

Table des matières

Introduction Générale	1
1 Cadre du projet	3
1.1 Présentation de l'organisme d'accueil	4
1.2 Présentation de l'établissement	4
1.3 Cadre du projet	4
1.4 Objectifs du projet	5
1.5 Étude et Critique du Système Existant	6
1.6 Problématique	6
1.7 Solution Proposée	6
1.8 Méthodologie Scrum	7
1.8.1 Définition de Scrum	7
1.8.2 Utilisation de Scrum	7
1.8.3 Mise en œuvre	7
2 Analyse des besoins	9
2.1 Spécifications des besoins	10
2.1.1 Besoins fonctionnels	10
2.1.2 Besoins non fonctionnels	11
2.2 Gantt Chart	11
2.3 Diagramme de cas d'utilisation	11
3 Conception	15
3.1 Choix technologiques	16

3.1.1	IntelliJ IDEA	16
3.1.2	Visual Paradigm	16
3.1.3	Wampp	17
3.1.4	GitHub	17
3.1.5	LaTeX	18
3.2	Diagrammes de conception	18
4	Implémentation	21
4.1	Architecture Client-Serveur	22
4.1.1	Flux de fonctionnement	23
4.2	Modèle Spring MVC	24
4.2.1	Composants du modèle	24
4.2.2	Étapes du traitement	25
4.2.3	Avantages de Spring MVC	26
4.3	Implémentation du site web	26
4.4	Déploiement sur Github	28

Table des figures

1.1	Logo School	4
2.1	Logo Gantt	11
2.2	Diagramme de cas d'utilisation de l'administrateur	12
2.3	Diagramme de cas d'utilisation de membre	13
2.4	Diagramme de cas d'utilisation de l'utilisateur	14
3.1	Logo Visual Paradigm	17
3.2	Logo Wamp	17
3.3	Logo de GitHub	18
3.4	Logo Latex	18
3.5	diagramme de classe	20
4.1	Architecture client-serveur	23
4.2	Modèle spring MVC	24
4.3	Sign in	27
4.4	Déploiement sur GitHub	28

Abbreviations list

— **Polytech**
Intl = **Polytechnique Internationale**

— **UML** = **Unified Modeling Language**

Introduction Générale

Ce projet vise à développer une application web dynamique dédiée à la gestion des clubs sportifs, dans le but de simplifier les processus administratifs et d'améliorer l'expérience utilisateur. Le secteur sportif fait face à de nombreux défis, notamment la planification des activités, la gestion des membres et le suivi des paiements. Dans ce contexte, l'adoption d'une solution numérique intuitive et centralisée s'impose comme une réponse essentielle aux besoins de modernisation et d'optimisation des clubs.

L'objectif principal de ce projet est de numériser et centraliser les fonctions essentielles des clubs sportifs, telles que l'inscription des membres, la gestion des plannings, des équipements et des paiements en ligne. La solution proposée offre une plateforme sécurisée et évolutive, intégrant des fonctionnalités avancées telles que l'affichage des statistiques en temps réel à travers des tableaux de bord interactifs, et une gestion simplifiée des adhérents et des activités. Elle permet ainsi de surmonter les limites des systèmes existants, souvent fragmentés, qui nuisent à l'efficacité et à la qualité du service.

Ce projet est réalisé en lien avec une démarche académique, il repose sur l'utilisation de technologies modernes telles que Java EE, Spring MVC, et suit la méthodologie agile Scrum, offrant une organisation flexible du travail en itérations courtes.

Le rapport est structuré en quatre chapitres principaux :

- Le premier chapitre présente le cadre général du projet, incluant la présentation de l'organisme d'accueil, les objectifs, la problématique et la solution proposée ainsi que la méthodologie.
- Le deuxième chapitre est consacré à l'analyse des besoins, avec la définition des spécifications fonctionnelles et non fonctionnelles, la planification à l'aide d'un diagramme de Gantt ainsi que le diagramme de cas d'utilisation.
- Le troisième chapitre porte sur la conception en détaillant les choix technologiques et les aspects techniques de l'implémentation.
- Le quatrième chapitre présente la phase de développement et de déploiement, en mettant en

lumière les résultats obtenus et les pistes d'amélioration.

Enfin, le rapport se conclut par une conclusion générale mettant en avant l'utilité du projet, tant sur le plan professionnel que personnel.

CADRE DU PROJET

Introduction

Dans ce chapitre, nous présentons d'abord l'organisme d'accueil dans lequel nous allons développer notre projet. Ensuite, nous présentons le sujet de notre projet et la méthodologie adoptée pour son développement.

1.1 Présentation de l'organisme d'accueil

Polytech Intl est une école d'ingénieurs privée en Tunisie qui vise l'excellence à travers des formations de qualité dédiées aux étudiants tunisiens et internationaux.



FIGURE 1.1 : Logo School

1.2 Présentation de l'établissement

Polytechnique Internationale est un établissement d'enseignement supérieur privé créé en 2013 et accrédité par le Ministère de l'Enseignement Supérieur et de la Recherche Scientifique Tunisien sous l'accréditation 04/2013, visant principalement à former de futurs cadres techniques.

— La stratégie de Polytech Intl :

Confier aux étudiants la tâche de créer un site Web dynamique de gestion de club sportif représente une occasion unique pour eux d'appliquer les compétences pratiques et théoriques acquises dans le domaine de l'ingénierie. En concevant et en développant un tel projet, les étudiants peuvent mettre en pratique leurs connaissances en programmation, en conception et en gestion de projet.

1.3 Cadre du projet

Dans le cadre de ce projet, l'objectif principal est de concevoir une application web pour la gestion des clubs sportifs en ligne, visant à moderniser et à optimiser les processus opérationnels

des clubs. L'application s'inscrit dans une démarche de transformation numérique, répondant aux besoins spécifiques du secteur sportif tout en simplifiant la gestion administrative et en améliorant l'expérience utilisateur.

Le secteur du sport est confronté à des défis multiples, tels que la gestion complexe des plannings d'activités, le suivi des paiements, etc. Ces contraintes, souvent amplifiées par l'absence de solutions technologiques adaptées, impactent la qualité du service offert aux membres et la productivité des administrateurs. Pour relever ces défis, il est indispensable d'adopter une solution numérique complète, intuitive et accessible.

