

Application de Suivi de Santé

Réalisé par :

BARAKATE Imane
DAGOUN Oumaima
MEKIANI Leila
NOR ELYAKINE Aya
SLIOUI Badr Eddine

Encadré par :

Mme. SBAI Hanae

Marrakech, Maroc
2025

Remerciements

Avant de présenter ce rapport, nous souhaitons exprimer nos sincères remerciements à notre encadrante **Mme SBAI Hanae**, pour son inestimable soutien tout au long de notre semestre.

Sa guidance éclairée, sa patience et son expertise ont été essentielles pour nous permettre d'accomplir nos objectifs avec succès.

Elle a su nous guider avec bienveillance, nous encourageant à repousser nos limites et à explorer de nouvelles idées. Ses conseils précieux et ses retours constructifs nous ont permis de prendre confiance en nos compétences et de nous améliorer tout au long du processus.

Grâce à elle, nous avons acquis des compétences essentielles pour notre future carrière, et nous sommes reconnaissants d'avoir eu la chance de travailler à ses côtés.

Résumé

Ce document est structuré en plusieurs chapitres pour présenter de manière détaillée le projet de développement d'une application web de suivi de santé.

Le premier chapitre expose le contexte du projet, met en lumière l'importance de fournir des outils modernes et personnalisés pour le suivi de la santé, et présente la problématique à laquelle le projet cherche à répondre.

Le deuxième chapitre précise les objectifs du projet, notamment la création d'une plate-forme interactive permettant aux utilisateurs de suivre leur état de santé (poids, activités physiques, nutrition) tout en offrant des analyses personnalisées.

Le troisième chapitre détaille la méthodologie adoptée, mettant en avant l'utilisation du framework **Spring Boot** pour le back-end et de **Angular** pour le front-end, ainsi que l'adoption de la méthodologie **SCRUM** pour une gestion efficace du projet.

Enfin, le quatrième chapitre présente les différentes technologies utilisées dans le développement du système, en mettant l'accent sur leur rôle dans l'amélioration de la convivialité, de la performance et de l'efficacité globale de l'application.

Mots clés : suivi de santé, Spring Boot, Angular, SCRUM, UML, graphiques.

Abstract

This document is structured into several chapters to provide a detailed presentation of the project focused on developing a web-based health tracking application.

The first chapter introduces the project's context, highlighting the importance of providing modern and personalized tools for health tracking and presenting the problem the project aims to address.

The second chapter specifies the project objectives, particularly the creation of an interactive platform that enables users to monitor their health data (weight, physical activities, nutrition) while offering personalized analyses.

The third chapter elaborates on the methodology adopted, emphasizing the use of the **Spring Boot** framework for the back-end and **Angular** for the front-end, along with the implementation of the **SCRUM** methodology for efficient project management.

Finally, the fourth chapter presents the various technologies used in the development of the system, focusing on their role in enhancing the application's user-friendliness, performance, and overall effectiveness.

Keywords : health tracking, Spring Boot, Angular, SCRUM, UML, charts.

Table des matières

Remerciements	i
Résumé	ii
Abstract	iii
Liste des figures	vi
Liste des tableaux	vii
Chapitre 1 : Contexte Général	1
1.1 Introduction	2
1.2 Présentation de l'entreprise	2
1.3 Contexte du projet	2
1.4 Étude de l'existant	2
1.4.1 Analyse comparative des solutions existantes	2
1.5 Problématique générale	2
1.6 Problématique du projet	2
1.7 Solution adoptée	3
1.8 Objectifs	4
1.9 Division du projet en modules (Sprints)	4
1.9.1 Répartition des tâches	4
1.9.2 Planification des sprints	4
1.9.3 Outils utilisés pour la planification	4
1.9.4 Livrables	5
Chapitre 2 : Étude Préliminaire et Fonctionnelle	6
2.1 Introduction	7
2.2 Cadre du Projet	7
2.3 Objectifs Fonctionnels	7
2.4 Analyse des Fonctionnalités (par Sprint)	7
2.4.1 Sprint 1 : Mise en Place des Bases	7
2.4.2 Sprint 2 : Développement des Bases Fonctionnelles	7
2.4.3 Sprint 3 : Intégration et Fonctionnalités Avancées	7
2.4.4 Sprint 4 : Finalisation et Optimisation	8
2.5 Analyse des Règles de Gestion (par Sprint)	8
2.5.1 Sprint 1	8
2.5.2 Sprint 2	8
2.5.3 Sprint 3	8
2.5.4 Sprint 4	8

