

Dédicace

Avec l'expression de ma toute reconnaissance, je dédie ce projet de fin d'études à

À mes chers parents

Mes héros ,vous êtes la lumière qui a guidé chacun de mes pas, vos sacrifices et votre dévouement sont des témoignages émouvant de votre amour ifini,vous avez sacrifié vos propres désirs pour me permettre de réaliser les miens, et pour cela, je vous serai éternellement reconnaissant.

À mes chers frère et sœur

Vous avez été mes anges gardiens, m'apportant réconfort et soutien dans les moments les plus sombres de ma vie. Votre présence, votre amour dévoué et votre tendresse ont été comme une lumière qui a dissipé les ténèbres de la solitude et de la souffrance. Je vous suis infiniment reconnaissant(e) d'être toujours à mes côtés, de partager vos cœurs généreux et votre amour inconditionnel. Merci d'avoir été mes compagnons fidèles

À tous les membres de ma famille

À ma précieuse famille, merci pour votre amour inconditionnel et votre soutien indéfectible à chaque étape de ma vie.

À tous mes enseignant du primaire à l'universitaire

Pour leurs efforts qui ont mené à ma réussite. Quels que soient les termes embrassés, je n'arriverais jamais à vous exprimer ma gratitude sincère.

À mes meilleur(e)s ami(e)s

Pour votre présence, vos encouragements et votre compréhension tout au long de ce parcours. Votre soutien moral, vos conseils avisés et votre camaraderie ont été des sources de motivation inestimables pour moi.

À tous ceux qui m'ont soutenu..

Table des matières

Introduction Générale	1
1 ÉTUDE PRÉALABLE	3
1.1 Introduction	4
1.2 Présentation générale de l'organisme d'accueil	5
1.2.1 Présentation de l'entreprise ExaDev	5
1.2.2 Services proposés	5
1.2.3 Localisation	6
1.2.4 L'organigramme de l'entreprise	7
1.2.5 Concepts de base : Développement mobile et web	8
1.3 Cadre général du projet	10
1.3.1 Problématique et motivation	10
1.3.2 Étude de l'existant	12
1.3.3 Solution proposée : ANAS' GYM	14
1.4 Méthodologie de travail	19
1.4.1 Rôles définis dans le projet	19
1.4.2 Processus Scrum adopté	20
1.4.3 Pilotage du projet avec Scrum	20
1.4.4 Planification détaillée des sprints	21
1.5 Conclusion	23
2 SPÉCIFICATION DES BESOINS ET ANALYSE	24
2.1 Introduction	25
2.2 Identification des acteurs	25
2.3 Besoins fonctionnels	25
2.3.0.0.1 Authentification et profil	25
2.3.0.0.2 Gestion des workouts	26

2.3.0.0.3	Programmes d'entraînement	26
2.3.0.0.4	Suivi de la progression (UserProgress)	26
2.3.0.0.5	Branches, équipements et coachs	26
2.3.0.0.6	Séances collectives (GroupTrainingSession)	26
2.3.0.0.7	Gestion des branches	27
2.3.0.0.8	Gestion des équipements	27
2.3.0.0.9	Gestion des exercices, workouts et programmes	27
2.3.0.0.10	Gestion des utilisateurs	27
2.3.0.0.11	Gestion des coachs	27
2.3.0.0.12	Gestion des séances collectives	27
2.3.0.0.13	Paramètres publics	28
2.4	Besoins non fonctionnels	28
2.5	Langage de modélisation	28
2.6	Diagramme de cas d'utilisation	29
2.7	Exemples de cas d'utilisation	30
2.7.1	Créer un workout	30
2.7.2	Gérer les utilisateurs	30
2.7.3	Suivre la progression	31
2.8	Conclusion	31
3	ÉTUDE CONCEPTUELLE	32
3.1	Introduction	32
3.2	Outils et Frameworks	32
3.3	Architecture du projet	33
3.4	Diagramme de classes	34
3.5	Diagrammes de Use Case	35
3.6	Conclusion	36
4	RÉALISATION GLOBALE DE LA PLATEFORME	37
4.1	Introduction	38
4.2	Environnement de développement	38
4.2.1	Outils utilisés	38
4.2.2	Structure globale du dépôt	38
4.3	Réalisation du backend (Laravel + Sanctum + MySQL)	39
4.3.1	Conception et création de la base de données	39

4.3.2	Seeders et données initiales	39
4.3.3	Authentification avec Laravel Sanctum	40
4.3.4	Implémentation des endpoints REST	40
4.3.5	Tests via Thunder Client	41
4.4	Réalisation de l'application mobile (React Native + Expo)	41
4.4.1	Structure et navigation	41
4.4.2	Connexion à l'API	41
4.4.3	Fonctionnalités principales réalisées	42
4.4.3.0.1	Gestion des workouts	42
4.4.3.0.2	Branches et équipements	42
4.4.3.0.3	Consultation des coachs	42
4.4.3.0.4	Sessions collectives	42
4.4.3.0.5	Progression	42
4.5	Réalisation du dashboard administrateur (React)	42
4.5.1	Structure générale	43
4.5.2	Authentification administrateur	43
4.5.3	Fonctionnalités réalisées	43
4.5.3.0.1	Gestion des ressources	43
4.5.3.0.2	Gestion des coachs	43
4.5.3.0.3	Sessions collectives	43
4.6	Conclusion	44
	Conclusion Générale	44
	Webographie	48

Table des figures

1.1	Logo de l'entreprise ExaDev	5
1.2	Organigramme de l'entreprise ExaDev	7
1.3	Architecture globale de la solution ANAS' GYM	18
1.4	Processus SCRUM appliqué au projet	20
2.1	StarUML	28
2.2	Diagramme de cas d'utilisation de la plateforme	29
3.1	Diagramme de classes de l'application MyGym	34
3.2	Diagramme général de Use Case de la plateforme	36

Liste des tableaux

1.1	Planification détaillée des sprints du projet ANAS' GYM	21
2.1	Cas d'utilisation — Créer un workout	30
2.2	Cas d'utilisation — Gérer les utilisateurs	30
2.3	Cas d'utilisation — Suivre la progression	31

Introduction Générale

Dans le contexte actuel où la pratique sportive et le suivi de la santé deviennent des priorités croissantes, les membres des salles de sport rencontrent souvent des difficultés majeures pour gérer efficacement leurs entraînements, suivre leur progression physique et accéder aux informations essentielles sur les équipements et les coachs disponibles. En effet, les programmes quotidiens ne sont pas historisés, les données corporelles se perdent facilement, et la réservation des cours collectifs reste complexe et source d'oubli. De plus, la multiplicité des locaux et l'absence d'un système centralisé compliquent considérablement l'accès aux informations nécessaires pour une pratique sportive optimale.

Le présent rapport synthétise le travail réalisé lors de mon **stage d'été de 2^e année** au sein de l'entreprise ExaDev à Sfax, visant à concevoir et développer une plateforme complète de gestion pour salles de sport multi-sites. Ce projet, nommé **ANAS' GYM** (nom de code interne : MyGym), comprend une application mobile destinée aux membres (développée avec React Native et Expo) et un tableau de bord administrateur (développé avec React), tous deux adossés à une API robuste Laravel sécurisée par Sanctum.

L'objectif principal de cette plateforme est de centraliser l'ensemble des fonctionnalités nécessaires à une gestion moderne et efficace des salles de sport : la gestion des entraînements personnalisés, la réservation simplifiée de cours collectifs, le suivi détaillé des progrès physiques, la consultation des profils des coachs (gérés côté administration et consultables par les membres), ainsi que la gestion complète des équipements disponibles dans chaque local. Cette solution vise à transformer l'expérience utilisateur en offrant un accès unifié, intuitif et fiable à toutes ces informations critiques.

Le développement de cette plateforme s'est articulé autour de trois volets complémentaires. Premièrement, la conception d'une interface mobile intuitive et ergonomique pour les membres, leur permettant d'accéder aux fonctionnalités de création et d'historisation des workouts, de suivi de progression, de consultation des coachs et des équipements, ainsi que de réservation des cours collectifs, le tout accompagné d'un système de notifications. Deuxièmement, le développement d'un dashboard administrateur web permettant la gestion centralisée de tous les locaux, des équipements (machines, charges, catégories, mouvements) et des séances d'entraînement collectives. Troisièmement, l'intégration d'un système robuste d'historisation et de stockage sécurisé des données liées aux entraînements quotidiens (workouts), aux programmes personnalisés, aux mesures corporelles et aux paramètres publics de chaque salle.

Ce stage a permis de mettre en pratique et d'approfondir des compétences techniques

variées et actuelles : le développement mobile multiplateforme avec **React Native et Expo**, le développement web moderne avec **React**, ainsi que la conception et l'implémentation d'API RESTful sécurisées avec **Laravel et Sanctum**. L'architecture adoptée garantit la scalabilité, la sécurité des données sensibles et la maintenabilité à long terme du système.

Le présent rapport expose de manière détaillée les différentes étapes de réalisation du projet ANAS' GYM, depuis l'étude préalable et l'analyse rigoureuse des besoins fonctionnels et non fonctionnels, jusqu'à la conception architecturale, le développement technique et la mise en œuvre opérationnelle de la solution.