Le projet décrit ici vise à développer une application robuste et flexible, qui permet de gérer efficacement les activités du club, les membres et les paiements. L'application intègre des fonctionnalités avancées, telles que la gestion des plannings (création, modification, suppression), un système d'inscription sécurisé, et une gestion des paiements en ligne.

Ainsi, cette application web contribue à la transformation numérique des clubs sportifs en répondant aux exigences du secteur.

1.4 Objectifs du projet

Le projet proposé vise à répondre à plusieurs problématiques rencontrées actuellement dans la gestion des clubs sportifs. Ces problématiques concernent la gestion des adhérents, des activités, des équipements et des paiements.

Ainsi, le développement de cette application web représente une opportunité de centraliser les fonctionnalités essentielles pour moderniser et simplifier la gestion des clubs sportifs.

Cette initiative permettra également de :

- Simplifier la gestion administrative des clubs sportifs grâce à une plateforme intuitive et conviviale.
- Renforcer la satisfaction des adhérents en automatisant les processus de paiement et de consultation des plannings.
- Améliorer la prise de décision grâce à des tableaux de bord interactifs offrant des statistiques en temps réel.
- Assurer une gestion centralisée des équipements et des activités pour optimiser l'utilisation des

ressources disponibles.

En somme, cette application vise à transformer la manière dont les clubs sportifs fonctionnent en apportant des outils modernes, accessibles et efficaces à leurs administrateurs et membres.

1.5 Étude et Critique du Système Existant

Dans cette section, nous analysons l'état actuel des solutions disponibles pour la gestion des clubs sportifs. Bien que plusieurs outils et logiciels existent, peu d'entre eux répondent entièrement aux besoins spécifiques des clubs sportifs en termes de gestion centralisée et numérique. Les systèmes existants, souvent hétérogènes ou limités à une seule fonctionnalité, ne permettent pas une intégration fluide entre les différents aspects de la gestion du club. Par exemple, les outils de gestion des membres ne sont pas connectés à ceux de planification des activités ou de gestion des paiements. Cette fragmentation rend le processus inefficace et difficile à suivre.

1.6 Problématique

Les clubs sportifs rencontrent des défis majeurs liés à une gestion manuelle et fragmentée, entraînant un manque d'efficacité, une coordination difficile, et une faible satisfaction des membres. L'absence de centralisation des données complique le suivi des adhérents, des activités, des paiements et des ressources. De plus, les outils limités réduisent l'expérience utilisateur et nuisent à la rentabilité des clubs, les rendant moins compétitifs face à des structures plus modernisées.

1.7 Solution Proposée

La solution proposée consiste en une application web JEE centralisée pour la gestion des clubs sportifs, intégrant les technologies Spring MVC et un design intuitif. Cette application permet de :

- Automatiser la gestion des adhérents, coachs, activités, équipements, et paiements.
- Centraliser toutes les données et opérations, facilitant la traçabilité et le suivi.
- Offrir des tableaux de bord interactifs pour une prise de décision éclairée.
- Améliorer l'expérience utilisateur grâce à une interface conviviale, accessible en ligne.

En modernisant la gestion des clubs sportifs, cette application vise à optimiser l'efficacité opérationnelle

et à améliorer la satisfaction des membres et du personnel.

1.8 Méthodologie Scrum

1.8.1 Définition de Scrum

La méthodologie Scrum est un cadre agile utilisé pour la gestion de projets, particulièrement adapté au développement de logiciels. Bien que ce projet soit réalisé dans un cadre académique, la méthodologie Scrum a été adoptée comme cadre de travail pour structurer et organiser les différentes phases de développement. Scrum, largement utilisée dans les environnements professionnels, a été simulée pour mieux comprendre son application dans un contexte de projet. Dans le contexte de ce projet de gestion de club sportif, Scrum a été adopté pour sa flexibilité, sa capacité à répondre aux changements rapides et son approche itérative, qui permet de livrer des fonctionnalités exploitables à chaque cycle.

1.8.2 Utilisation de Scrum

- Organiser le développement en plusieurs itérations courtes (sprints).
- Découper les fonctionnalités en tâches plus petites et gérables.
- Tester et valider régulièrement les parties développées pour garantir leur bon fonctionnement

1.8.3 Mise en œuvre

- **Sprint Planning** : Les fonctionnalités principales (gestion des adhérents, activités, paiements) ont été priorisées et planifiées en sprints.
- **Daily Meetings** : Bien que de manière informelle, des réunions rapides ont été organisées entre les membres pour discuter des progrès et résoudre les obstacles techniques.
- **Sprint Review et Retrospective** : À la fin de chaque sprint, les modules développés ont été revus et améliorés sur la base des retours internes et des besoins du projet.

Conclusion

Dans ce chapitre introductif, nous avons présenté le contexte de notre projet, l'organisme d'accueil, nous avons aussi présenté une analyse du système existant dans le domaine du sport et a exploré l'utilisation de la méthodologie Scrum pour structurer le développement du projet.

CHAPITRE 2

ANALYSE DES BESOINS

Introduction

Ce deuxième chapitre a pour objectif de présenter l'analyse et les spécifications des besoins de ce projet, ainsi que le diagramme approprié.

2.1 Spécifications des besoins

Le système à développer doit permettre une gestion complète des membres, des activités et des opérations associées. Les besoins fonctionnels principaux sont les suivants :

2.1.1 Besoins fonctionnels

- **Gestion des membres :**

Création, modification et suppression des membres (adhérents et coachs).

Consultation des profils et historique des inscriptions.

- **Gestion des activités :**

Ajout, modification, suppression des activités.

- **Gestion des plannings :**

Affichage des horaires des activités pour les membres.

- **Paielements :**

Génération de factures et intégration de paiement en ligne

- **Tableaux de bord :**

Statistiques sur les activités, membres et paiements.

2.1.2 Besoins non fonctionnels

Voici les exigences qui caractérisent le système. Notre système doit répondre aux critères suivants :

- **Sécurité** : Authentification et autorisation pour les membres (Spring Security).
Chiffrement des données sensibles (ex. : mots de passe).
- **Scalabilité** : Capacité à gérer un nombre croissant de membres, activités et paiements sans dégradation des performances.
- **Maintenance** : Code clair et bien documenté pour faciliter les mises à jour futures.

