équipe créée avec succès.
- Commandite enregistrée avec succès.

#### **Messages d'erreur :**

- Inscription impossible : tournoi complet.
- Inscription impossible : inscriptions fermées.
- Paiement échoué (simulation).
- Code d'équipe invalide ou équipe complète.
- Accès refusé : droits insuffisants.

#### **Notifications :**

- Confirmation par message à l'écran après chaque action importante.
- Messages visibles dans l'interface utilisateur (pas de courriel en contexte académique).

## **3. Description technique**

### **3.1 Technologies utilisées**

Les technologies ont été choisies pour leur fiabilité, leur compatibilité avec le programme d'études et leur facilité d'intégration.

Frontend :

- HTML, CSS, JavaScript
- Création d'interfaces web accessibles, responsives et compatibles avec tous les navigateurs.

Backend :

- ASP.NET
- Gestion de la logique métier, des utilisateurs, des rôles et de la sécurité.

Base de données :

- SQL Server

→ Stockage structuré des données (utilisateurs, tournois, équipes, commandites).

Gestion de version :

- Git / GitHub

→ Suivi des modifications du code et travail collaboratif.

Paiement :

- Simulation de paiement

→ Aucun lien avec un système bancaire réel (contexte académique).

### **3.2 Architecture du système**

Architecture client-serveur :

- Client web pour l'interface utilisateur
- Serveur applicatif pour la logique métier
- Base de données pour la persistance des données

### **3.3 Sécurité**

La sécurité est un aspect essentiel de la plateforme afin de protéger les données des utilisateurs.

Mesures de sécurité mises en place :

- Hashage des mots de passe avant stockage en base de données.
- Gestion sécurisée des sessions utilisateurs.
- Accès restreint aux fonctionnalités selon le rôle (administrateur, participant, commanditaire).
- Validation des données saisies dans les formulaires.
- Protection contre les accès non autorisés aux pages sensibles.
- Séparation des responsabilités entre les couches du système.

### **3.4 Performance et scalabilité**

Objectifs de performance :

- Temps de réponse inférieur à 2 secondes pour les actions principales.
- Support d'au moins 100 utilisateurs simultanés.
- Optimisation des requêtes SQL.

- Chargement progressif des données dans l'interface.

Le système est conçu pour être évolutif et supporter une augmentation future du nombre d'utilisateurs.

#### 4. Planification et livrables

##### 4.1 Phases du projet

- Semaine 1 : Analyse et rédaction du cahier des charges
- Semaine 2 : Conception (diagrammes, backlog, sprints)
- Semaine 3 à 6 : Développement et tests (Sprint de 4 semaines)
- Semaine 7 : Finalisation et présentation

##### 4.2 Backlog du produit (Product Backlog)

Le backlog du produit représente l'ensemble des fonctionnalités à développer pour la plateforme de gestion du tournoi de golf.

Ces fonctionnalités sont priorisées afin de guider le développement du projet selon une approche Agile.

| ID   | Description                                       | priorité      |
|------|---|---------------|
| US1  | Création de la base de données                    | <b>High</b>   |
| US2  | Authentification des utilisateurs                 | <b>High</b>   |
| US3  | Création et gestion d'un tournoi                  | <b>High</b>   |
| US4  | Ouverture et fermeture des inscriptions           | <b>High</b>   |
| US5  | Inscription des participants                      | <b>High</b>   |
| US6  | Paiement de l'inscription (simulation)            | <b>High</b>   |
| US7  | Création d'une équipe avec code secret            | <b>High</b>   |
| US8  | Rejoindre une équipe existante à l'aide d'un code | <b>Medium</b> |
| US9  | Gestion des équipes (administrateur)              | <b>Medium</b> |
| US10 | Configuration des types de commandites            | <b>Medium</b> |
| US11 | Inscription des commanditaires                    | <b>Medium</b> |
| US12 | Ajout des joueurs commanditaires                  | <b>Medium</b> |
| US13 | Tableau de bord administrateur                    | <b>Medium</b> |
| US14 | Sécurisation des accès par rôles                  | <b>High</b>   |
| US15 | Tests fonctionnels                                | <b>Medium</b> |
| US16 | Documentation du projet                           | <b>Low</b>    |

##### 4.3 Planification des sprints (Sprint Backlog)

Le projet est réalisé en plusieurs sprints afin d'avancer étape par étape et de développer les fonctionnalités progressivement.