Notre travail est structuré en cinq chapitres cohérents et progressifs. Le premier chapitre présente l'entreprise d'accueil ExaDev, son organisation et le contexte général du projet, en exposant la problématique rencontrée et les motivations à l'origine de cette solution. Le second chapitre expose l'étude préalable et l'analyse approfondie des besoins des différents acteurs, ainsi que les solutions existantes sur le marché, avant de justifier les choix technologiques retenus. Le troisième chapitre est consacré à l'étude conceptuelle complète, incluant la spécification fonctionnelle, la modélisation UML des besoins (diagrammes de cas d'utilisation et de classes) et l'architecture technique détaillée. Le quatrième chapitre détaille la **réalisation globale** de la plateforme, couvrant le développement du backend Laravel (base de données, authentification, API), de l'application mobile React Native, ainsi que du dashboard administrateur React. Le cinquième chapitre présente les tests, l'intégration finale et le déploiement de la solution. Enfin, une conclusion générale synthétise les résultats obtenus et ouvre sur les perspectives d'évolution du système.

ÉTUDE PRÉALABLE

Plan

1.1	Introduction	4
1.2	Présentation générale de l'organisme d'accueil	5
1.2.1	Présentation de l'entreprise ExaDev	5
1.2.2	Services proposés	5
1.2.3	Localisation	6
1.2.4	L'organigramme de l'entreprise	7
1.2.5	Concepts de base : Développement mobile et web	8
1.3	Cadre général du projet	10
1.3.1	Problématique et motivation	10
1.3.2	Étude de l'existant	12
1.3.3	Solution proposée : ANAS' GYM	14
1.4	Méthodologie de travail	19
1.4.1	Rôles définis dans le projet	19
1.4.2	Processus Scrum adopté	20
1.4.3	Pilotage du projet avec Scrum	20
1.4.4	Planification détaillée des sprints	21
1.5	Conclusion	23

1.1 Introduction

Ce chapitre explore notre projet ANAS' GYM à travers plusieurs perspectives clés, en commençant par une présentation détaillée de l'organisme d'accueil, ExaDev, suivie d'un aperçu général de notre plateforme complète pour la gestion et l'amélioration de l'expérience des membres dans les salles de sport multi-sites.

Dans un premier temps, nous présenterons l'entreprise ExaDev, son organisation structurelle, ainsi que les différentes entités qui composent cette organisation dynamique, avec un accent particulier sur le pôle de développement, cadre principal de notre stage. Nous aborderons également les concepts fondamentaux nécessaires à la compréhension approfondie de notre projet, tels que le développement mobile multiplateforme, le développement web moderne, l'architecture client-serveur RESTful, ainsi que les technologies spécifiques utilisées (React Native avec Expo, React pour le web, et Laravel avec Sanctum pour le backend).

Dans un second temps, nous plongerons dans le cadre général de notre projet ANAS' GYM en identifiant de manière précise les problèmes rencontrés par les membres des salles de sport et les gestionnaires : l'absence d'historisation structurée des entraînements quotidiens, la perte fréquente des données corporelles essentielles, la difficulté à réserver et se rappeler des cours collectifs, l'inaccessibilité des informations détaillées sur les coachs et leurs disponibilités, le manque de visibilité sur les équipements disponibles par local, ainsi que la complexité inhérente à la gestion de plusieurs locaux sans système centralisé. Nous exposerons également la motivation profonde qui nous a conduit à développer cette application innovante et réaliserons une analyse comparative des solutions existantes sur le marché local et international.

Nous présenterons ensuite la solution complète proposée, qui se compose d'une application mobile ergonomique pour les membres, d'un tableau de bord administrateur web centralisé et d'une API backend robuste. Cette solution intégrée permet d'historiser précisément les séances d'entraînement avec tous leurs détails (mouvements, machines, charges, séries, répétitions), de suivre la progression corporelle des membres avec visualisations graphiques, de gérer efficacement les réservations de cours collectifs avec notifications, de consulter les informations complètes sur les coachs (spécialités, disponibilités, vidéos), de visualiser les équipements disponibles par salle, tout en centralisant toutes les données pour l'ensemble des locaux via un dashboard unique. Nous détaillerons également la méthodologie de travail agile adoptée durant le stage, inspirée des principes Scrum, permettant une planification itérative et un suivi efficace du développement par sprints.

Enfin, ce chapitre établit les fondations nécessaires pour une compréhension approfondie

de notre projet, tout en mettant en lumière les défis techniques et organisationnels rencontrés ainsi que les choix stratégiques effectués pour répondre de manière optimale aux besoins des utilisateurs finaux et de l'entreprise cliente.

1.2 Présentation générale de l'organisme d'accueil

1.2.1 Présentation de l'entreprise ExaDev

ExaDev est une société tunisienne spécialisée dans les services informatiques, créée le 30 décembre 2018 et basée à Sfax. Elle fournit une gamme complète de services informatiques conçus pour améliorer la productivité et l'efficacité opérationnelle des entreprises clientes. ExaDev intervient depuis l'installation experte de solutions logicielles, l'externalisation de services IT, jusqu'aux solutions hébergées dans le cloud, avec pour objectif stratégique d'accompagner ses clients dans leur transformation digitale et l'optimisation de leurs processus métiers.

La société vise à proposer des solutions complètes, personnalisées et sur mesure, parfaitement adaptées aux besoins spécifiques de chaque entreprise, que ce soit en Tunisie ou à l'international. ExaDev se positionne comme un partenaire technologique de confiance, capable d'accompagner ses clients tout au long de leur parcours de digitalisation.



FIGURE 1.1 – *Logo de l'entreprise ExaDev*

1.2.2 Services proposés

ExaDev offre un ensemble de services diversifiés et complémentaires pour répondre aux besoins numériques variés des entreprises :

Graphisme et Design

- Création d'identités visuelles complètes et de chartes graphiques professionnelles

- Ergonomie et design d’interfaces utilisateur modernes et intuitives
- Design d’interaction et conception d’expérience utilisateur (UX/UI) optimisée
- Prototypage et wireframing pour valider les concepts avant développement

Développement Web

- Conception de sites web vitrines modernes et entièrement responsives
- Développement de solutions e-commerce personnalisées avec systèmes de paiement sécurisés
- Applications web sur mesure avec intégration de bases de données complexes
- Développement de plateformes web collaboratives et portails d’entreprise
- Optimisation SEO et performances web

Développement d’applications mobiles

- Applications natives iOS et Android haute performance
- Applications hybrides multiplateforme (React Native, Flutter)
- Développement d’applications métiers et de plateformes de gestion mobile
- Maintenance évolutive et support technique continu des applications
- Intégration avec des API tierces et services cloud

Solutions informatiques et accompagnement

- Conseil stratégique en transformation digitale
- Externalisation de services IT et infogérance
- Mise en place de systèmes ERP (Enterprise Resource Planning)
- Solutions cloud et hébergement sécurisé
- Audit de sécurité et mise en conformité RGPD

1.2.3 Localisation

ExaDev est située au cœur de Sfax, à l’adresse suivante : Avenue de la Liberté, Immeuble Ayda, 1^{er} étage, Bureau N°105, Sfax 3000, Tunisie. Cette localisation stratégique facilite l’accès pour les clients locaux tout en permettant une collaboration efficace avec des partenaires internationaux.

1.2.4 L'organigramme de l'entreprise

L'organisation d'ExaDev est structurée de manière hiérarchique et fonctionnelle autour de plusieurs pôles spécialisés, garantissant une expertise pointue dans chaque domaine d'intervention. Ces pôles comprennent notamment le pôle développement (backend, frontend, mobile), le pôle design et ergonomie (UX/UI), le pôle marketing et communication, le pôle support et maintenance technique, ainsi que le pôle ressources humaines et administration. Cette organisation matricielle favorise la collaboration transversale et l'efficacité opérationnelle.