2.2 Gantt Chart

Nous estimons le temps de développement de chaque tâche à réaliser tout au long de notre stage. Pour ce faire, nous utiliserons un outil de planification de projet, le diagramme de Gantt. Ce dernier est illustré dans la figure suivante :

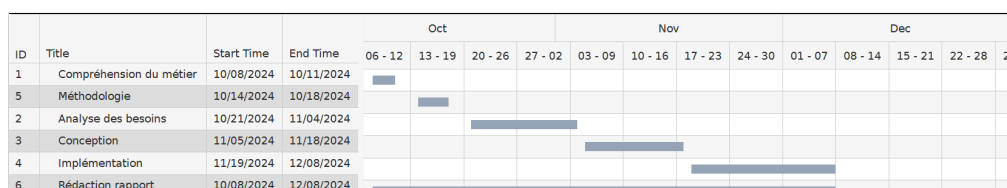


FIGURE 2.1 : Logo Gantt

2.3 Diagramme de cas d'utilisation

Dans cette section, nous présenterons le diagramme côté analyse.

— Diagramme de cas d'utilisation

Ci-dessous le diagramme de cas d'utilisation :

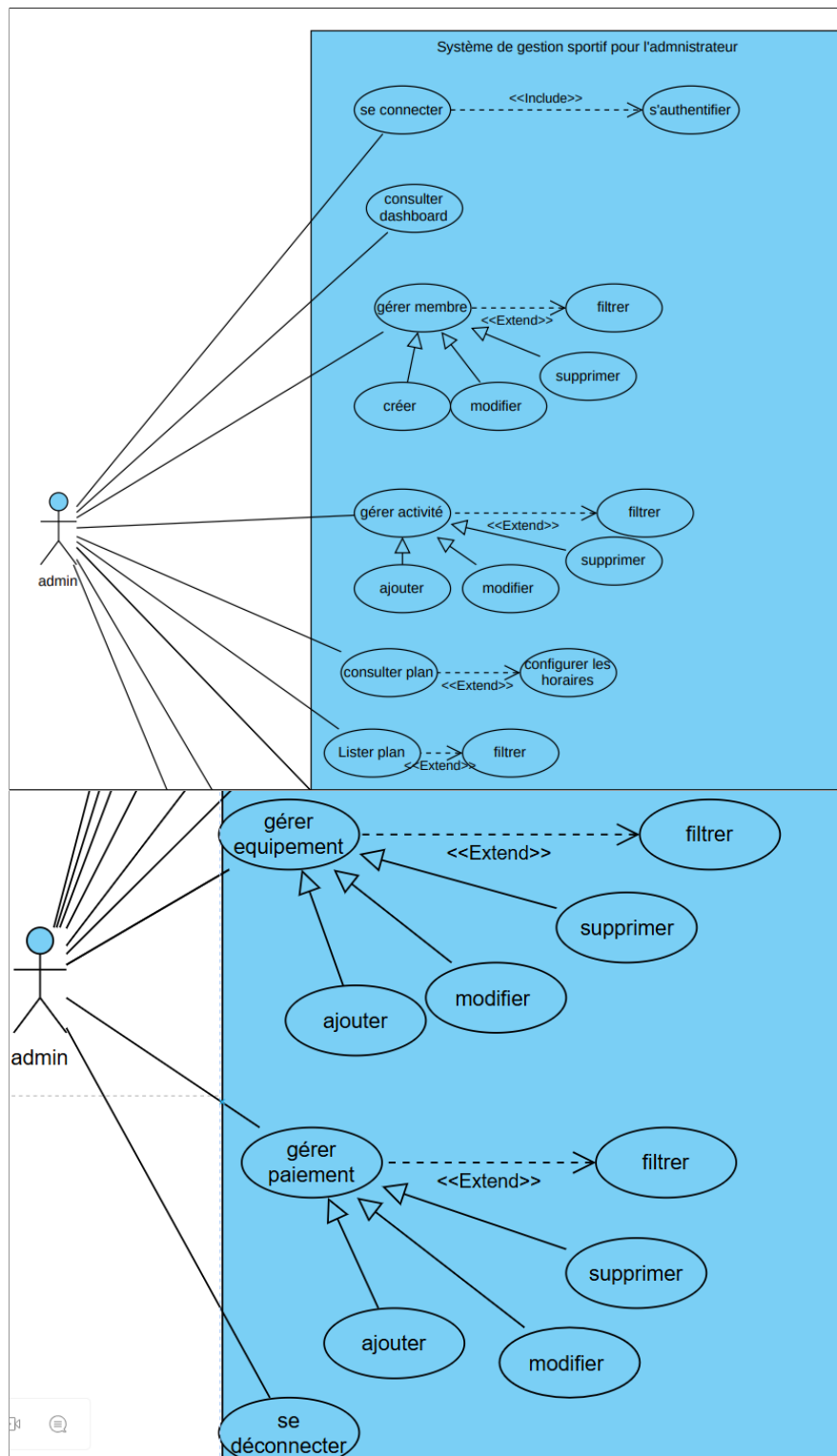


FIGURE 2.2 : Diagramme de cas d'utilisation de l'administrateur

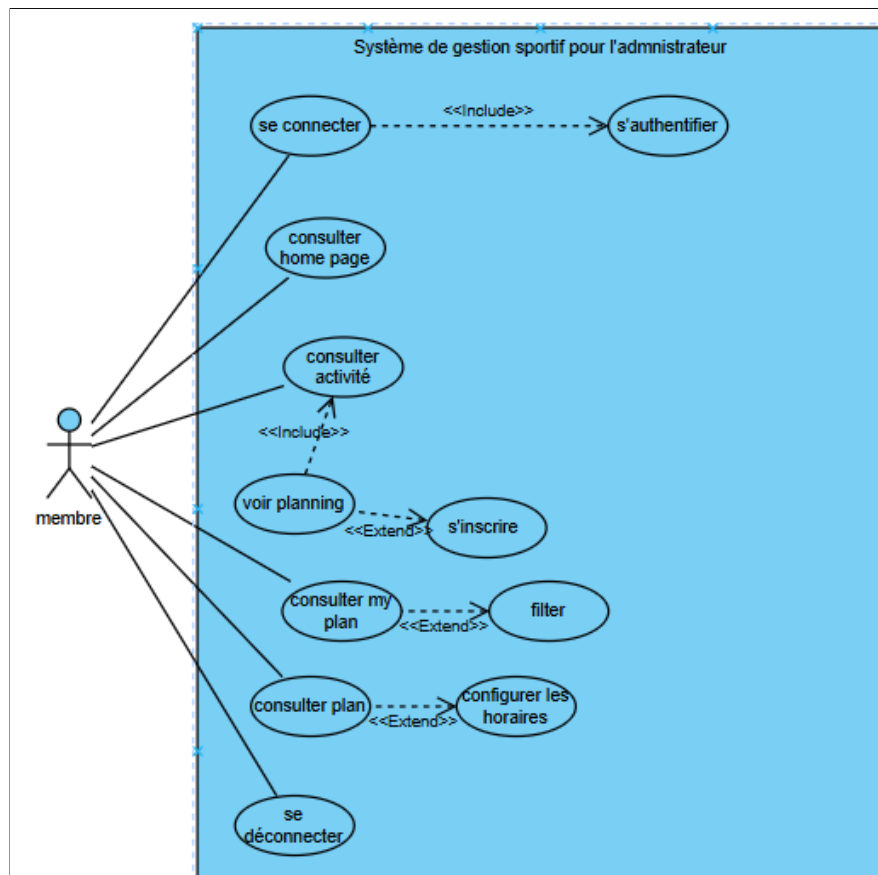


FIGURE 2.3 : Diagramme de cas d'utilisation de membre

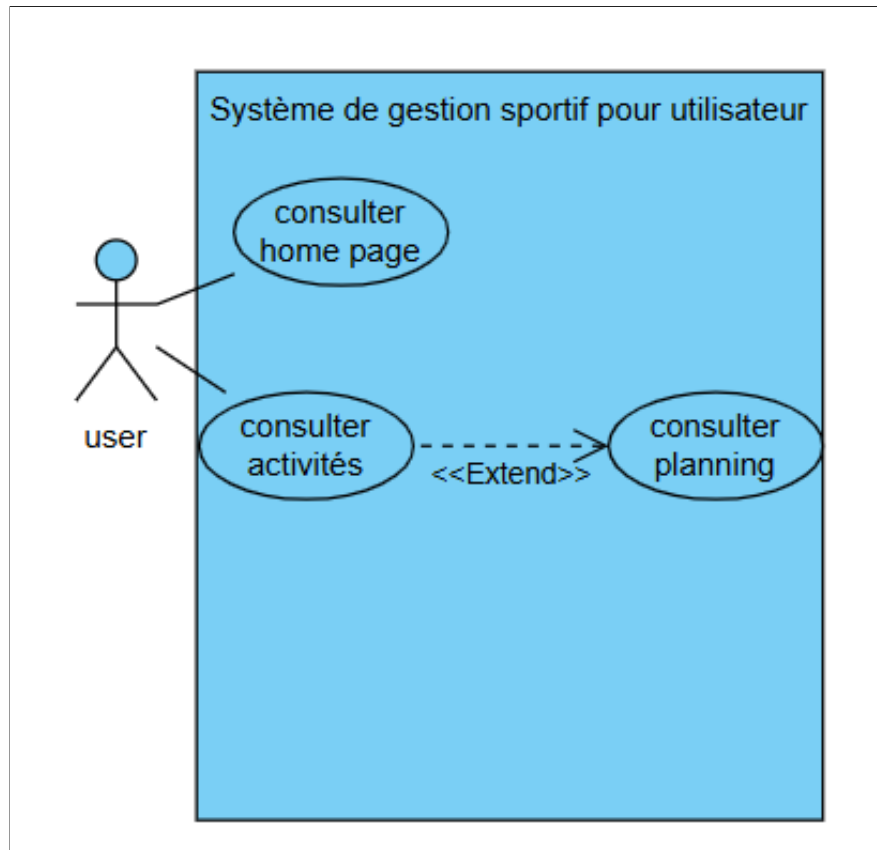


FIGURE 2.4 : Diagramme de cas d'utilisation de l'utilisateur

Conclusion

Cette conclusion souligne les avancées réalisées dans le deuxième chapitre, principalement centrées sur la définition des exigences fonctionnelles et non fonctionnelles, ainsi que sur l'analyse du diagramme côté analyse. Le chapitre suivant se penchera ensuite sur la phase de conception, offrant une exploration approfondie de notre approche et de nos stratégies.

CHAPITRE 3

CONCEPTION

Introduction

Ce troisième chapitre est consacré à la conception technique du système ainsi qu'au choix des technologies utilisées pour le développement de l'application web de gestion de clubs sportifs.