2.6 Besoins et Contraintes	9
2.6.1 Besoins Techniques	9
2.6.2 Besoins Non Techniques	9
2.6.3 Contraintes	9
2.6.4 Exigences	9
2.7 Conclusion	9
Chapitre 3 : Analyse et Conception	10
3.1 Introduction	11
3.2 Le Choix d'UML	11
3.3 Diagrammes Utilisés	11
3.3.1 Diagramme de Classe	11
3.3.2 Diagramme de Cas d'Utilisation	12
3.3.3 Diagramme de Séquence	14
3.3.4 Diagramme de Composant	15
3.4 Conclusion	16
Chapitre 4 : Technologies et Réalisation	17
4.1 Introduction	18
Introduction	18
4.2 Architecture applicative de l'application	18
4.3 Environnement de développement technique	18
4.4 Environnement de développement logiciel	19
4.5 Mise en oeuvre	19
4.5.1 Interface d'inscription	19
4.5.2 Interface de connexion	20
4.5.3 Tableau de bord - Accueil	20
4.5.4 Suivi de progression	21
4.5.5 Rapport de suivi de santé	21
4.5.6 Plan de repas personnel	22
4.5.7 Sélection des repas	23
4.5.8 Planning des repas	23
4.5.9 Choix d'une activité	24
4.5.10 Activité sélectionnée	24
4.5.11 Historique des activités	25
4.6 Conclusion	25
Bibliographie	27

Liste des figures

Figure 3.1 : Diagramme de classe : Modèle conceptuel du système	12
Figure 3.2 : Diagramme de cas d'utilisation	13
Figure 3.3 : Diagramme de séquence : Consultation des activités	14
Figure 3.4 : Diagramme de séquence : Consultation des repas	15
Figure 3.5 : Diagramme de composant : Architecture logicielle du système	16
Figure 4.1 : Architecture applicative.	18
Figure 4.2 : Interface d'inscription.	20
Figure 4.3 : Interface de connexion.	20
Figure 4.4 : Tableau de bord - Résumé.	21
Figure 4.5 : Suivi de progression hebdomadaire.	21
Figure 4.6 : Rapport de suivi de santé.	22
Figure 4.7 : Interface du plan de repas personnel.	23
Figure 4.8 : Interface de sélection des repas.	23
Figure 4.9 : Interface de planning des repas.	24
Figure 4.10 : Interface de choix d'une activité.	24
Figure 4.11 : Interface de l'activité sélectionnée.	25
Figure 4.12 : Interface de l'historique des activités.	25

Liste des tableaux

Table 1.1 : Analyse comparative des applications de suivi de santé	3
Table 1.2 : Planification des sprints et tâches principales.	5
Table 3.1 : Description du diagramme de classe	12
Table 3.2 : Description du diagramme de cas d'utilisation	13
Table 3.3 : Description du diagramme de séquence : Consultation des activités	14
Table 3.4 : Description du diagramme de séquence : Consultation des repas .	15
Table 3.5 : Description du diagramme de composant	16

Chapitre 1

Contexte Général

1.1 Introduction

L'objectif de ce projet est de développer une application web de suivi de santé qui permet aux utilisateurs de surveiller leur état physique, en combinant les technologies modernes.

1.2 Présentation de l'entreprise

L'école **EMSI Marrakech** est une institution d'enseignement spécialisée dans l'ingénierie informatique, offrant un cadre idéal pour la réalisation de projets innovants comme celui-ci.

1.3 Contexte du projet

Aujourd'hui, avec l'essor des technologies, il est essentiel de fournir des outils simples et efficaces permettant aux utilisateurs de surveiller leur santé physique, leurs activités et leur alimentation. Ce projet s'inscrit dans ce besoin en proposant une solution personnalisée.

1.4 Étude de l'existant

Plusieurs applications de suivi de santé existent sur le marché, comme *Google Fit*, *MyFitnessPal*, et autres. Toutefois, elles sont souvent limitées ou non adaptées aux besoins spécifiques des utilisateurs, comme l'intégration d'analyses précises et de conseils personnalisés.

1.4.1 Analyse comparative des solutions existantes

Le tableau ci-dessous présente une analyse comparative des principales applications disponibles sur le marché :

1.5 Problématique générale

Comment développer une solution moderne permettant aux utilisateurs de surveiller leur santé, tout en fournissant des analyses et des recommandations basées sur leurs données personnelles ?

1.6 Problématique du projet

Le projet vise à répondre aux questions suivantes :

- Comment centraliser les informations liées au poids, à l'activité physique et à la nutrition dans une seule application ?

