[I. Liste des compétences du référentieL 3](#_Toc194519299)

[II. Résumé du projet 4](#_Toc194519300)

[III. Cahier des charges 5](#_Toc194519301)

[1) Description du cahier des charges du projet 6](#_Toc194519302)

[2) Les utilisateurs du projet (Membres, Professeurs, Admin) 6](#_Toc194519303)

[a. Membres (Utilisateurs inscrits) 6](#_Toc194519304)

[b. Professeurs 6](#_Toc194519305)

[c. Administrateurs 6](#_Toc194519306)

[3) Fonctionnalités principales du site (Accueil, Cours, Paiement, etc. 6](#_Toc194519307)

[a. Pour les visiteurs (non connectés) 6](#_Toc194519308)

[b. Pour les membres (connectés) 6](#_Toc194519309)

[c. Pour l’administrateur 7](#_Toc194519310)

[4) Exigences non fonctionnelles (Performance, Scalabilité, Sécurité) 7](#_Toc194519311)

[a. Performance 7](#_Toc194519312)

[b. Scalabilité 7](#_Toc194519313)

[c. Sécurité 7](#_Toc194519314)

[5) Architecture du système (Frontend, Backend, Bases de données) 7](#_Toc194519315)

[IV. Gestion de projet 7](#_Toc194519316)

[V. Spécifications techniques du projet 9](#_Toc194519317)

[1) Les contraintes techniques 9](#_Toc194519318)

[2) Les composants d’accès aux données (MySQL, MongoDB) 9](#_Toc194519319)

[3) Choix des technologies utilisées (Angular, Java, Spring Boot) 9](#_Toc194519320)

[4) Diagrammes et modélisation 9](#_Toc194519321)

[a. Diagramme de cas d'utilisation 9](#_Toc194519322)

[b. Diagramme de séquence 9](#_Toc194519323)

[c. Diagramme d'activité 9](#_Toc194519324)

[d. Diagramme de déploiement 9](#_Toc194519325)

[e. MCD (Modèle Conceptuel des Données) 9](#_Toc194519326)

[f. MLD (Modèle Logique des Données) 9](#_Toc194519327)

[g. MPD (Modèle Physique des Données) 9](#_Toc194519328)

[VI. . Réalisation du candidat 9](#_Toc194519329)

[1) Frontend (Angular) 9](#_Toc194519330)

[a. Structure du projet et navigation 9](#_Toc194519331)

[b. Implémentation des pages (Accueil, Cours, Paiement, etc.) 9](#_Toc194519332)

[c. Sécurisation de l’interface utilisateur (Authentification, Autorisations) 9](#_Toc194519333)

[2) Backend (Java, Spring Boot) 9](#_Toc194519334)

[a. Architecture du serveur et API REST 9](#_Toc194519335)

[b. Gestion des bases de données (MySQL et MongoDB) 9](#_Toc194519336)

[c. Sécurisation et gestion des utilisateurs (JWT, Authentification) 9](#_Toc194519337)

[3) Déploiement 9](#_Toc194519338)

[a. Configuration du serveur et choix de l’hébergement (Heroku, AWS, etc.) 9](#_Toc194519339)

[b. Processus de déploiement et mise en production 9](#_Toc194519340)

[4) Tests et validation 9](#_Toc194519341)

[a. Tests unitaires et fonctionnels (JUnit, Selenium) 9](#_Toc194519342)

[b. Tests de performance et de sécurité 9](#_Toc194519343)

[5) Outils de gestion de projet et versioning (Git, GitHub) 9](#_Toc194519344)

[VII. Analyse des performances et sécurité 10](#_Toc194519345)

[1) Analyse des performances du site (temps de réponse, optimisation) 10](#_Toc194519346)

[2) Sécurisation des données (cryptage, protections contre les attaques) 10](#_Toc194519347)

[VIII. Présentation du jeu d’essai 10](#_Toc194519348)

[1) Scénarios de tests réalisés 10](#_Toc194519349)