3.1 Choix technologiques

Dans cette section, nous présenterons les descriptions des outils logiciels et des technologies à mettre en œuvre.

3.1.1 IntelliJ IDEA

IntelliJ IDEA est un environnement de développement intégré (IDE) compatible avec de nombreux langages (Java, Kotlin, Groovy, Scala, JavaScript, etc.). Développé par JetBrains, il est reconnu comme l'un des IDE les plus performants et productifs pour les développeurs.

- Il est compatible avec les frameworks comme Spring Boot, Hibernate, JPA, Thymeleaf et bien d'autres.

3.1.2 Visual Paradigm

Visual Paradigm est un logiciel de modélisation et de conception de systèmes informatiques, qui permet de gérer et de suivre les exigences des projets.

il permet aux développeurs de :

- Créez des diagrammes UML tels que des diagrammes de cas d'utilisation, diagramme de classe, etc.

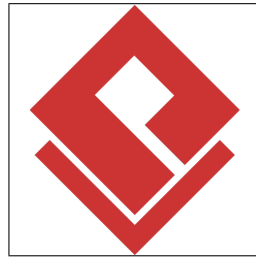


FIGURE 3.1 : Logo Visual Paradigm

3.1.3 Wampp

WAMP est un acronyme qui désigne un environnement de développement web combinant plusieurs technologies : Windows (OS), Apache (serveur web), MySQL (système de gestion de bases de données) et PHP (langage de programmation web). WAMP est une solution idéale pour les développeurs souhaitant configurer un serveur local pour créer, tester et déployer des applications web sur une machine fonctionnant sous Windows.