Auteur	Date de création	Description	Avantages	Inconvénients	Lien de source
Google LLC	2014	Application gratuite pour le suivi de la condition physique et des activités quotidiennes.	Synchronisation avec de nombreux appareils et intégration avec Google Health.	Fonctionnalités limitées pour l'analyse nutritionnelle et les conseils personnalisés.	https://fit.google.com/
Under Armour	2005	MyFitnessPal propose un suivi alimentaire avancé et une grande base de données de repas.	Suivi précis de l'apport calorique et des macronutriments.	Version gratuite limitée, la version premium est payante.	https://www.myfitnesspal.com/
Nike Inc.	2010	Nike Training Club offre des séances d'entraînement guidées et un suivi des activités physiques.	Entraînements guidés et adaptés aux niveaux débutants et avancés.	Limité aux entraînements, pas de suivi nutritionnel.	https://www.nike.com/ntc-app
Samsung	2015	Samsung Health fournit un suivi global de la santé avec des fonctionnalités pour l'activité, le sommeil et la nutrition.	Interface conviviale et prise en charge de multiples aspects de la santé.	Certaines fonctionnalités avancées nécessitent des appareils Samsung.	https://health.apps.samsung.com/

TABLE 1.1 – Analyse comparative des applications de suivi de santé

- Comment fournir des graphiques d'analyse et des rapports adaptés aux besoins des utilisateurs ?
- Comment permettre une expérience fluide sur le web et mobile avec une synchronisation des données ?

1.7 Solution adoptée

Pour répondre à ces problématiques, une solution basée sur les technologies **Spring Boot** pour le backend, **Angular** pour le frontend.

1.8 Objectifs

Les objectifs principaux de ce projet sont :

- Mettre en place un système d’inscription et d’authentification sécurisé.
- Permettre aux utilisateurs d’enregistrer et suivre leur poids, leurs activités physiques et leur alimentation.
- Offrir des analyses visuelles (graphiques) pour chaque catégorie de suivi.
- Proposer des conseils de santé personnalisés.
- Générer des rapports PDF pour faciliter le partage avec des professionnels de la santé.

1.9 Division du projet en modules (Sprints)

Pour assurer une gestion optimale du projet, nous avons appliqué la méthodologie **Agile Scrum** en divisant le projet en plusieurs sprints. Chaque sprint correspond à une période définie avec des objectifs clairs et mesurables. L’équipe est composée de 5 membres, chacun ayant des responsabilités spécifiques.

1.9.1 Répartition des tâches

La répartition des tâches au sein de l’équipe est la suivante :

- **Deux développeurs backend** : Implémentation des API avec Spring Boot et configuration de la base de données.
- **Deux développeurs frontend/mobile** : Développement de l’interface utilisateur avec Angular et Flutter, intégration avec les API.
- **Testeur** : Validation des fonctionnalités développées, identification des bugs, et rédaction des rapports de tests.

1.9.2 Planification des sprints

Le projet a démarré le **10 novembre** et doit être finalisé avant le **15 décembre**, soit une durée de 5 semaines. Nous avons organisé le travail en 5 sprints hebdomadaires comme le montre le tableau 1.2

1.9.3 Outils utilisés pour la planification

Pour suivre l’avancement des sprints et des tâches, nous avons utilisé les outils suivants :

- **Jira** : Pour la gestion des tâches et le suivi des sprints.
- **GitHub** : Pour le versionnement du code et la collaboration entre les membres de l’équipe.

Sprint	Tâches principales
Sprint 1 (10-16 novembre)	<ul style="list-style-type: none"> - Mise en place de l'environnement de développement. - Conception de la base de données. - Création des entités principales. - Développement des endpoints de base pour l'authentification et l'inscription.
Sprint 2 (17-23 novembre)	<ul style="list-style-type: none"> - Développement des fonctionnalités backend pour l'enregistrement des données. - Création des services d'API pour le frontend. - Développement des premières interfaces utilisateur avec Angular.
Sprint 3 (24-30 novembre)	<ul style="list-style-type: none"> - Intégration complète entre le backend et les interfaces frontend. - Mise en place des graphiques d'analyse. - Développement de la fonctionnalité de génération de rapports. - Tests unitaires des services backend.
Sprint 4 (1-7 décembre)	<ul style="list-style-type: none"> - Finalisation des fonctionnalités principales. - Intégration des notifications pour les rappels. - Tests et correction des bugs. - Préparation de la documentation utilisateur.
Sprint 5 (8-15 décembre)	<ul style="list-style-type: none"> - Tests finaux et optimisation des performances. - Validation complète du système. - Préparation de la présentation finale du projet. - Livraison du projet.

TABLE 1.2 – Planification des sprints et tâches principales.

1.9.4 Livrables

À la fin de chaque sprint, un livrable est prévu :

- **Sprint 1** : Base de données et endpoints d'authentification et d'inscription.
- **Sprint 2** : API fonctionnelles pour l'enregistrement des données et interfaces de base.
- **Sprint 3** : Intégration complète et fonctionnalités avancées (graphiques, rapports).
- **Sprint 4** : Version finale de l'application prête pour la démonstration.

Chapitre 2

Étude Préliminaire et Fonctionnelle

2.1 Introduction

Ce chapitre est dédié à l'étude préliminaire et fonctionnelle du projet. Il vise à analyser les bases du projet, identifier les fonctionnalités principales, les règles de gestion, ainsi que les besoins et contraintes techniques et non techniques.