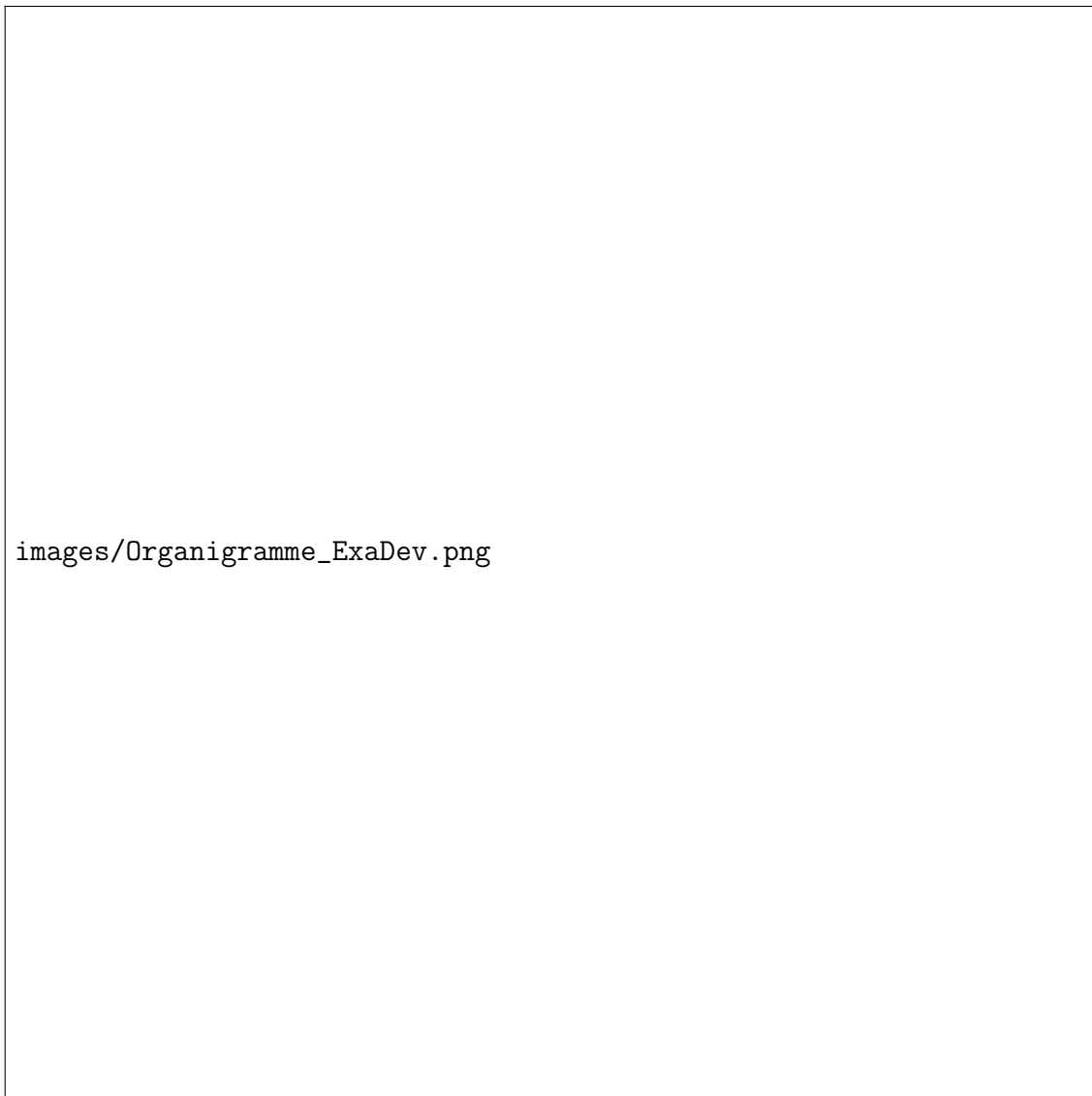


FIGURE 1.2 – *Organigramme de l'entreprise ExaDev*

La réalisation de notre projet ANAS' GYM s'est déroulée principalement au sein du pôle développement, en étroite coopération avec le pôle support pour les aspects techniques et

le pôle design pour assurer l’ergonomie optimale et l’expérience utilisateur cohérente sur l’ensemble des interfaces (mobile et web).

1.2.5 Concepts de base : Développement mobile et web

Le projet ANAS’ GYM s’appuie sur un ensemble de technologies modernes et éprouvées de développement mobile et web, permettant de créer une application performante, évolutive et maintenable à long terme.

React Native avec Expo

React Native est un framework JavaScript développé par Facebook permettant de créer des applications mobiles natives pour iOS et Android à partir d’une base de code unique. Expo est une plateforme et un ensemble d’outils qui simplifient considérablement le développement React Native en fournissant des API prêtes à l’emploi, un système de build dans le cloud et la possibilité de tester en temps réel sur des appareils physiques via l’application Expo Go.

Les avantages de cette combinaison incluent : le développement multiplateforme avec une seule base de code, le rechargement à chaud pour un développement rapide, l’accès simplifié aux fonctionnalités natives (caméra, géolocalisation, notifications push), et une communauté active avec de nombreuses bibliothèques tierces.

React pour le web

React est une bibliothèque JavaScript développée par Facebook pour construire des interfaces utilisateur interactives et composables. Utilisée pour le tableau de bord administrateur de ANAS’ GYM, React permet de créer des interfaces web modernes, réactives et performantes grâce à son système de composants réutilisables et son Virtual DOM optimisé.

Les points forts de React incluent : une architecture basée sur des composants modulaires, un rendu optimal grâce à la réconciliation du Virtual DOM, une gestion d’état facilitée (hooks, context), un écosystème riche (React Router, state management) et une excellente compatibilité avec les outils de build modernes (Webpack, Vite).

Laravel avec Sanctum

Laravel est un framework PHP robuste et élégant pour le développement backend d’applications web. Il est utilisé dans notre projet pour créer une API RESTful sécurisée et performante. Laravel Sanctum fournit un système d’authentification léger basé sur des tokens

API, parfait pour les applications SPA (Single Page Applications) et mobiles.

Les fonctionnalités clés exploitées incluent : l'ORM Eloquent pour une gestion élégante de la base de données, le système de routing flexible, les migrations et seeders pour versionner le schéma de base de données, la validation robuste des requêtes, l'authentification sécurisée via tokens Sanctum, et le stockage de fichiers avec gestion des médias.

Architecture et intégration

L'architecture globale du système ANAS' GYM repose sur une séparation claire entre le backend (API Laravel) et les frontends (React Native mobile + React web), communiquant via des endpoints REST sécurisés. Cette architecture offre plusieurs avantages stratégiques :

- **Historisation et suivi précis** : Collecte structurée et stockage sécurisé de toutes les données d'entraînement (workouts, exercices, charges) et des mesures corporelles des utilisateurs avec traçabilité complète
- **Gestion centralisée multi-sites** : Possibilité de gérer efficacement plusieurs locaux (branches) et leurs cours collectifs depuis un tableau de bord administrateur unique
- **Système de réservation et notifications** : Planification intuitive des cours collectifs avec rappels automatiques par notifications push et confirmation instantanée
- **Accessibilité et ergonomie optimales** : Interfaces utilisateur intuitives et modernes pour les membres et le personnel administratif, optimisées pour le mobile et le web
- **Sécurité renforcée** : Authentification par tokens, hachage des mots de passe, validation stricte des données, contrôle d'accès basé sur les rôles

Domaines d'application du projet ANAS' GYM

Le projet ANAS' GYM trouve son utilité dans plusieurs domaines essentiels liés à la gestion moderne des salles de sport :

- **Historisation complète des entraînements** : Suivi détaillé et chronologique de tous les mouvements effectués, machines utilisées, charges levées, nombre de séries et répétitions, avec possibilité de consulter l'historique complet
- **Suivi personnalisé de la progression corporelle** : Enregistrement régulier et visualisation graphique de l'évolution du poids, des mensurations (tour de taille, tour de bras, tour de cuisses), de la masse musculaire et de l'IMC

- **Gestion intelligente des coachs et des cours** : Consultation des profils détaillés des coachs (spécialités, expérience, disponibilités), réservation simplifiée de sessions individuelles et de cours collectifs
- **Administration multi-locaux efficace** : Centralisation de toutes les informations pour chaque salle (horaires, équipements, coachs) et intégration d'un tableau de bord administrateur unique pour une gestion globale
- **Accès facilité aux équipements** : Fiches techniques détaillées pour chaque machine ou mouvement avec instructions, photos et vidéos pour guider efficacement l'utilisateur
- **Gestion des paramètres publics** : Configuration centralisée des informations de marque (nom de l'application, messages d'accueil, horaires d'ouverture, coordonnées, liens réseaux sociaux)

En résumé, ExaDev fournit le cadre technique, organisationnel et humain nécessaire pour le développement réussi de ANAS' GYM, tout en mettant en œuvre des solutions numériques innovantes et adaptées aux besoins spécifiques des utilisateurs finaux (membres, coachs) et des gestionnaires de salles de sport modernes.

1.3 Cadre général du projet

La pratique sportive régulière et la gestion professionnelle des salles de fitness sont devenues des enjeux sociétaux majeurs dans notre société moderne, où la santé et le bien-être occupent une place centrale. Avec l'essor continu du numérique et des technologies mobiles, les sportifs recherchent activement des outils innovants leur permettant de suivre efficacement leurs performances et leur progression, tandis que les gestionnaires de salles de sport ont besoin de solutions digitales robustes pour centraliser et optimiser l'ensemble de leurs activités opérationnelles.

Dans cette section, nous présentons de manière détaillée la problématique actuelle de la gestion des salles et des entraînements, les solutions existantes sur le marché avec leurs limites respectives, puis la solution innovante et complète que nous proposons à travers la plateforme ANAS' GYM.

1.3.1 Problématique et motivation

Aujourd'hui, la plupart des salles de sport en Tunisie et dans la région rencontrent plusieurs difficultés majeures liées à la gestion quotidienne et au suivi des membres, qui impactent

directement la qualité du service et la satisfaction client :

Manque de centralisation des informations

Les informations essentielles concernant les adhérents, leurs historiques d'entraînement, les profils des coaches, les caractéristiques des machines et équipements, ainsi que les planings des cours collectifs sont souvent éparpillées entre plusieurs outils hétérogènes et non connectés : fichiers Excel vieillissants, applications mobiles non synchronisées, carnets papier traditionnels, tableaux blancs physiques. Cette fragmentation crée des incohérences, des pertes d'information et une complexité opérationnelle inutile.

Absence de suivi personnalisé et historisation

Les clients ne disposent généralement pas d'un historique clair, structuré et accessible de leurs entraînements passés, de leurs progrès corporels mesurables (évolution du poids, des mensurations, de la masse musculaire, de l'IMC), ni de recommandations personnalisées adaptées à leurs objectifs spécifiques. Les programmes d'entraînement quotidiens ne sont pas sauvegardés de manière systématique, les fiches de progression physique se perdent facilement, et il devient difficile de mesurer objectivement les résultats obtenus sur le moyen et long terme.

Organisation complexe des cours collectifs

La planification et la gestion des sessions collectives (Zumba, Yoga, CrossFit, etc.) restent particulièrement complexes et sujettes à de nombreuses erreurs humaines : conflits d'horaires entre les coaches, oublis de réservations par les membres, dépassements de capacité des salles, absence de rappels automatiques, difficulté à identifier les sessions spécifiques (100% femmes, enfants, gratuites ou payantes). Cette désorganisation entraîne des annulations de dernière minute, des frustrations clients et une sous-utilisation des créneaux disponibles.