FIGURE 3.2 : Logo Wamp

- WAMP simplifie la configuration manuelle d'Apache, MySQL et PHP.
- Accessible gratuitement, ce qui en fait une solution populaire pour les développeurs indépendants et les étudiants.

3.1.4 GitHub

GitHub est une plateforme collaborative de développement logiciel basée sur le système de gestion de versions Git.

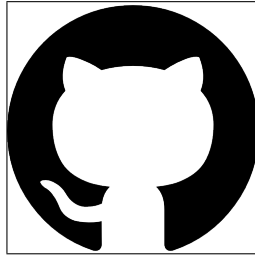


FIGURE 3.3 : Logo de GitHub

Elle offre aux développeurs un ensemble de fonctionnalités permettant de :

- Collaborer efficacement sur des projets en équipe
- Partager et héberger du code source
- Suivre et gérer les problèmes (issues)
- Gérer les différentes versions du code grâce au système de branches
- Bénéficier d'outils pour la gestion de projet, l'intégration continue et le déploiement automatisé

3.1.5 LaTeX

LaTeX est un système de composition de documents largement utilisé pour créer des documents techniques et scientifiques de haute qualité.



FIGURE 3.4 : Logo Latex

3.2 Diagrammes de conception

Dans cette section, nous présentons le diagramme de conception réalisés dans le cadre du projet.

— Diagramme de classe

Ci-dessous, le diagramme de classe représentant la structure générale des entités principales du système :

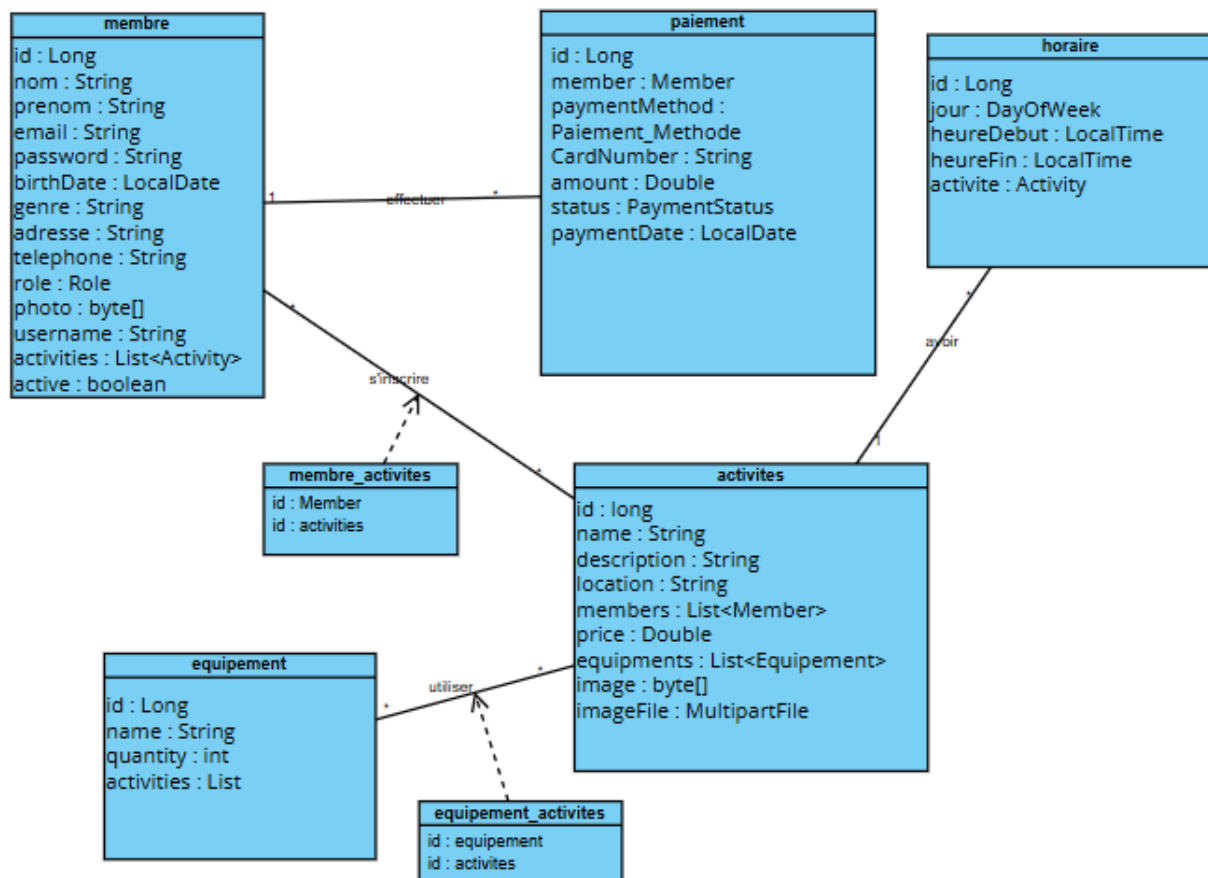


FIGURE 3.5 : diagramme de classe

Conclusion

Ce troisième chapitre met en évidence l'avancement du projet, en se concentrant principalement sur la conception technique à travers la modélisation du diagramme de classe. Cette représentation permet de mieux visualiser l'architecture du système et les relations entre ses composants.

IMPLÉMENTATION

Introduction

Ce quatrième chapitre est consacré à la présentation de la mise en œuvre pratique du projet ainsi que l'architecture Client-Serveur.

4.1 Architecture Client-Serveur

Dans cette section, nous présentons l'architecture Client-Serveur. C'est une architecture où plusieurs clients (utilisateurs) interagissent avec un serveur centralisé, qui traite les requêtes, exécute la logique métier, accède aux données, et renvoie les résultats aux clients. L'image ci-dessous illustre l'architecture Client-Serveur :