[2) Résultats des tests et validation 10](#_Toc194519350)

[IX. Veille technologique et recherche 10](#_Toc194519351)

[X. Conclusion 10](#_Toc194519352)

[1) Bilan du projet 10](#_Toc194519353)

[2) Perspectives d’évolution 10](#_Toc194519354)

[XI. Annexes 10](#_Toc194519355)

[1) Diagrammes UML détaillés 10](#_Toc194519356)

[2) Exemples de code source (extraits) 10](#_Toc194519357)

# Liste des compétences du référentieL

# Résumé du projet

Le projet **"Plateforme de gestion pour un club de Taekwondo"** vise à développer un site web complet permettant à un club de Taekwondo de gérer ses cours, ses adhésions, ses paiements, et ses actualités de manière moderne et intuitive. Cette plateforme sera utilisée par plusieurs types d’utilisateurs : les membres, les professeurs, et les administrateurs du site.

Le site web se compose de plusieurs pages principales, dont :

* **Page d'accueil** : qui présente les horaires des cours, les informations sur les professeurs, les avis des membres, ainsi que les actualités du club.
* **Page des tarifs** : permettant aux membres de consulter les tarifs et de procéder au *paiement des licences et des cours en ligne.*
* **Page d'inscription** : où les nouveaux membres peuvent s'inscrire en déposant leurs documents nécessaires (certificat médical, photo, etc.).
* **Page galerie** : pour afficher des photos et vidéos des événements et entraînements du club.
* **Page de contact** : offrant un moyen de communication entre le club et ses membres.
* **Page de connexion** : avec un accès distinct pour les membres et un tableau de bord administrateur.

L'administrateur a accès à un tableau de bord complet pour gérer les informations du site, y compris la gestion des horaires, des cours, des inscriptions, des tarifs, et des galeries. Ce dernier pourra aussi suivre les paiements et gérer les membres. De son côté, l'utilisateur peut suivre ses paiements, consulter ses documents déposés, et visualiser ses inscriptions via son propre tableau de bord.

Le développement du site repose sur des technologies modernes, avec un frontend développé avec Angular et un backend en Java avec Spring Boot. Les données sont stockées à la fois dans une base de données relationnelle (MySQL) pour la gestion des informations structurées, et une base de données NoSQL (MongoDB) pour la gestion des documents et des fichiers multimédias.

L'objectif de ce projet est de fournir une solution de gestion complète et centralisée pour le club de Taekwondo, tout en assurant une sécurisation des données (membres, paiements, documents) et une expérience utilisateur fluide.

# Cahier des charges

## Description du cahier des charges du projet

Le projet consiste à développer une plateforme web pour un club de **Taekwondo**, permettant la gestion des inscriptions, des paiements, des cours et des documents administratifs.

L’objectif est d’offrir une solution numérique **ergonomique, sécurisée et performante** qui simplifie la gestion du club pour les administrateurs et facilite l’accès aux informations pour les membres.

Le projet repose sur une **architecture moderne**, combinant **une base de données relationnelle (MySQL)** et **une base NoSQL (MongoDB)**, avec une interface développée en **Angular** et un backend en **Java Spring Boot**.

## Les utilisateurs du projet (Membres, Professeurs, Admin)

#### Membres **(**Utilisateurs inscrits)

* **Inscription en ligne** et dépôt des documents obligatoires (certificat médical, photo, etc.).
* **Consultation des horaires** et des informations du club.
* **Paiement en ligne** des licences et des cours.
* **Accès à un tableau de bord** personnel pour suivre ses paiements et documents.

#### Professeurs

* Consultation des plannings des cours.
* Ajout de contenus pédagogiques (documents, vidéos, etc.).

#### Administrateurs

* **Gestion des membres** (validation des inscriptions et documents).
* **Suivi des paiements** et validation des transactions.
* **Gestion des cours et horaires**.
* **Mise à jour du contenu du site** (galerie, actualités, avis).