Expérience utilisateur limitée et peu engageante

Les applications et outils existants manquent souvent d'ergonomie moderne, d'intuitivité et d'intégration mobile fluide, ce qui freine considérablement leur adoption par les utilisateurs finaux (membres et staff). Les interfaces sont vieillissantes, peu attrayantes, et n'exploitent pas les possibilités des smartphones modernes (notifications push, géolocalisation, appareil photo pour le suivi visuel). De plus, l'absence d'une application mobile dédiée oblige les membres à se déplacer physiquement pour obtenir des informations basiques.

Gestion multi-sites fragmentée

Pour les chaînes de salles possédant plusieurs locaux, l'absence d'un système centralisé rend la gestion globale extrêmement difficile : impossibilité d'avoir une vue d'ensemble en temps réel, duplication des efforts administratifs, incohérence des données entre les sites, difficulté à transférer les membres d'un local à un autre, reporting consolidé inexistant ou réalisé manuellement avec risques d'erreurs.

Ces problèmes multiples entraînent des conséquences directes et mesurables : perte de temps considérable pour le personnel administratif et les coaches, manque d'efficacité organisationnelle globale, erreurs fréquentes dans la planification et la facturation, et surtout une expérience utilisateur insatisfaisante qui peut conduire à l'attrition des membres. Notre motivation principale est donc de proposer une application mobile et web moderne, intégrée et ergonomique, permettant aux salles de sport de gérer efficacement l'ensemble de leurs activités tout en offrant aux clients des outils numériques puissants et centralisés pour améliorer significativement leur expérience sportive et leurs résultats.

1.3.2 Étude de l'existant

Avant de développer la solution ANAS' GYM, nous avons réalisé une étude comparative approfondie de plusieurs solutions déjà présentes sur le marché tunisien et international, afin d'identifier leurs points forts et leurs limitations respectives.

Deciplus

Deciplus est une solution de gestion de clubs de sport utilisée par plusieurs établissements en Tunisie. Elle propose principalement la gestion des abonnements, des plannings d'activités et des réservations en ligne.

Points forts identifiés :

- Centralisation efficace des adhésions et de la facturation
- Gestion correcte des cours collectifs avec planning intégré
- Système de reporting de base pour les statistiques d'affluence

Limites constatées :

- Interface utilisateur parfois complexe et peu intuitive, nécessitant une formation
- Personnalisation limitée pour les besoins spécifiques des salles locales

- Absence de fonctionnalités avancées de suivi sportif individuel (historique détaillé des entraînements, charges, machines)
- Application mobile basique avec fonctionnalités réduites
- Coût d'abonnement élevé pour les petites structures

MyClub

MyClub est une application plus récente qui permet aux adhérents de réserver des cours et de gérer leurs abonnements de manière autonome via leur smartphone.

Avantages reconnus :

- Interface simple et moderne pour les membres
- Intégration fluide avec les systèmes de facturation
- Notifications push pour les rappels de cours

Inconvénients majeurs :

- Absence totale de fonctionnalités avancées de suivi sportif (historique des entraînements, progression des charges, machines utilisées)
- Pas de gestion des programmes personnalisés ni des workouts
- Fonctionnalités limitées pour les coachs (pas de suivi des clients)
- Dashboard administrateur basique avec peu d'analytics

PerfectGym

PerfectGym est une solution internationale très complète et professionnelle, utilisée par de grandes chaînes de fitness. Elle offre un large éventail de fonctionnalités couvrant tous les aspects de la gestion.

Avantages significatifs :

- Fonctionnalités extrêmement riches et professionnelles
- Système de reporting et d'analytics avancé avec KPIs détaillés
- Intégrations multiples avec des outils tiers (CRM, comptabilité, marketing)
- Support client international et mises à jour régulières

Limites importantes :

- Coût très élevé et hors de portée des petites et moyennes salles locales
- Complexité d'implémentation et de configuration initiale importante

- Interface parfois surchargée pour des besoins simples
- Nécessite une formation approfondie du personnel
- Peu adapté aux spécificités du marché tunisien (langue, devises, habitudes locales)

Synthèse de l'analyse comparative

Ces solutions existantes répondent partiellement aux besoins identifiés, mais elles présentent toutes des limitations significatives. La plupart sont soit trop coûteuses pour le marché local, soit trop générales et non adaptées aux attentes spécifiques des salles de sport tunisiennes et de leurs adhérents, soit limitées en termes de fonctionnalités de suivi sportif détaillé. Il existe donc un besoin réel et non satisfait pour une solution à la fois abordable, complète, moderne et parfaitement adaptée au contexte local.

1.3.3 Solution proposée : ANAS' GYM

Face aux limites des solutions existantes et aux besoins clairement identifiés, nous proposons **ANAS' GYM** (nom de code : MyGym), une plateforme mobile et web complète et intégrée, dédiée à la gestion moderne des salles de sport et au suivi personnalisé et détaillé des membres.

Notre solution apporte un ensemble cohérent et complet de fonctionnalités innovantes répondant à tous les besoins identifiés :

Gestion complète des adhérents

- Création et maintenance des profils utilisateurs détaillés avec informations personnelles, objectifs sportifs, contraintes médicales éventuelles
- Historique complet des abonnements avec dates, types, renouvellements et paiements
- Suivi de l'assiduité et des statistiques de fréquentation par membre
- Système de rôles et permissions (membre, coach, staff, administrateur)

Historisation détaillée des entraînements

- Création et sauvegarde automatique des workouts quotidiens avec tous les détails : mouvements effectués, machines utilisées, charges levées, nombre de séries et répétitions, temps de repos
- Visualisation chronologique et graphique des performances passées

- Comparaison de la progression sur des périodes définies (semaine, mois, trimestre)
- Possibilité de marquer un workout comme accompli ou de le reprogrammer

Suivi corporel et progression physique

- Enregistrement régulier des mesures corporelles : poids, tour de taille, tour de bras, tour de cuisses, tour de hanches
- Calcul automatique de l'IMC (Indice de Masse Corporelle) et de son évolution
- Visualisation graphique claire de l'évolution de toutes les métriques dans le temps
- Possibilité d'ajouter des photos de progression (before/after)
- Statistiques de progression personnalisées

Gestion intelligente des coaches

- Profils détaillés des coaches avec photo, biographie, certifications, spécialités (musculature, cardio, nutrition, rééducation)
- Visualisation des disponibilités hebdomadaires de chaque coach avec créneaux horaires
- Système de rating et avis des membres sur les coaches
- Possibilité de réserver une session individuelle avec un coach spécifique
- Portfolio de vidéos d'exercices et conseils par coach

Gestion complète des cours collectifs

- Planning interactif et visuel de toutes les sessions collectives (Zumba, Yoga, CrossFit, etc.)
- Réservation en ligne simple et rapide avec confirmation immédiate
- Annulation possible dans les délais définis avec libération automatique de la place
- Affichage des sessions spéciales avec filtres : 100% femmes, cours pour enfants, sessions gratuites ou payantes
- Gestion automatique des capacités maximales par session avec liste d'attente
- Notifications push de rappel 24h et 1h avant la session
- Historique complet des réservations passées et statistiques de participation

Gestion multi-salles centralisée

- Administration de plusieurs branches (locaux) depuis un tableau de bord unique et unifié
- Configuration spécifique par salle : horaires d'ouverture, jours de fermeture, coordonnées, équipements disponibles
- Affectation des coachs et du staff par branche
- Reporting consolidé global et par branche avec statistiques détaillées
- Possibilité pour les membres de basculer entre les différents locaux

Catalogue des équipements

- Fiches détaillées pour chaque machine avec nom, type, description complète, photo et vidéo explicative
- Liste des charges disponibles par machine (disques, haltères, bandes élastiques)
- Catégorisation des équipements (cardio, musculation, fonctionnel, étirement)
- Instructions de sécurité et conseils d'utilisation pour chaque machine
- Statut de disponibilité en temps réel (disponible, en maintenance)

Programmes d'entraînement personnalisés

- Création de programmes structurés sur plusieurs semaines avec objectifs définis
- Association de workouts spécifiques à chaque jour de la semaine
- Activation/désactivation de programmes selon les besoins
- Suivi de la complétion du programme avec statistiques de réalisation
- Possibilité de dupliquer et adapter des programmes existants

Système de notifications et rappels

- Notifications push automatiques pour les réservations de cours confirmées
- Rappels programmés avant les sessions collectives (24h, 1h)
- Alertes pour les renouvellements d'abonnement
- Notifications personnalisées pour encourager la régularité des entraînements
- Historique des notifications envoyées et statut de lecture

Paramètres publics configurables

- Configuration centralisée du nom de l'application et des messages d'accueil personnalisés
- Gestion des horaires globaux et par branche
- Coordonnées complètes (adresse, téléphone, email) par local
- Liens vers les réseaux sociaux (Facebook, Instagram, YouTube)
- Textes d'accueil et présentation de la salle personnalisables