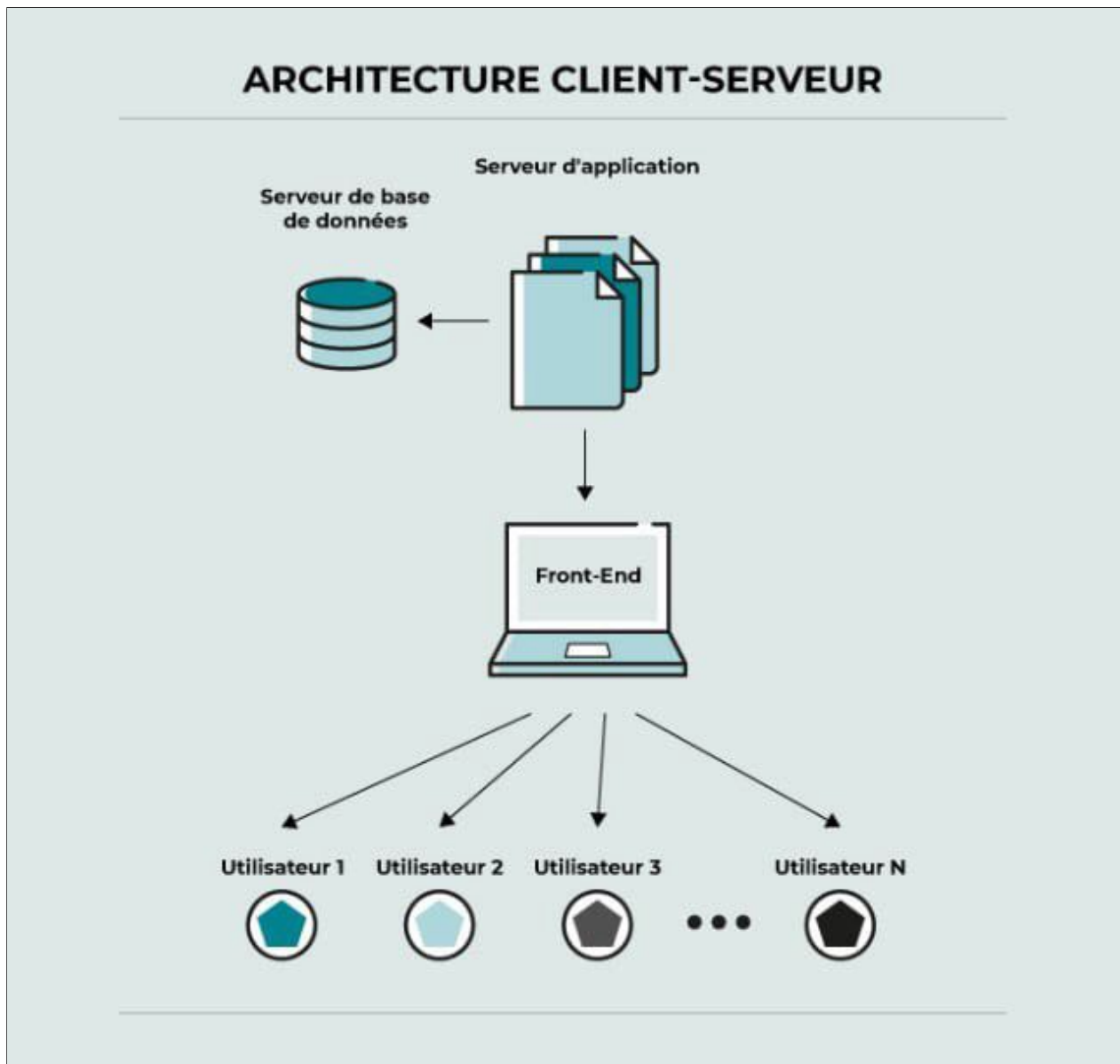


FIGURE 4.1 : Architecture client-serveur

4.1.1 Flux de fonctionnement

- L'utilisateur envoie une requête via l'interface front-end.
- La requête est transmise au serveur d'application.
- Le serveur d'application traite la logique, interagit si nécessaire avec la base de données.
- Il renvoie une réponse (résultat) à l'interface front-end.

— L'utilisateur voit le résultat à l'écran.

4.2 Modèle Spring MVC

Dans cette section, nous présentons le Spring MVC. L'image ci-dessous illustre le modèle Spring MVC (Model-View-Controller), un modèle largement utilisé dans les applications web Java. L'objectif principal de ce modèle est de séparer les responsabilités entre la gestion des données (Modèle), l'affichage (Vue), et la logique de traitement (Contrôleur).

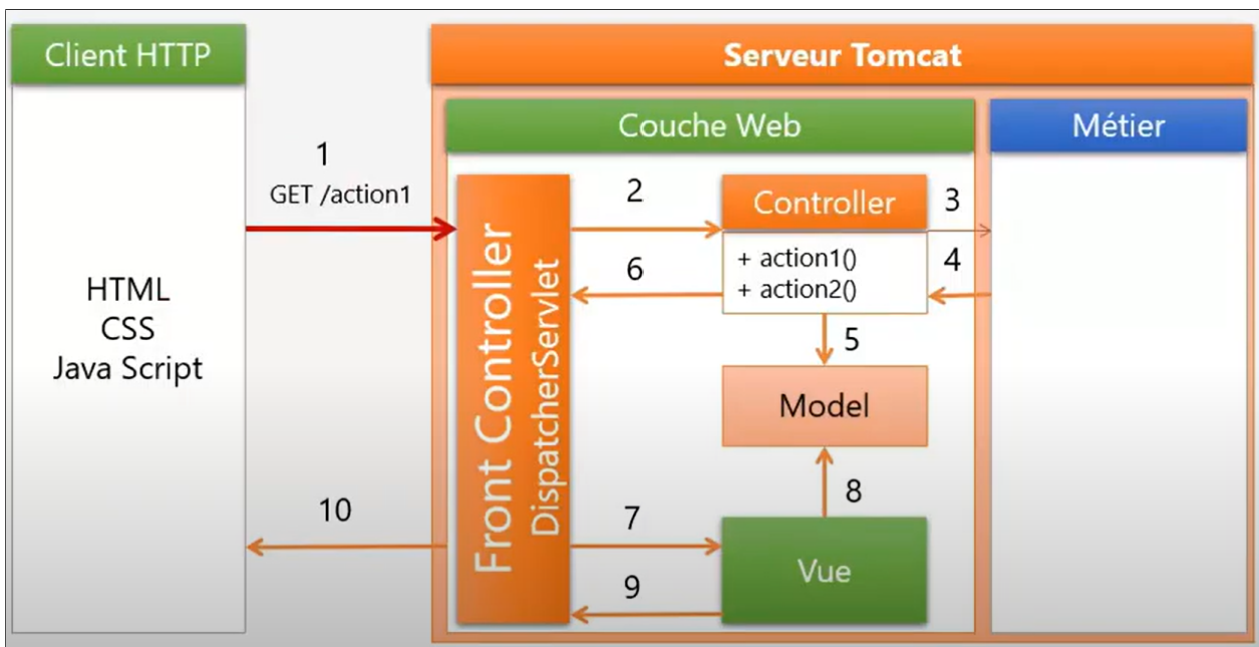


FIGURE 4.2 : Modèle spring MVC

4.2.1 Composants du modèle

- **Client HTTP** : Le navigateur ou autre client qui envoie une requête HTTP (GET, POST, etc.).
- **Front Controller (DispatcherServlet)** : Cœur du framework Spring MVC, il intercepte toutes les requêtes entrantes.