2.2 Cadre du Projet

Le projet s'inscrit dans une dynamique de transformation numérique en proposant une solution personnalisée et innovante pour le suivi de santé. L'objectif principal est de fournir une plateforme interactive et intuitive permettant aux utilisateurs de surveiller leurs données physiques, d'analyser leur évolution et de recevoir des recommandations adaptées.

2.3 Objectifs Fonctionnels

Les principaux objectifs fonctionnels de l'application sont les suivants :

- Offrir un espace personnel pour chaque utilisateur où il peut suivre ses données physiques (poids, activité physique, alimentation).
- Générer des graphiques d'analyse pour visualiser les tendances et évolutions.
- Proposer des rapports personnalisés sous format PDF.
- Intégrer des rappels automatiques pour maintenir la motivation des utilisateurs.
- Garantir une expérience utilisateur fluide sur les plateformes web et mobile.

2.4 Analyse des Fonctionnalités (par Sprint)

2.4.1 Sprint 1 : Mise en Place des Bases

- Création de la base de données et des entités principales.
- Développement des endpoints pour l'authentification et l'inscription.
- Mise en place de l'environnement de développement.

2.4.2 Sprint 2 : Développement des Bases Fonctionnelles

- Ajout des fonctionnalités backend pour la saisie des données utilisateur (poids, activités, nutrition).
- Création des premières interfaces utilisateur avec Angular.
- Intégration des services API pour permettre la communication entre le backend et le frontend.

2.4.3 Sprint 3 : Intégration et Fonctionnalités Avancées

- Développement des graphiques d'analyse pour chaque catégorie de suivi.

- Mise en place de la génération de rapports PDF.
- Intégration complète entre les interfaces utilisateur et les services backend.
- Réalisation des tests unitaires pour valider les fonctionnalités.

2.4.4 Sprint 4 : Finalisation et Optimisation

- Intégration des notifications pour les rappels personnalisés.
- Tests finaux et corrections des bugs.
- Optimisation des performances pour une meilleure expérience utilisateur.
- Préparation de la documentation utilisateur et développeur.

2.5 Analyse des Règles de Gestion (par Sprint)

2.5.1 Sprint 1

- Les utilisateurs doivent s'inscrire avec un email et un mot de passe sécurisé (hachage des mots de passe).
- Validation des données d'inscription avant leur enregistrement dans la base de données.

2.5.2 Sprint 2

- Les données utilisateur doivent être validées avant leur sauvegarde (poids, activités, nutrition).
- Chaque utilisateur ne peut consulter que ses propres données, garantissant la confidentialité.

2.5.3 Sprint 3

- Les graphiques d'analyse doivent refléter les données sur des périodes personnalisées (quotidienne, hebdomadaire, mensuelle).
- Les rapports PDF doivent inclure un résumé des données clés et les graphiques associés.

2.5.4 Sprint 4

- Les rappels doivent être configurés en fonction des préférences utilisateur.
- Le système doit être optimisé pour gérer un grand nombre d'utilisateurs simultanément.

2.6 Besoins et Contraintes

2.6.1 Besoins Techniques

- Développement du backend avec **Spring Boot**.
- Création des interfaces utilisateur avec **Angular**.
- Stockage des données dans une base relationnelle (**MySQL**).
- Utilisation de services API pour la communication entre le backend et le frontend.

2.6.2 Besoins Non Techniques

- Fournir une interface intuitive et ergonomique.
- Permettre une utilisation fluide sur les plateformes web et mobile.
- Assurer la confidentialité et la sécurité des données utilisateur.

2.6.3 Contraintes

- Respect du délai de 5 semaines pour le développement complet.
- Nécessité d'intégrer des fonctionnalités complexes tout en garantissant la stabilité.
- Limitations liées aux outils et technologies choisis.

2.6.4 Exigences

- Le système doit être sécurisé pour protéger les données sensibles.
- Les performances doivent être optimisées pour garantir une réponse rapide.
- La compatibilité doit être assurée avec les navigateurs modernes.

2.7 Conclusion

Cette étude préliminaire et fonctionnelle établit une vision claire et structurée du projet. Elle met en évidence les principales fonctionnalités à développer, les règles de gestion à respecter, et les contraintes à prendre en compte pour assurer le succès du projet.

Chapitre 3

Analyse et Conception

3.1 Introduction

Ce chapitre est consacré à l’analyse et à la conception de l’application. Nous détaillerons les choix méthodologiques et les outils utilisés pour modéliser le système, en mettant l’accent sur la structuration et la représentation graphique des fonctionnalités principales. La méthodologie adoptée repose sur l’utilisation du langage de modélisation UML (Unified Modeling Language) pour concevoir les différents aspects du système.