Architecture technique de la solution

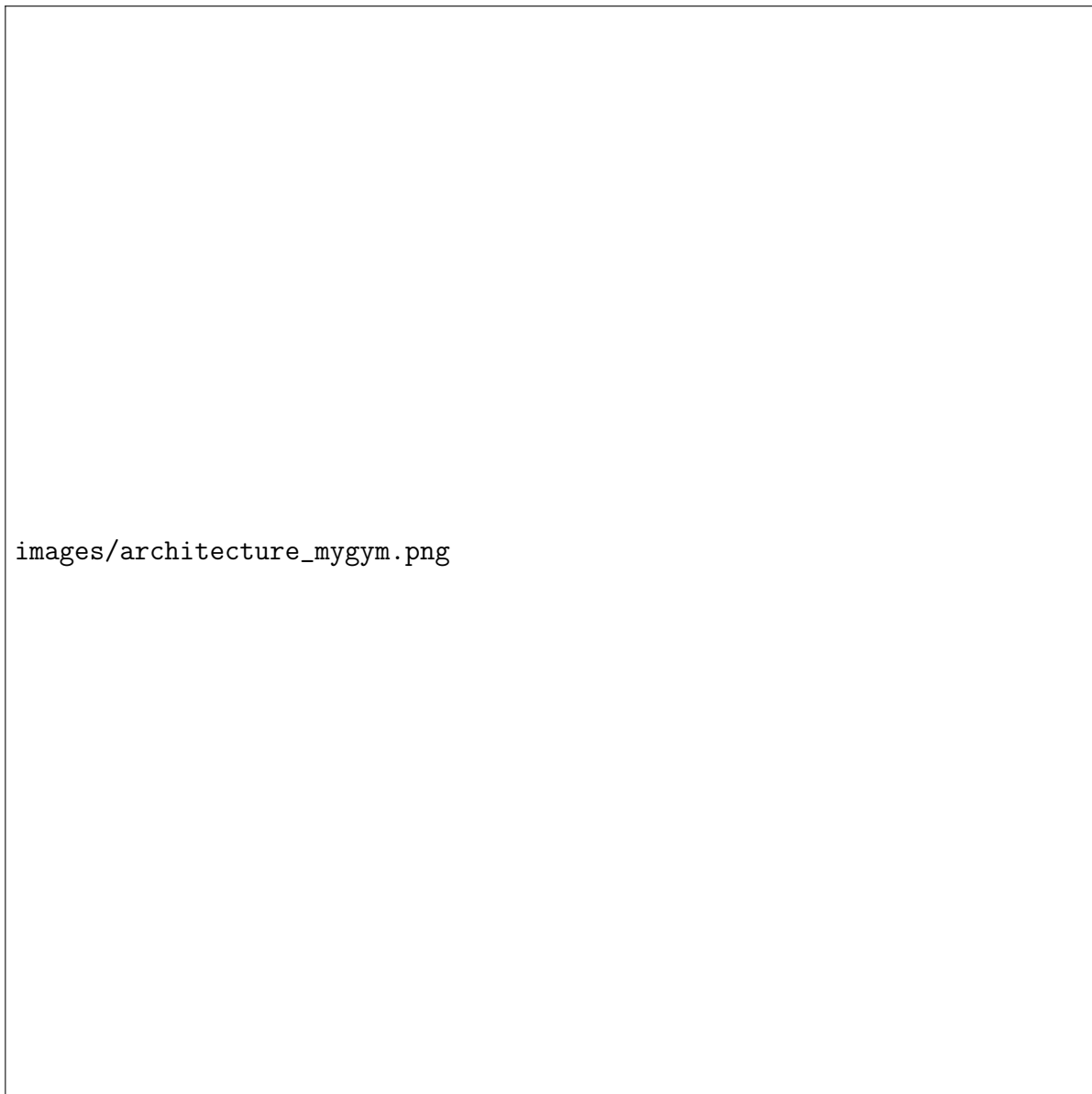


FIGURE 1.3 – *Architecture globale de la solution ANAS' GYM*

L'architecture technique de ANAS' GYM repose sur une séparation claire et modulaire entre trois composants principaux :

1. **Backend Laravel (API REST)** : Serveur central exposant des endpoints sécurisés via Sanctum, gérant toute la logique métier, l'accès à la base de données MySQL, la validation des données, l'authentification et les autorisations par rôle.
2. **Frontend Mobile (React Native + Expo)** : Application mobile native pour iOS et Android, offrant une expérience utilisateur fluide et moderne aux membres avec

accès à toutes les fonctionnalités (workouts, réservations, progression, consultation coaches/équipements).

3. **Frontend Web (React)** : Dashboard administrateur responsive accessible via navigateur, permettant la gestion complète des ressources (branches, coaches, machines, utilisateurs, paramètres, sessions).

Grâce à cette architecture moderne, scalable et sécurisée, ANAS' GYM propose une solution complète adaptée aussi bien aux petites salles locales qu'aux grandes franchises multi-sites, avec une interface utilisateur intuitive et ergonomique. Elle vise ainsi à améliorer significativement l'organisation opérationnelle des salles, la satisfaction et la fidélisation des adhérents, et globalement la qualité de l'expérience sportive offerte.

1.4 Méthodologie de travail

Le choix d'une méthodologie de développement adéquate et adaptée au contexte est absolument essentiel afin de garantir la qualité du livrable final et de respecter scrupuleusement les délais impartis par l'entreprise. Dans le cadre de ce stage d'été de durée limitée, nous avons adopté la méthodologie agile **Scrum**, reconnue internationalement pour son efficacité dans les projets de développement logiciel.

Cette méthode se prête particulièrement bien à notre projet puisqu'elle favorise une livraison incrémentale et régulière de fonctionnalités testées et validées, une grande flexibilité face aux changements de priorités ou de spécifications, et une interaction continue et constructive entre l'équipe technique de développement et l'encadrement métier.

1.4.1 Rôles définis dans le projet

Scrum définit trois rôles principaux clairement identifiés, chacun avec des responsabilités spécifiques :

- **Product Owner** : Responsable de la vision produit, de la définition des fonctionnalités et de la priorisation du backlog selon la valeur métier. Ce rôle stratégique a été assuré par notre encadrante, Mme Syrine Khmirie, qui a défini les priorités et validé les incréments livrés à chaque sprint.
- **Équipe de développement** : Chargée de transformer les besoins exprimés en fonctionnalités utilisables, testées et documentées. Dans notre projet, ce rôle a été assuré

intégralement par moi-même, l'étudiante stagiaire, responsable du développement backend (Laravel), frontend mobile (React Native) et frontend web (React).

- **Scrum Master** : Garant de la bonne application de la méthodologie Scrum, il facilite le travail de l'équipe, supprime les obstacles organisationnels ou techniques, et anime les cérémonies Scrum. Ce rôle a été assuré par M. Helmi Ben Mahfoudh, qui a assuré le suivi quotidien et résolu les blocages rencontrés.

1.4.2 Processus Scrum adopté

Le processus Scrum que nous avons suivi est illustré de manière synthétique dans la figure ci-dessous :



FIGURE 1.4 – *Processus SCRUM appliqué au projet*

Chaque sprint suivait un cycle complet : planification initiale avec sélection des user stories prioritaires, développement quotidien avec daily standup informel, revue de sprint avec démonstration des fonctionnalités au Product Owner, et rétrospective pour identifier les axes d'amélioration continue.

1.4.3 Pilotage du projet avec Scrum

Notre projet ANAS' GYM a été découpé stratégiquement en plusieurs sprints successifs, chacun permettant de livrer un incrément fonctionnel testé et potentiellement déployable de l'application. La durée standard d'un sprint était fixée à environ deux semaines, permettant un rythme soutenu mais soutenable de développement et de validation.

1.4.4 Planification détaillée des sprints

Le tableau ci-dessous présente la planification complète et détaillée des sprints de notre projet, avec les tâches principales et les périodes correspondantes :

TABLE 1.1 – *Planification détaillée des sprints du projet ANAS' GYM*

Sprint	Tâches principales	Période
Sprint 0	Installation et configuration complète de l'environnement de travail (Laravel 10, React Native avec Expo, React pour dashboard). Mise en place du dépôt Git avec structure de branches (develop, feature, hotfix). Configuration des outils de développement (VS Code, extensions, linters).	5–16 Février
Sprint 1	Conception détaillée de la base de données relationnelle : modélisation des entités (utilisateurs avec rôles, branches, machines, charges, catégories, mouvements, exercices, workouts, programmes, coachs, sessions collectives). Création des migrations Laravel pour toutes les tables. Développement des seeders et factories pour générer des données de test réalistes.	17 Février–1 Mars
Sprint 2	Implémentation complète des API REST Laravel avec Sanctum : authentification sécurisée (register, login, logout, profil), gestion CRUD des utilisateurs avec rôles, gestion des branches et disponibilités, gestion des machines avec relations (charges, catégories), gestion des mouvements et exercices, gestion des workouts et programmes, gestion des coachs avec spécialités et disponibilités. Tests exhaustifs avec Thunder Client / Postman.	2–15 Mars
Suite à la page suivante		

Table 1.1 – suite de la page précédente

Sprint	Tâches principales	Période
Sprint 3	Développement du frontend mobile React Native : écran d'authentification (AuthScreen) avec login/-register, système de sauvegarde sécurisée du token Sanctum dans AsyncStorage, écran de liste des workouts avec affichage chronologique, écran de création de workout avec sélection de machines/charges/mouvements, navigation entre écrans avec React Navigation.	16–31 Mars
Sprint 4	Ajout des écrans détaillés dans l'application mobile : écran BranchDetailScreen affichant informations d'une salle (horaires, coordonnées, carte), liste des coachs disponibles dans la salle avec leurs spécialités, liste des sessions collectives avec filtres (femmes, enfants, gratuit), système de réservation de sessions avec gestion de capacité. Design et cartes interactives avec React Native Maps.	1–15 Avril
Sprint 5	Développement du tableau de bord administrateur avec React : interface de gestion des utilisateurs (ajout, modification, suppression, changement de rôle), gestion des réservations avec validation/annulation, consultation de la progression des membres, gestion des branches et paramètres publics, gestion des machines, charges, catégories, gestion des coachs et de leurs disponibilités, gestion des sessions collectives. Connexion complète avec l'API Laravel.	16–30 Avril
Suite à la page suivante		

Table 1.1 – suite de la page précédente

Sprint	Tâches principales	Période
Sprint 6	Phase de tests, corrections de bugs et intégration finale : tests fonctionnels de bout en bout (end-to-end), tests de sécurité (authentification, autorisations), tests de performance (charge API, temps de réponse), corrections des anomalies identifiées, optimisations (requêtes BD, bundle mobile). Rédaction de la documentation technique (API, architecture) et documentation fonctionnelle (guide utilisateur). Préparation de la présentation finale.	1–15 Mai

Cette planification rigoureuse et progressive a permis d’avancer de manière structurée, en validant à chaque sprint les fonctionnalités développées avant de passer aux suivantes, garantissant ainsi la qualité et la cohérence du système dans son ensemble.