## Fonctionnalités principales du site (Accueil, Cours, Paiement, etc.

#### Pour les visiteurs (non connectés)

* **Page d’accueil** : Présentation du club, des professeurs, des horaires et des actualités.
* **Page des tarifs** : Consultation des offres et possibilité de paiement en ligne.
* **Page d’inscription** : Formulaire pour déposer les documents requis.
* **Page galerie** : Photos et vidéos des événements du club.
* **Page contact** : Formulaire pour contacter le club.

#### Pour les membres (connectés)

* **Tableau de bord personnel** pour suivre les paiements et documents déposés.
* **Historique des inscriptions** et paiements.
* **Accès aux horaires et informations des cours.**

#### Pour l’administrateur

* **Gestion des inscriptions et documents** (validation, suppression).
* **Gestion des paiements et suivi des transactions**.
* **Ajout, modification et suppression des cours et horaires**.
* **Gestion des avis, actualités et contenus multimédias du site**.

## Exigences non fonctionnelles (Performance, Scalabilité, Sécurité)

#### Performance

* Temps de réponse inférieur à **3 secondes** pour l’affichage des pages principales.
* Optimisation des requêtes SQL et NoSQL pour minimiser la charge serveur.

#### Scalabilité

* Possibilité d’ajouter de nouvelles fonctionnalités sans modifier l’architecture principale.
* Gestion efficace du nombre croissant d’utilisateurs et de documents.

#### Sécurité

* **Authentification sécurisée** avec JWT.
* **Chiffrement des données sensibles** (mots de passe, paiements, documents).
* **Protection contre les attaques courantes** (SQL Injection, XSS, CSRF).
* **Gestion des rôles et permissions** (utilisateurs, professeurs, administrateurs).

## Architecture du système (Frontend, Backend, Bases de données)

* **Frontend : Développé avec Angular pour une interface utilisateur interactive et moderne.**
* **Backend : Gestion des requêtes et de la logique métier via Java et Spring Boot.**
* **Bases de données : Stockage des informations utilisateurs et contenus pédagogiques via MySQL et MongoDB.**

# Gestion de projet

# Spécifications techniques du projet

## Les contraintes techniques

## Les composants d’accès aux données (MySQL, MongoDB)

## Choix des technologies utilisées (Angular, Java, Spring Boot)

## Diagrammes et modélisation

## Diagramme de cas d'utilisation

## Diagramme de séquence

## Diagramme d'activité

## Diagramme de déploiement

## MCD (Modèle Conceptuel des Données)

## MLD (Modèle Logique des Données)

## MPD (Modèle Physique des Données)

# . Réalisation du candidat

## Frontend (Angular)

### Structure du projet et navigation

### Implémentation des pages (Accueil, Cours, Paiement, etc.)

### Sécurisation de l’interface utilisateur (Authentification, Autorisations)

## Backend (Java, Spring Boot)

### Architecture du serveur et API REST

### Gestion des bases de données (MySQL et MongoDB)

### Sécurisation et gestion des utilisateurs (JWT, Authentification)

## Déploiement

### Configuration du serveur et choix de l’hébergement (Heroku, AWS, etc.)

### Processus de déploiement et mise en production

## Tests et validation

### Tests unitaires et fonctionnels (JUnit, Selenium)

### Tests de performance et de sécurité

## Outils de gestion de projet et versioning (Git, GitHub)

# VII. Analyse des performances et sécurité

## 1) Analyse des performances du site (temps de réponse, optimisation)

## 2) Sécurisation des données (cryptage, protections contre les attaques)

# VIII. Présentation du jeu d’essai

## 1) Scénarios de tests réalisés

## 2) Résultats des tests et validation

# IX. Veille technologique et recherche

# X. Conclusion

## 1) Bilan du projet

## 2) Perspectives d’évolution

# XI. Annexes

## 1) Diagrammes UML détaillés

## 2) Exemples de code source (extraits)