3.2 Le Choix d’UML

Pour la modélisation de l’application, nous avons opté pour la modélisation UML. L’UML (Unified Modeling Language ou Langage de modélisation unifiée en français) est un langage graphique de modélisation informatique. Ce langage est désormais la référence en modélisation objet, ou programmation orientée objet. Cette dernière consiste à modéliser des éléments du monde réel (immeuble, ingrédients, personne, logos...) ou virtuel (temps, prix, compétence...) en un ensemble d’entités informatiques appelées « objet ». [1]

Les raisons de leur choix sont multiples :

- l’UML facilite la communication, simplifie la complexité et améliore la qualité du travail.
- l’UML n’est pas liée à un langage de programmation spécifique, ce qui la rend compatible avec tous les langages.

3.3 Diagrammes Utilisés

Pour assurer une modélisation complète et détaillée du projet, plusieurs types de diagrammes UML ont été utilisés :

3.3.1 Diagramme de Classe

Le diagramme de classe représente la structure statique du système. Il met en évidence les entités principales, leurs attributs, leurs méthodes, ainsi que les relations entre elles. Ce diagramme est essentiel pour concevoir la base de données et le modèle objet.

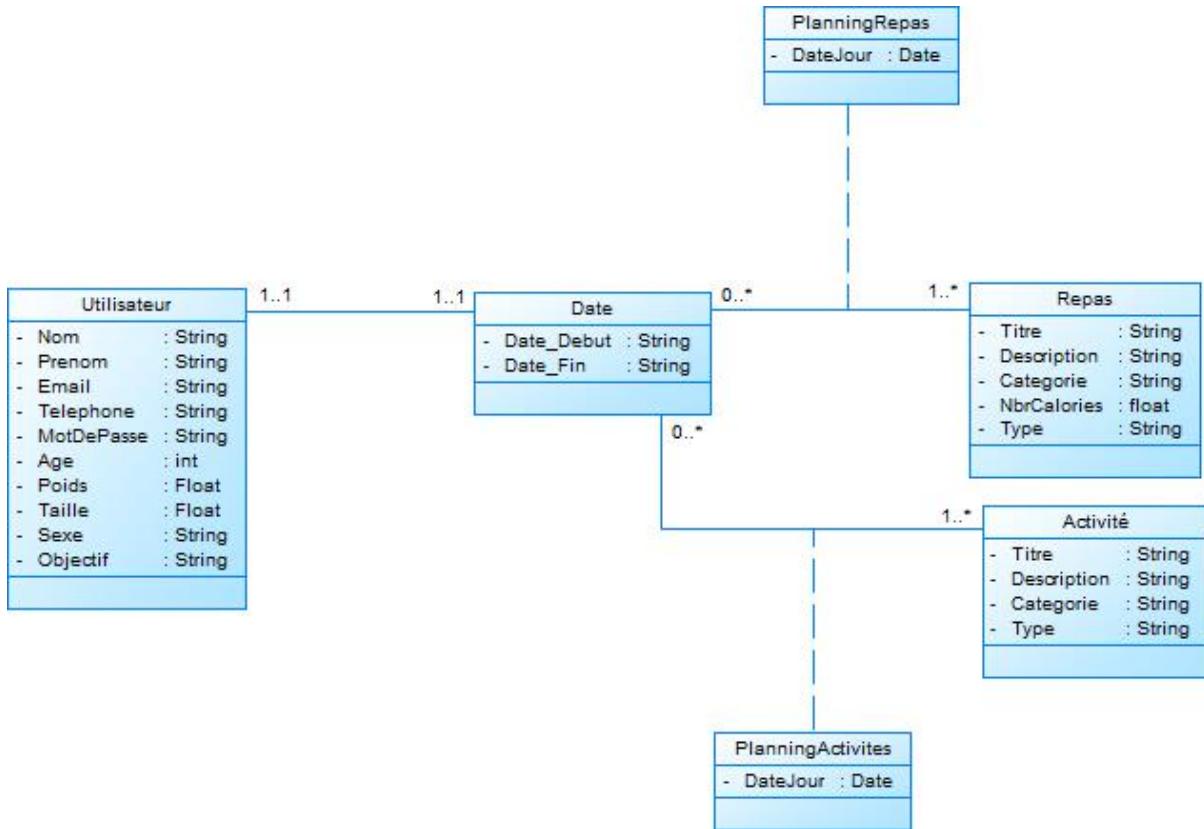


FIGURE 3.1 – Diagramme de classe : Modèle conceptuel du système

Classe	Description
Utilisateur	Représente un utilisateur du système avec ses informations personnelles (nom, email, poids, objectif, etc.).
Repas	Décrit un repas avec ses attributs (titre, description, catégorie, nombre de calories, etc.).
Activité	Décrit une activité avec ses attributs (titre, description, catégorie, etc.).
PlanningRepas	Gère la planification des repas par date.
PlanningActivités	Gère la planification des activités par date.