1.5 Conclusion

Dans ce chapitre, nous avons présenté le cadre complet de notre projet : l’organisme d’accueil ExaDev avec son organisation et ses activités, le contexte et la problématique des salles de sport nécessitant une solution numérique moderne, l’analyse des solutions existantes sur le marché et leurs limitations, ainsi que la solution complète ANAS’ GYM que nous proposons avec toutes ses fonctionnalités innovantes.

Nous avons également exposé la méthodologie agile Scrum adoptée pour le pilotage du projet, avec la définition des rôles, le processus suivi et la planification détaillée des différents sprints. Cette approche itérative et incrémentale nous a permis d’avancer par étapes successives et validées, en garantissant une bonne communication avec notre encadrement et une livraison régulière de fonctionnalités testées.

Le prochain chapitre sera consacré à l’analyse approfondie et à la spécification détaillée des besoins fonctionnels et non fonctionnels de notre application, ainsi qu’à la modélisation UML des cas d’utilisation principaux du système.

SPÉCIFICATION DES BESOINS ET ANALYSE

Plan

2.1	Introduction	25
2.2	Identification des acteurs	25
2.3	Besoins fonctionnels	25
2.4	Besoins non fonctionnels	28
2.5	Langage de modélisation	28
2.6	Diagramme de cas d'utilisation	29
2.7	Exemples de cas d'utilisation	30
2.7.1	Créer un workout	30
2.7.2	Gérer les utilisateurs	30
2.7.3	Suivre la progression	31
2.8	Conclusion	31

2.1 Introduction

La réussite de tout projet logiciel repose sur une compréhension claire et structurée des besoins des utilisateurs finaux. Dans ce chapitre, nous présentons la phase de **spécification des besoins et d'analyse** de notre plateforme ANAS' GYM, qui repose sur une API RESTful Laravel sécurisée par Sanctum, consommée par une application mobile (React Native) et un tableau de bord administrateur (React).

Cette plateforme a pour objectif de centraliser la gestion des salles de sport multi-sites : workouts, programmes, machines, charges, mouvements, exercices, progression des utilisateurs, branches, séances collectives et paramètres publics.

Nous allons identifier les acteurs du système, détailler leurs besoins fonctionnels, présenter les contraintes non fonctionnelles et illustrer ces interactions à travers des cas d'utilisation.

2.2 Identification des acteurs

Dans l'état actuel du projet, deux acteurs interagissent directement avec le système :

- **Client sportif (membre)** Utilise l'application mobile. Il peut consulter ses workouts, créer ses entraînements, suivre sa progression corporelle et accéder aux informations des coaches et des séances collectives.
- **Administrateur** Utilise le dashboard web. Il gère toutes les ressources du système : branches, machines, charges, catégories, mouvements, exercices, programmes, séances collectives, utilisateurs et paramètres publics.

Important : Les **coachs ne sont pas des acteurs du système**. Ils ne disposent pas d'espace dédié.

Ils sont uniquement :

- **gérés par l'administrateur** (création, modification, spécialités, disponibilité) ;
- **consultés par les utilisateurs** dans l'application mobile (profil, spécialités, séances associées).

2.3 Besoins fonctionnels

Besoins fonctionnels du client sportif

2.3.0.0.1 Authentification et profil

- Création de compte, connexion et déconnexion via API Laravel Sanctum.
- Mise à jour des informations personnelles.

2.3.0.0.2 Gestion des workouts

- Consulter son historique d'entraînements (triés par date).
- Créer un workout et y ajouter plusieurs exercices.
- Pour chaque exercice : sélection d'une machine, charge, mouvement et définition des séries/répétitions.
- Modifier ou supprimer un workout.
- Marquer un workout comme accompli.

2.3.0.0.3 Programmes d'entraînement

- Consulter les programmes disponibles.
- Visualiser la structure d'un programme (workouts par jour).
- Suivre l'état d'avancement d'un programme.

2.3.0.0.4 Suivi de la progression (UserProgress)

- Visualiser l'évolution de ses performances (charges, fréquences, historique).

2.3.0.0.5 Branches, équipements et coaches

- Consulter la liste des branches et leurs informations.
- Consulter les machines par branch, ainsi que leurs charges et catégories.
- Accéder aux **profils des coaches** : photo, nom, bio, spécialités, séances associées.

2.3.0.0.6 Séances collectives (GroupTrainingSession)

- Consulter les séances collectives planifiées.
- Voir les détails : titre, coach associé, date, durée, capacité.
- S'inscrire à une séance, si la capacité le permet.
- Annuler son inscription.

Besoins fonctionnels de l'administrateur

2.3.0.0.7 Gestion des branches

- Créer, modifier, supprimer une branch.
- Gérer les disponibilités et horaires des branches.

2.3.0.0.8 Gestion des équipements

- Gestion complète des machines.
- Gestion des charges associées aux machines.
- Gestion des catégories.
- Gestion des mouvements.

2.3.0.0.9 Gestion des exercices, workouts et programmes

- Créer, modifier ou supprimer des exercices.
- Créer des workouts types.
- Créer des programmes d'entraînement complets.

2.3.0.0.10 Gestion des utilisateurs

- Ajouter, modifier, supprimer des utilisateurs.
- Affecter un rôle (membre, admin).

2.3.0.0.11 Gestion des coaches

- Créer/modifier les profils des coaches.
- Gérer leurs spécialités.
- Assigner un coach à une branch.
- (Optionnel) Gérer leurs disponibilités.

2.3.0.0.12 Gestion des séances collectives

- Planifier des séances collectives.
- Définir la capacité, la branch et le coach associé.
- Consulter les inscrits.

2.3.0.0.13 Paramètres publics

- Modifier les paramètres visibles par tous (nom de l'application, messages, réseaux sociaux, horaires, etc.).

2.4 Besoins non fonctionnels

- **Performance** : temps de réponse API raisonnable, pagination des grandes listes.
- **Sécurité** : authentification Sanctum, hachage des mots de passe, validation serveur.
- **Cohérence** : capacité des séances respectée, transactions sécurisées.
- **Ergonomie** : interfaces intuitives sur mobile et web.
- **Évolutivité** : architecture REST prêt à étendre (nutrition, statistiques avancées...).
- **Portabilité** : compatibilité mobile Android/iOS et navigateurs modernes.

2.5 Langage de modélisation

La modélisation du système a été réalisée avec **UML** via l'outil **StarUML**, permettant de représenter clairement :

- les cas d'utilisation,
- le diagramme de classes (extrait du modèle backend),
- la structure générale des modules métiers.

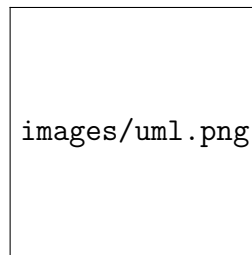
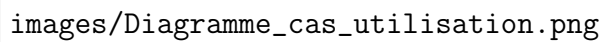


FIGURE 2.1 – *StarUML*

2.6 Diagramme de cas d'utilisation



images/Diagramme_cas_utilisation.png

FIGURE 2.2 – *Diagramme de cas d'utilisation de la plateforme*

2.7 Exemples de cas d'utilisation

2.7.1 Créer un workout

TABLE 2.1 – *Cas d'utilisation — Créer un workout*

Acteur	Client sportif
Précondition	L'utilisateur est authentifié. Machines, charges et mouvements existent.
Scénario nominal	<ol style="list-style-type: none">1. Le client ouvre l'onglet "Workouts".2. Il choisit "Créer un workout".3. Il saisit un titre et des notes.4. Il sélectionne une branch, puis une ou plusieurs machines.5. Pour chaque machine, il choisit une charge et un mouvement.6. Il renseigne séries et répétitions.7. Il enregistre son workout.
Postcondition	Le workout apparaît dans l'historique du client.

2.7.2 Gérer les utilisateurs

TABLE 2.2 – *Cas d'utilisation — Gérer les utilisateurs*

Acteur	Administrateur
Scénario nominal	<ol style="list-style-type: none">1. L'administrateur ouvre la section "Utilisateurs".2. Il consulte la liste complète.3. Il peut ajouter, modifier ou supprimer un utilisateur.

2.7.3 Suivre la progression

TABLE 2.3 – *Cas d'utilisation — Suivre la progression*

Acteur	Client sportif
Scénario nominal	<ol style="list-style-type: none">1. Le client accède à “Progression”.2. Le système affiche ses workouts ou métriques UserProgress.3. Le client visualise l'évolution de ses performances.