- Controller : Composant qui traite les requêtes, exécute la logique métier (souvent en appelant un service) et choisit la vue à retourner.
- Model : Données que l'application manipule (ex. : objets Java).
- Vue (View) : Interface utilisateur, généralement rendue via Thymeleaf, ou autre moteur de rendu.
- Serveur Tomcat : Conteneur de servlets qui exécute l'application web.

4.2.2 Étapes du traitement

- Requête du client : Le client envoie une requête HTTP (ex. : GET /action1) au serveur.
- Interception par DispatcherServlet : Le Front Controller (DispatcherServlet) intercepte cette requête.
- Redirection vers le bon contrôleur : Le DispatcherServlet analyse la requête et détermine quel Controller doit la gérer.
- Traitement métier : Le contrôleur peut faire appel à la couche métier (services, DAO...) pour exécuter la logique de l'application.
- Création/modification du modèle : Le résultat du traitement est stocké dans un objet Model (les données à afficher dans la vue).
- Retour du modèle au DispatcherServlet : Le contrôleur renvoie le modèle (et le nom de la vue) au DispatcherServlet.
- Sélection de la vue : Le DispatcherServlet choisit la vue appropriée (par exemple, une page JSP ou Thymeleaf).
- Transfert des données du modèle à la vue : Les données du modèle sont injectées dans la vue.

- Rendu de la vue : La vue est générée (HTML, CSS, JS) avec les données du modèle.
- Réponse au client : Le contenu HTML est renvoyé au client (navigateur), qui l’affiche à l’utilisateur.

4.2.3 Avantages de Spring MVC

- Séparation claire des responsabilités (Modèle / Vue / Contrôleur).
- Flexible et modulaire, chaque composant peut évoluer indépendamment.
- Intégration facile avec d’autres modules Spring (Spring Security, Spring Data, etc.).
- Gestion centralisée des requêtes grâce au DispatcherServlet.

4.3 Implémentation du site web

Dans cette section, nous présentons l’implémentation concrète de l’application web à travers une série de captures d’écran illustrant les principales fonctionnalités développées. Ces interfaces ont été conçues dans le respect des spécifications fonctionnelles définies lors de l’analyse des besoins. L’objectif est de montrer le rendu visuel du site, son ergonomie, ainsi que le fonctionnement des différentes fonctionnalités.

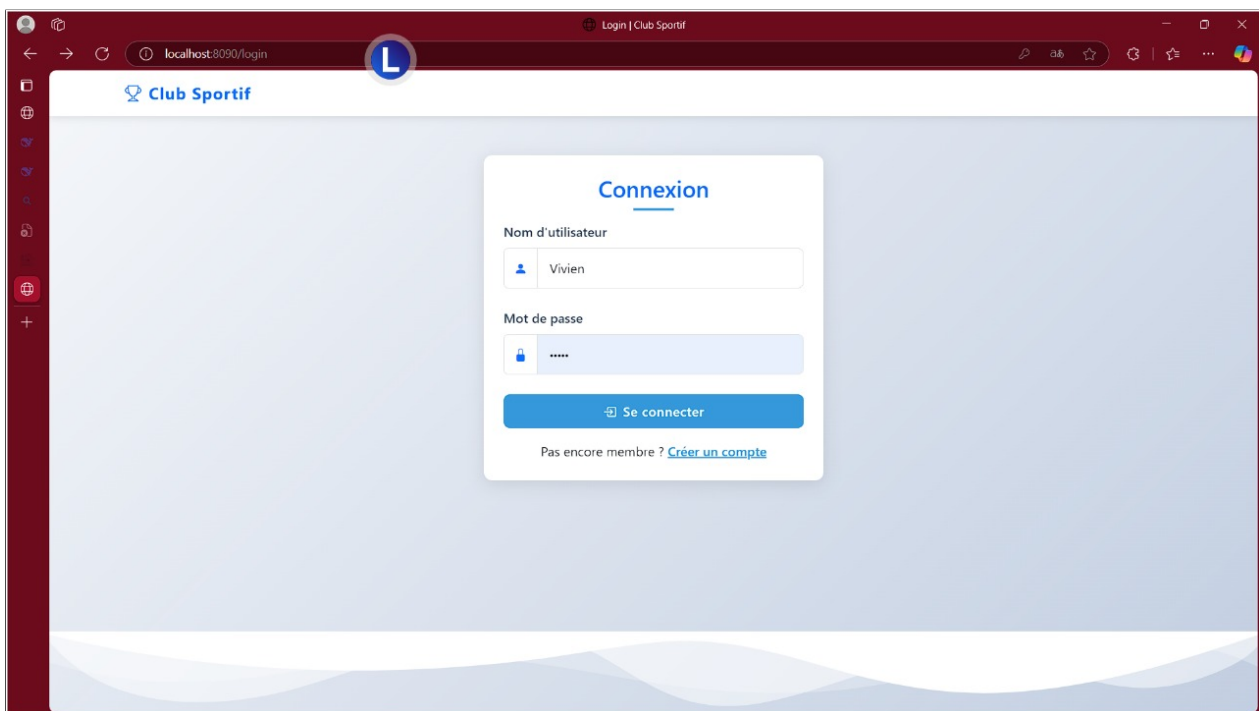


FIGURE 4.3 : Sign in

4.4 Déploiement sur Github

Afin d'assurer une gestion efficace du code source, le projet a été versionné et déployé sur la plateforme GitHub. Cette section décrit les étapes suivies pour l'initialisation du dépôt, le suivi des modifications, la collaboration éventuelle ainsi que les bonnes pratiques adoptées pour le contrôle de version.