TABLE 3.1 – Description du diagramme de classe

3.3.2 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation met en évidence les interactions entre les utilisateurs (*acteurs*) et le système. Il identifie les fonctionnalités principales de l'application et montre comment elles sont utilisées.

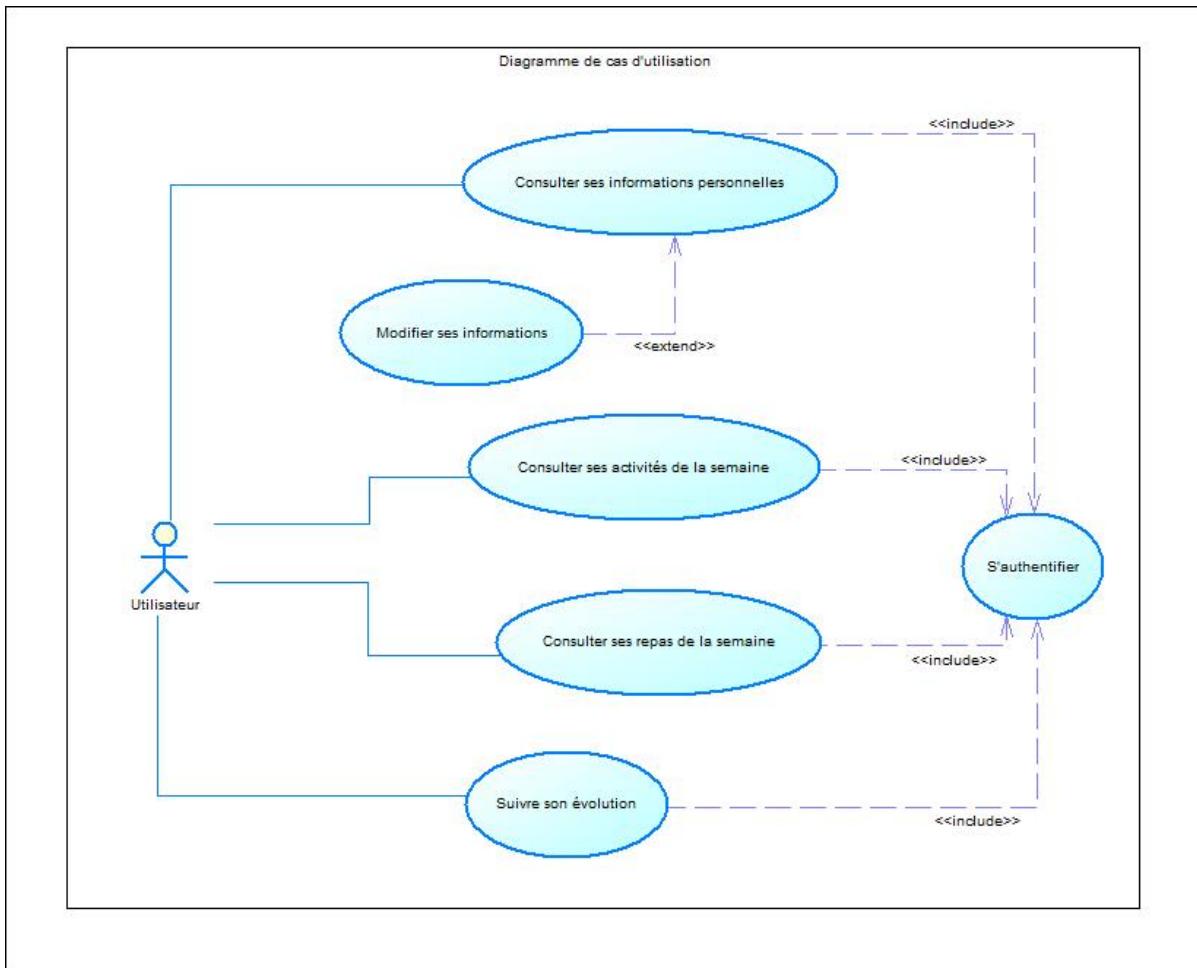


FIGURE 3.2 – Diagramme de cas d'utilisation

Cas d'utilisation	Description
S'authentifier	Permet à l'utilisateur de se connecter au système pour accéder à ses fonctionnalités.
Consulter ses informations personnelles	Permet à l'utilisateur de visualiser ses données personnelles (poids, objectif, etc.).
Modifier ses informations	Permet à l'utilisateur de mettre à jour ses informations personnelles.
Consulter ses activités de la semaine	Permet à l'utilisateur de visualiser les activités prévues pour la semaine.
Consulter ses repas de la semaine	Permet à l'utilisateur de visualiser les repas prévus pour la semaine.
Suivre son évolution	Permet à l'utilisateur de suivre ses progrès au fil du temps.