2.8 Conclusion

Ce chapitre a défini les acteurs réellement pris en charge (client sportif et administrateur), leurs besoins fonctionnels et non fonctionnels, ainsi que les principaux cas d'utilisation. Cette analyse constitue la base du chapitre suivant, dédié à la conception technique et conceptuelle du système.

Plan

3.1	Introduction	32
3.2	Outils et Frameworks	32
3.3	Architecture du projet	33
3.4	Diagramme de classes	34
3.5	Diagrammes de Use Case	35
3.6	Conclusion	36

3.1 Introduction

Dans la continuité de l'analyse des besoins réalisée au chapitre précédent, ce chapitre présente l'étude conceptuelle de l'application mobile et web développée durant mon stage chez ExaDev. L'objectif est de décrire la structure technique du système, les outils utilisés ainsi que la modélisation conceptuelle permettant de garantir une architecture cohérente, scalable et conforme aux besoins fonctionnels.

L'application repose sur une API REST développée avec **Laravel** (authentification **Sanctum**), un frontend mobile réalisé avec **React Native** via **Expo Go**, et un tableau de bord administrateur utilisant **React**. L'ensemble communique avec une base de données **MySQL** et a été testé via **ThunderClient**. Cette architecture assure une centralisation fiable des données relatives aux membres, branches, équipements, workouts, programmes, coaches et sessions collectives.

3.2 Outils et Frameworks

Afin de développer une solution moderne, robuste et maintenable, plusieurs outils ont été utilisés :

- **Laravel** : Framework backend utilisé pour concevoir l'API RESTful, gérer l'authentification (Sanctum), les modèles métiers, les contrôleurs et la communication avec MySQL.
- **React Native (Expo)** : Utilisé pour développer l'application mobile multi-plateforme, tester en temps réel sur smartphone via Expo Go.
- **React** : Employé pour créer le tableau de bord administrateur (gestion des utilisateurs, ressources et données).
- **MySQL** : Système de gestion de base de données utilisé pour stocker toutes les entités applicatives.
- **ThunderClient** : Outil pour tester les endpoints de l'API Laravel.
- **VS Code** : Environnement de développement utilisé pour l'ensemble du projet.
- **Git / GitHub** : Gestion de versions et collaboration.

3.3 Architecture du projet

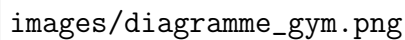
L'architecture générale suit une organisation **API-first**, où toutes les fonctionnalités de l'application mobile et du dashboard reposent sur les endpoints Laravel :

- **Backend Laravel** : Gère toute la logique métier (branches, machines, charges, catégories, programmes, workouts, mouvements, coachs, profils publics, sessions collectives, réservations, paramètres globaux).
- **Application mobile React Native** : Accès utilisateur final (authentification, historique des workouts, création d'entraînements, consultation des machines, coachs et sessions collectives, progression).
- **Tableau de bord admin React** : Gestion centralisée de toutes les ressources (CRUD complet sur utilisateurs, branches, machines, charges, mouvements, programmes, sessions collectives, paramètres publics).
- **Base MySQL** : Stockage structuré des entités et relations (workouts, exercices, machines, charges, coachs, programmes...).

L'interaction entre ces composants est illustrée dans l'architecture globale du PDF fourni.

3.4 Diagramme de classes

Le modèle conceptuel de données regroupe toutes les entités majeures du système et leurs relations. Le diagramme ci-dessous reprend strictement les données issues de l'API Laravel et des notes techniques du projet.



images/diagramme_gym.png

FIGURE 3.1 – *Diagramme de classes de l'application MyGym*

Ce modèle inclut notamment :

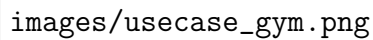
— **User** : membres, administrateurs,

- **Branch** et ses disponibilités.
- **Machines**, **charges** et **catégories**.
- **Workouts**, **exercices** et mouvements.
- **Programmes** et assignments.
- **Coachs**, spécialités et disponibilités.
- **Sessions collectives** et réservations.
- **UserProgress** : évolution corporelle.

3.5 Diagrammes de Use Case

Les use cases modélisent les interactions principales entre les acteurs identifiés :

- Client sportif : consulter ses workouts, créer un workout, suivre sa progression, consulter les coachs, réserver des sessions collectives.
- Administrateur : gérer utilisateurs, machines, charges, mouvements, catégories, sessions collectives, paramètres publics.



images/usecase_gym.png

FIGURE 3.2 – *Diagramme général de Use Case de la plateforme*

3.6 Conclusion

Ce chapitre a présenté l'étude conceptuelle de la solution MyGym : les outils utilisés, l'architecture globale de la plateforme, ainsi que les modèles UML permettant de structurer l'ensemble. Ce socle conceptuel garantit la cohérence du développement qui sera détaillé dans le chapitre suivant.

RÉALISATION GLOBALE DE LA PLATE-FORME

Plan

4.1	Introduction	38
4.2	Environnement de développement	38
4.2.1	Outils utilisés	38
4.2.2	Structure globale du dépôt	38
4.3	Réalisation du backend (Laravel + Sanctum + MySQL)	39
4.3.1	Conception et création de la base de données	39
4.3.2	Seeders et données initiales	39
4.3.3	Authentification avec Laravel Sanctum	40
4.3.4	Implémentation des endpoints REST	40
4.3.5	Tests via Thunder Client	41
4.4	Réalisation de l'application mobile (React Native + Expo)	41
4.4.1	Structure et navigation	41
4.4.2	Connexion à l'API	41
4.4.3	Fonctionnalités principales réalisées	42
4.5	Réalisation du dashboard administrateur (React)	42
4.5.1	Structure générale	43
4.5.2	Authentification administrateur	43
4.5.3	Fonctionnalités réalisées	43
4.6	Conclusion	44
	Conclusion Générale	44

4.1 Introduction

Après avoir défini les besoins fonctionnels (Chapitre 2) et établi l'étude conceptuelle et l'architecture générale (Chapitre 3), ce chapitre présente la réalisation effective de la plateforme ANAS' GYM.

Cette réalisation s'appuie sur une architecture en trois blocs :

- une **API RESTful backend** développée avec **Laravel** et sécurisée par **Sanctum**, connectée à une base **MySQL** ;
- une **application mobile** développée avec **React Native** via **Expo**, destinée aux membres ;
- un **dashboard web administrateur** développé avec **React**, destiné uniquement à la gestion.

L'objectif du chapitre est de détailler la mise en œuvre technique de ces trois volets, depuis l'environnement de travail jusqu'aux principaux modules réalisés.

4.2 Environnement de développement

Le développement a été réalisé dans un environnement stable et standardisé, permettant une productivité optimale et une compatibilité avec les outils modernes.

4.2.1 Outils utilisés

- **VS Code** : éditeur principal utilisé pour le backend, le mobile et le dashboard.
- **Laragon** : serveur local facilitant l'exécution de Laravel et MySQL en environnement Windows.
- **Git / GitHub** : gestion de versions, commits, organisation du code et sauvegarde du projet.
- **Thunder Client** : tests manuels des endpoints API (CRUD, authentification, réponses JSON).
- **Expo Go** : tests de l'application mobile sur téléphone réel.

4.2.2 Structure globale du dépôt

Le projet a été organisé en trois parties séparées, chacune avec un rôle clair :

- `/backend` : API Laravel, modèles, migrations, contrôleurs.
- `/mobile` : application React Native et navigation.
- `/dashboard` : interface React pour l'administration.

Cette séparation assure une meilleure lisibilité du code et simplifie la maintenance.

4.3 Réalisation du backend (Laravel + Sanctum + MySQL)

Le backend constitue le noyau central de la solution. Il expose l'ensemble des fonctionnalités nécessaires via une API REST sécurisée.

4.3.1 Conception et création de la base de données

Sur la base du diagramme de classes validé, les principales entités ont été créées dans MySQL via **migrations Laravel**. Les tables mises en place correspondent aux modules métiers du projet :

- **users** : comptes des membres et administrateurs avec rôles.
- **branches** et **branch_availabilities**.
- **machines**, **charges**, **categories**, **movements**.
- **exercises**.
- **workouts** et **workout_exercises**.
- **programmes** et **programme_workouts**.
- **user_progress**.
- **coaches** et **coach_availabilities**.
- **group_training_sessions** et **reservations**.

Chaque migration a été versionnée afin de garantir une création cohérente du schéma sur n'importe quel environnement.

4.3.2 Seeders et données initiales

Des **seeders** Laravel ont été développés afin de générer des données réalistes permettant :

- de tester les relations entre entités ;

- d'alimenter l'interface mobile et le dashboard ;
- de simuler un contexte réel multi-branches.

4.3.3 Authentification avec Laravel Sanctum

L'authentification repose sur **Sanctum**, adapté aux applications mobiles et SPA. Les endpoints suivants ont été implémentés :

- **POST /register** : inscription membre.
- **POST /login** : connexion et génération du token.
- **POST /logout** : révocation du token.
- **GET /profile** : récupération du profil authentifié.

Chaque route protégée nécessite un token valide, assurant la sécurité des données utilisateurs.