---

## **Sprint 1 – Mise en place du projet (Semaines 1 à 4)**

Objectifs :

- Préparer l'environnement de travail
  - Créer la base de données
  - Mettre en place la connexion des utilisateurs
  - Créer le tournoi
  - Permettre l'inscription des participants
- 

## **Sprint 2 – Fonctionnalités principales (Semaines 5 à 8)**

Objectifs :

- Ajouter le paiement (simulation)
  - Créer et gérer les équipes
  - Rejoindre une équipe avec un code
  - Gérer les commandites
  - Incrire les joueurs des commanditaires
- 

## **Sprint 3 – Finalisation du projet (Semaines 9 à 12)**

Objectifs :

- Ajouter le tableau de bord administrateur
- Gérer les accès selon les rôles
- Tester les principales fonctionnalités
- Corriger les erreurs détectées

### **4.4 Livrables attendus**

- Cahier des charges validé
- Code source de l'application
- Base de données
- Documentation technique
- Présentation finale

### **5. Modalités de validation**

La validation du projet permettra de vérifier que la plateforme répond bien aux besoins définis dans le cahier des charges.

Plusieurs méthodes de validation seront utilisées tout au long du projet.

#### **Validation fonctionnelle :**

- Vérifier que chaque fonctionnalité principale fonctionne correctement (inscription, création d'équipe, paiement simulé, gestion des commandites).
- Tester les actions selon les différents rôles (administrateur, participant, commanditaire).

### **Tests de scénarios d'utilisation :**

- Un participant peut s'inscrire au tournoi et rejoindre une équipe.
- Un commanditaire peut choisir une commandite et inscrire ses joueurs.
- Un administrateur peut créer un tournoi et gérer les équipes.

### **Tests techniques :**

- Vérification de la connexion à la base de données.
- Vérification de la sécurité de base (connexion utilisateur, accès selon les rôles).

### **Validation finale :**

- Présentation du projet avec une démonstration des fonctionnalités principales.
- Validation du projet selon les exigences définies au départ.
- Correction des erreurs détectées avant la remise finale.

## **5.2 Stratégie de tests**

Une stratégie de tests est mise en place afin de garantir la qualité et la conformité de la plateforme par rapport au cahier des charges.

### **Types de tests utilisés :**

- Tests fonctionnels : vérification du bon fonctionnement de chaque fonctionnalité principale.
- Tests unitaires : validation des fonctions principales du système (authentification, inscription, gestion des équipes).
- Tests d'intégration : vérification des échanges entre l'interface utilisateur, le serveur et la base de données.

### **Outils utilisés :**

- Navigateur web (tests manuels).
- Visual Studio Code pour l'exécution et la correction du code.
- SQL Server pour la validation des données stockées.
- GitHub pour le suivi des corrections et des versions.

### **Critères de réussite :**

- Aucune erreur bloquante lors de l'utilisation.
- Fonctionnalités principales opérationnelles.
- Données enregistrées correctement en base de données.

- Accès sécurisé respectant les rôles utilisateurs.**6. Annexes**

### Ce qu'on fait dans le projet alors:

- Développer la plateforme web
- Gérer le code
- Gérer la base de données
- Tester et corriger

### Applications / outils utilisés

- **Visual Studio Code :**
  - Écriture du code (site web, logique, formulaires)
- **GitHub :**
  - Sauvegarde du code, travail en équipe
- **SQL (MySQL / SQL Server) :**
  - Stockage des données (utilisateurs, tournois, équipes)

- **Jira :**
  - Gestion des tâches

- **Scrum :**
  - Organisation du travail par sprints

- **Diagrammes :**

- Use Case → astah UML

- Classes → astah UML

- Séquence → astahUML

Tests :

- Fonctionnels → Navigateur web

- Base de données → SQL (MySQL / SQL Server)

- Corrections → VS Code

- **schéma finale**

- PLATEFORME WEB (Tournoi de golf)

- Administrateur (Gestion complète)

\*Gestion des utilisateurs

- Ajouter / supprimer administrateurs
- Consulter les inscriptions
- Gestion des tournois
- Créer / supprimer un tournoi
- Ouvrir / fermer les inscriptions
- Configurer le nombre de places et d'équipes

\* Gestion des équipes

- Créer des équipes (vides ou complètes)
  - Ajouter / déplacer / supprimer des joueurs
  - Vérifier les équipes complètes (4 joueurs)
- \* Gestion des commandites
- Créer les types de commandites
  - Définir les places / équipes incluses
  - Limiter le nombre de commandites par type
    - Employé / Retraité (Participant)

\*Inscription au tournoi

- Remplir informations personnelles
- Payer les frais d'inscription

\*Gestion d'équipe

- Créer une équipe (code secret)
- Rejoindre une équipe existante (code)

\*Participation

- \* Jouer dans une équipe de 4 joueurs maximum
- Commanditaire (Sponsor)

\*Choix de commandite

- Commandite monétaire
  - Commandite avec places
  - Commandite avec équipes
- \*Inscription et paiement
- Informations personnelles et entreprise
  - Paiement sécurisé
- \*Gestion des joueurs
- Ajouter les joueurs inclus
  - Fournir leurs informations

• **conclusion**

Ce cahier des charges présente une solution complète, réaliste et professionnelle pour la gestion d'un tournoi de golf. Le projet répond à un besoin réel du collège La Cité et démontre les compétences techniques et organisationnelles acquises dans le cadre du programme d'études.