TABLE 3.2 – Description du diagramme de cas d'utilisation

3.3.3 Diagramme de Séquence

Le diagramme de séquence est utilisé pour modéliser le comportement dynamique du système. Il montre l'interaction entre les différents objets et les acteurs à travers une séquence d'événements ordonnés dans le temps.

Diagramme de séquence : Consultation des activités

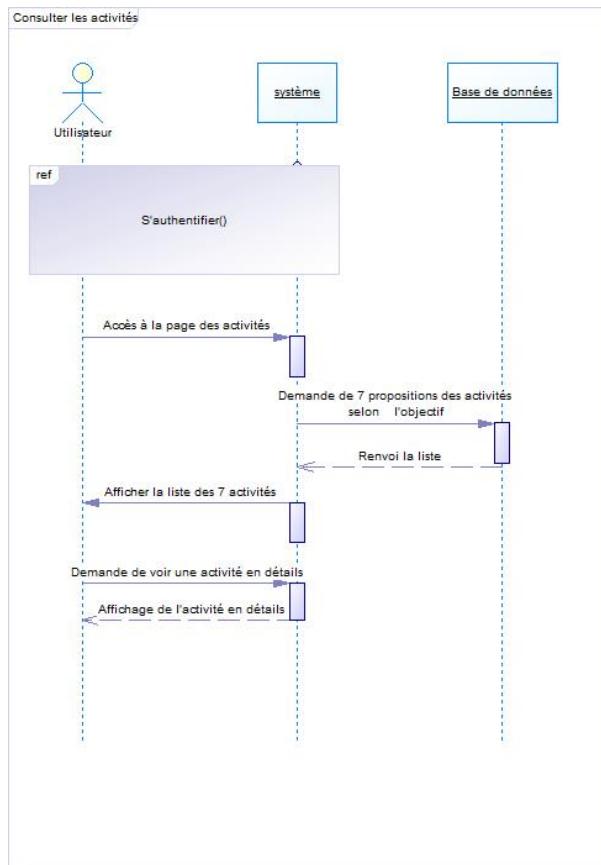


FIGURE 3.3 – Diagramme de séquence : Consultation des activités

Élément	Description
Acteur	L'utilisateur authentifié qui interagit avec le système pour consulter les activités.
Système	Le système reçoit la requête de l'utilisateur et interagit avec la base de données pour fournir des données.
Base de données	Stocke les informations sur les activités et renvoie la liste des activités correspondant à l'objectif.
Description	L'utilisateur s'authentifie, accède à la liste des activités, et peut consulter les détails d'une activité spécifique.

TABLE 3.3 – Description du diagramme de séquence : Consultation des activités

Diagramme de séquence : Consultation des repas

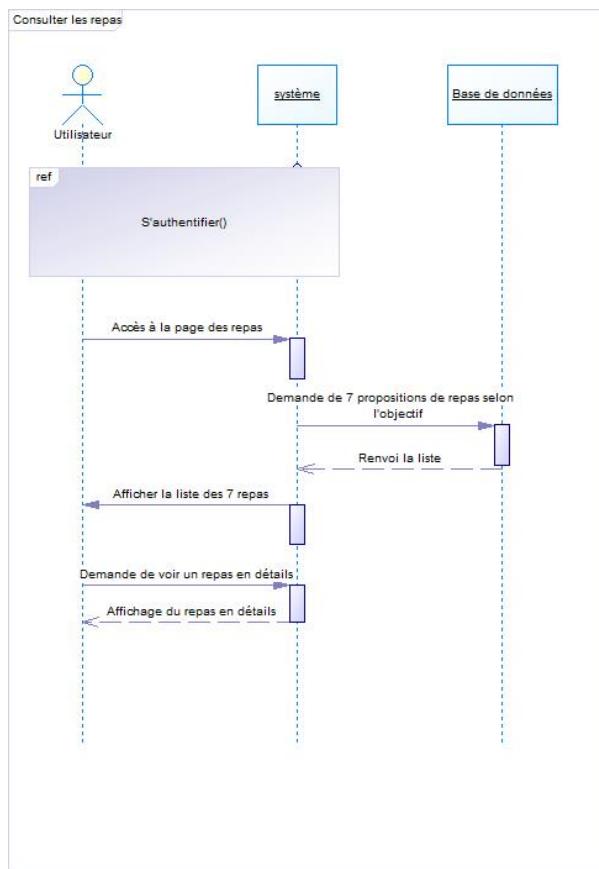


FIGURE 3.4 – Diagramme de séquence : Consultation des repas

Élément	Description
Acteur	L'utilisateur authentifié qui interagit avec le système pour consulter les repas.
Système	Le système reçoit la requête de l'utilisateur et interagit avec la base de données pour fournir des données.
Base de données	Stocke les informations sur les repas et renvoie la liste des repas correspondant à l'objectif.
Description	L'utilisateur s'authentifie, accède à la liste des repas, et peut consulter les détails d'un repas spécifique.