4.3.4 Implémentation des endpoints REST

La logique métier a été structurée via **Controllers Laravel**. Les modules API réalisés couvrent :

- **Branches** : CRUD complet, disponibilités, informations publiques.
- **Machines et charges** : gestion des machines, association aux branches, gestion des charges et catégories.
- **Mouvements et exercices** : création et consultation des mouvements, utilisation dans les workouts.
- **Workouts** : création, modification, suppression, listing et historique par utilisateur.
- **Programmes** : création/consultation des programmes et affectation des workouts.
- **Progression** : endpoints pour ajouter et consulter les mesures `user_progress`.
- **Coachs** : CRUD profils coach, consultation publique côté mobile.
- **Sessions collectives** : planning, capacité, détails, réservations.

Toutes les réponses sont retournées au format **JSON** pour être consommées par les frontends.

4.3.5 Tests via Thunder Client

Chaque endpoint a été validé manuellement via **Thunder Client** :

- vérification des réponses HTTP (200, 401, 404, 422) ;
- validation des payloads JSON ;
- tests des routes protégées par token.

Ces tests ont permis de corriger les erreurs avant l'intégration mobile/web.

4.4 Réalisation de l'application mobile (React Native + Expo)

L'application mobile constitue l'interface principale des membres. Elle a été développée avec React Native, en s'appuyant sur Expo pour accélérer les tests et le déploiement local.

4.4.1 Structure et navigation

La navigation a été organisée sous forme d'écrans dédiés, accessibles via une structure cohérente :

- Authentification (register / login).
- Accueil / Workouts.
- Création d'un workout.
- Détails d'une branch.
- Consultation coaches.
- Sessions collectives.
- Progression.

4.4.2 Connexion à l'API

L'application consomme directement l'API Laravel via des requêtes HTTP sécurisées. Le token Sanctum est :

- récupéré à la connexion ;
- stocké localement ;
- attaché automatiquement aux requêtes sécurisées.

4.4.3 Fonctionnalités principales réalisées

4.4.3.0.1 Gestion des workouts

- affichage chronologique des workouts ;
- création d'un workout avec ajout d'exercices (machine, charge, mouvement, séries/répétitions) ;
- suppression/modification.

4.4.3.0.2 Branches et équipements

- listing des branches disponibles ;
- détails d'une branch (infos publiques, équipements associés) ;
- consultation machines + charges.

4.4.3.0.3 Consultation des coaches Les coaches sont affichés sous forme de profils consultables :

- informations principales ;
- spécialités ;
- sessions collectives animées.

Remarque : il n'existe pas d'espace coach dans cette version.

4.4.3.0.4 Sessions collectives

- affichage planning global ;
- filtre par branch/type ;
- réservation et annulation selon capacité.

4.4.3.0.5 Progression

- consultation de l'évolution de ses mesures corporelles via l'API ;
- visualisation historique.

4.5 Réalisation du dashboard administrateur (React)

Le dashboard constitue l'interface de gestion globale de la plateforme, accessible uniquement aux administrateurs.

4.5.1 Structure générale

L'application React a été structurée en modules CRUD correspondant aux ressources backend :

- utilisateurs ;
- branches ;
- machines / charges / catégories ;
- mouvements / exercices ;
- programmes / workouts types ;
- coaches ;
- sessions collectives ;
- paramètres publics.

4.5.2 Authentification administrateur

L'accès au dashboard est soumis à authentification via token Sanctum. Après connexion, l'administrateur dispose d'une interface sécurisée.

4.5.3 Fonctionnalités réalisées

4.5.3.0.1 Gestion des ressources

- ajout, modification, suppression de chaque entité ;
- visualisation sous forme de tableaux ;
- formulaires de création et édition.

4.5.3.0.2 Gestion des coaches

- création et modification de profils coach ;
- spécialités et affectation à une branch.

4.5.3.0.3 Sessions collectives

- création d'une session avec branch, coach, date et capacité ;
- suivi des réservations.

L'interface React communique avec le backend pour garantir une gestion centralisée.

4.6 Conclusion

Ce chapitre a présenté la réalisation globale de la plateforme ANAS' GYM. Nous avons détaillé la mise en œuvre du backend Laravel sécurisé par Sanctum, le développement de l'application mobile React Native sous Expo, ainsi que la réalisation du dashboard administrateur en React.

Le chapitre suivant présentera les tests finaux, l'intégration complète et la mise en production de la solution.

Conclusion et perspectives

Ce stage d'été de 2^e année, effectué au sein de l'entreprise **ExaDev** à Sfax, a constitué une expérience professionnelle et technique très enrichissante. L'objectif principal qui nous a été confié était de concevoir et développer une plateforme complète destinée à moderniser la gestion des salles de sport multi-sites et à améliorer l'expérience des membres. Ce projet, nommé **ANAS' GYM (MyGym)**, repose sur une architecture moderne combinant une API RESTful **Laravel** sécurisée par **Sanctum**, une application mobile **React Native** via **Expo**, et un dashboard web administrateur développé avec **React**.

Tout au long de ce projet, nous avons pu couvrir l'ensemble du cycle de développement logiciel : étude préalable, analyse et spécification des besoins, conception UML, implémentation technique et intégration finale. La solution développée permet aujourd'hui aux membres de consulter leur historique d'entraînements, créer des workouts détaillés, suivre leur progression corporelle, consulter les profils des coachs disponibles et réserver des cours collectifs. Par ailleurs, le dashboard administrateur offre une gestion centralisée des branches, équipements, sessions collectives, utilisateurs, programmes et coachs. L'ensemble de ces fonctionnalités répond à la problématique initiale de manque de centralisation, d'absence d'historisation des entraînements et de difficulté de gestion multi-sites.

La réalisation du projet a nécessité la maîtrise et l'intégration de plusieurs technologies complémentaires. Nous avons consolidé nos compétences en **développement backend** (Laravel, migrations, relations Eloquent, API sécurisée), en **développement mobile multiplateforme** (React Native, Expo Go, navigation, consommation API) ainsi qu'en **développement web moderne** (React, dashboard administratif). Ce travail nous a également permis d'apprendre à structurer un projet complet, à garantir la cohérence des données et à assurer la sécurité des échanges entre les applications clientes et le serveur.

Bien que la plateforme réponde pleinement aux besoins actuellement définis, plusieurs perspectives d'amélioration peuvent être envisagées afin de renforcer son impact et d'élargir ses fonctionnalités. Parmi les axes d'évolution possibles, nous proposons :

- **Ajout d'un espace coach dédié** : dans la version actuelle, les coachs ne disposent pas encore d'un espace de connexion. Une extension future consisterait à intégrer un module coach complet permettant la création de programmes personnalisés, le suivi direct des membres, et la gestion autonome de ses séances collectives.
- **Amélioration du suivi sportif** : enrichir le module de progression en ajoutant davantage d'indicateurs (masse musculaire, masse grasse, statistiques avancées) et des

visualisations plus détaillées afin d'offrir une analyse encore plus précise de l'évolution du sportif.

- **Système de recommandations intelligentes** : intégrer un moteur de recommandations basé sur l'historique des workouts et les objectifs de l'utilisateur, afin de suggérer automatiquement des exercices, charges ou programmes adaptés.
- **Notifications évoluées** : améliorer le système de notifications push pour inclure davantage de scénarios (rappels de progression, alertes de changement de planning, encouragements automatiques), avec une personnalisation par utilisateur.
- **Déploiement sur stores** : préparer une version stable destinée à être publiée sur Google Play et App Store, avec prise en charge complète des builds Expo et configuration des environnements de production.

En conclusion, ce stage a constitué une opportunité concrète de travailler sur un produit réel, destiné à un besoin pratique et actuel. Il nous a permis de mettre en application nos acquis académiques, de développer une solution utile et fonctionnelle, et d'approfondir nos compétences dans des technologies modernes largement utilisées dans le monde professionnel. Nous remercions l'équipe ExaDev pour son encadrement, sa disponibilité et la confiance accordée tout au long de ce projet. Cette expérience renforce notre motivation à poursuivre dans le domaine du développement logiciel et nous prépare efficacement aux exigences des projets professionnels à venir.

Bibliographie

- [1] Laravel Documentation, *The PHP Framework for Web Artisans*. Disponible sur : <https://laravel.com/docs> (consulté en décembre 2025). :contentReference[oaicite :0]index=0
- [2] Laravel Sanctum Documentation, *API Token Authentication for Laravel*. Disponible sur : <https://laravel.com/docs/sanctum> (consulté en décembre 2025). :contentReference[oaicite :1]index=1
- [3] React Documentation, *Library for building user interfaces*. Disponible sur : <https://react.dev> (consulté en décembre 2025). :contentReference[oaicite :2]index=2
- [4] React Native Documentation, *Framework for building native apps using React*. Disponible sur : <https://reactnative.dev/docs> (consulté en décembre 2025). :contentReference[oaicite :3]index=3
- [5] Expo Documentation, *Framework et outils pour React Native*. Disponible sur : <https://docs.expo.dev> (consulté en décembre 2025). :contentReference[oaicite :4]index=4
- [6] MySQL Documentation, *MySQL 8.0 Reference Manual*. Disponible sur : <https://dev.mysql.com/doc/> (consulté en décembre 2025).
- [7] Git Documentation, *Distributed Version Control System*. Disponible sur : <https://git-scm.com/doc> (consulté en décembre 2025).
- [8] GitHub Docs, *Hosting and collaboration platform*. Disponible sur : <https://docs.github.com> (consulté en décembre 2025).
- [9] Visual Studio Code Documentation, *Code editor for modern development*. Disponible sur : <https://code.visualstudio.com/docs> (consulté en décembre 2025).
- [10] Laragon Documentation, *Local development environment for PHP and Node.js*. Disponible sur : <https://laragon.org/docs> (consulté en décembre 2025). :contentReference[oaicite :5]index=5

Webographie