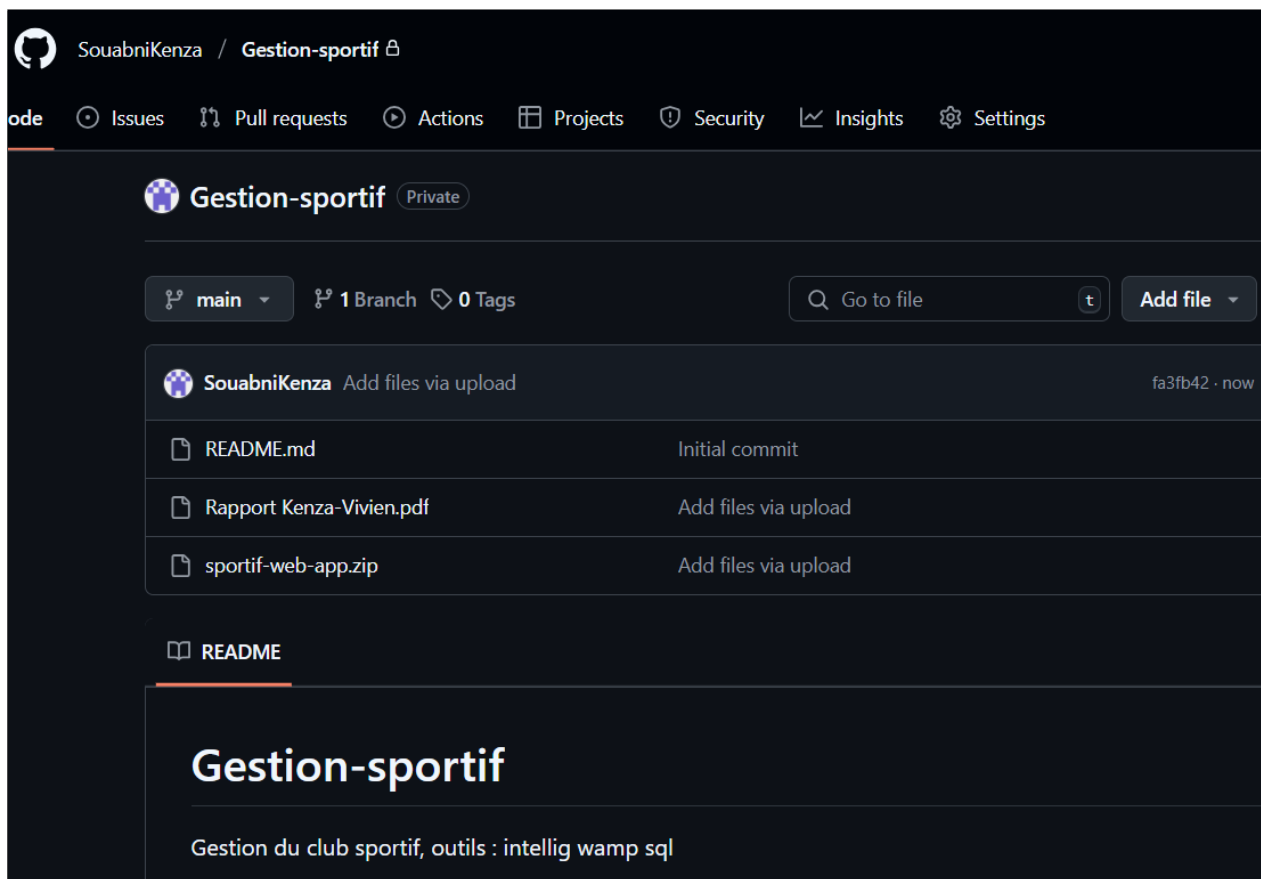


FIGURE 4.4 : Déploiement sur GitHub

Conclusion

Ce chapitre met l'accent sur l'architecture Spring MVC adoptée dans le projet, tout en présentant l'aboutissement du processus de développement du site web à travers les différentes interfaces réalisées. Il montre également le déploiement du projet sur GitHub, assurant ainsi la gestion du code source et la traçabilité du travail effectué.

Conclusion générale

Ce projet de développement d'une application web de gestion des clubs sportifs s'inscrit dans une démarche de modernisation et de transformation numérique du secteur sportif. En réponse aux nombreux défis auxquels sont confrontés les clubs, cette solution centralisée et intuitive apporte des outils concrets pour simplifier les processus, renforcer l'efficacité organisationnelle et améliorer l'expérience utilisateur.

Grâce à l'utilisation de technologies modernes telles que Java EE et Spring MVC, ainsi qu'à l'adoption de la méthodologie agile Scrum, le projet a permis de développer une application robuste, évolutive et adaptée aux besoins réels du terrain. L'intégration de fonctionnalités avancées, comme les tableaux de bord interactifs et la gestion en ligne des plannings et paiements, constitue un véritable atout pour les clubs en quête d'optimisation.

Sur le plan académique, ce projet a représenté une opportunité précieuse pour appliquer de manière concrète les connaissances acquises en gestion de projet, en conception d'interfaces et en développement web. Sur le plan professionnel, il répond à une problématique actuelle et réelle, offrant une solution innovante qui pourrait être facilement déployée et utilisée par des structures sportives de toutes tailles.

En somme, cette application contribue activement à la digitalisation du secteur sportif tout en constituant une expérience formatrice enrichissante pour les porteurs du projet. Elle pose les bases d'une évolution vers une gestion plus fluide, plus performante et résolument tournée vers l'avenir.

Abstract

Ce rapport présente le développement d'une application web destinée à la gestion des clubs sportifs. Conçue pour centraliser et automatiser les processus liés aux adhérents, activités, équipements et paiements, cette solution vise à améliorer l'efficacité administrative et l'expérience utilisateur. Réalisé avec les technologies Java EE et Spring MVC, le projet suit la méthodologie agile Scrum. Il s'inscrit dans un cadre académique et répond aux besoins croissants de digitalisation dans le secteur sportif.

This report presents the development of a web application for managing sports clubs. Designed to centralize and automate processes related to members, activities, equipment, and payments, the solution aims to enhance administrative efficiency and user experience. Built using Java EE and Spring MVC technologies, the project follows the Agile Scrum methodology. It was carried out in an academic context and addresses the growing need for digital transformation in the sports sector.