TABLE 3.4 – Description du diagramme de séquence : Consultation des repas

3.3.4 Diagramme de Composant

Le diagramme de composant représente les différents modules logiciels du système et leurs interrelations. Il permet de visualiser l'architecture logicielle du projet et de planifier l'intégration des composants.

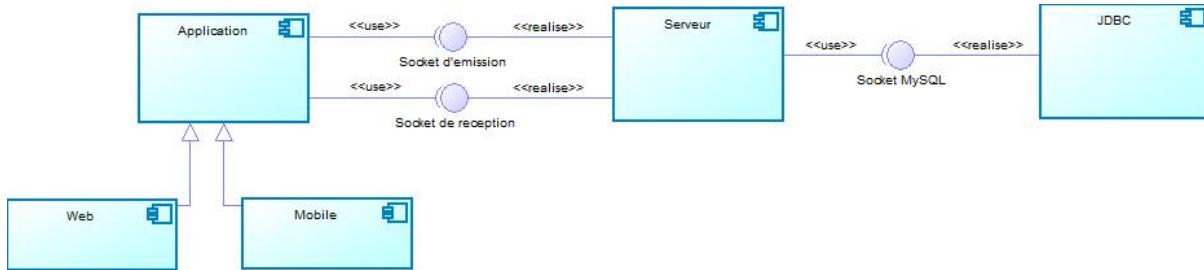


FIGURE 3.5 – Diagramme de composant : Architecture logicielle du système

Composant	Description
Application	Composant principal regroupant les interfaces utilisateur pour le web et le mobile.
Serveur	Gère la logique métier et les interactions entre l'application et la base de données.
JDBC	Fournit l'accès à la base de données via des sockets MySQL.
Web	Interface utilisateur pour la plateforme web.
Mobile	Interface utilisateur pour la plateforme mobile.

TABLE 3.5 – Description du diagramme de composant

3.4 Conclusion

L'utilisation d'UML et de ses différents diagrammes offre une vision claire et détaillée du système, assurant une conception bien structurée et facilement compréhensible. Les diagrammes produits dans ce chapitre constituent une base solide pour le développement de l'application, en guidant les développeurs dans la mise en œuvre des fonctionnalités et des interactions planifiées.

Chapitre 4

Technologies et Réalisation

4.1 Introduction

Ce chapitre illustre la concrétisation de notre projet, passant de la théorie à la réalité. Nous explorerons le processus de création de l'application, mettant en lumière les choix techniques et les pratiques qui ont donné vie à notre plateforme.

4.2 Architecture applicative de l'application

Cette figure présente l'architecture utilisée pour ce projet, de l'interaction du client avec le système jusqu'à la base de données qui sert à répondre à ses besoins.

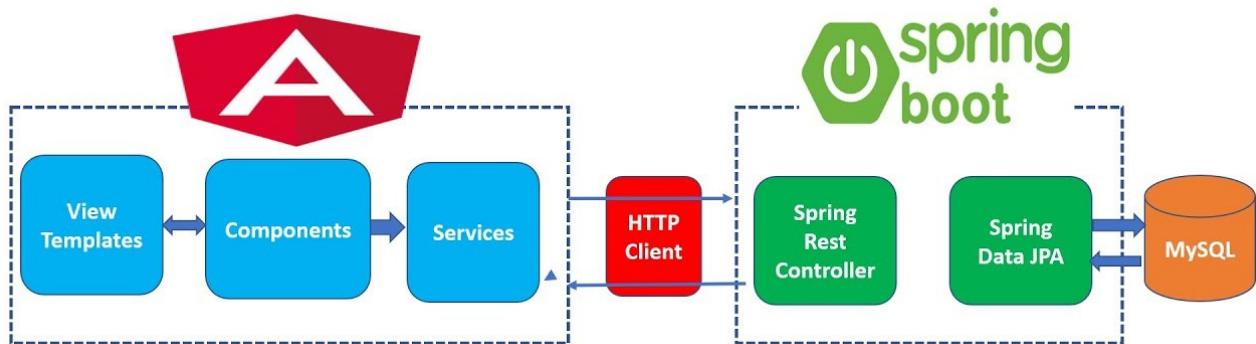


FIGURE 4.1 – Architecture applicative.

4.3 Environnement de développement technique

- **Java Spring Boot** : est un outil qui accélère et simplifie le développement d'applications Web et de microservices avec Spring Framework grâce à trois fonctionnalités principales : Configuration automatique. Approche directive de la configuration. Possibilité de créer des applications autonomes. [2]
- **Angular** : Angular est un framework de développement web open-source, maintenu par Google, qui permet de créer des applications web dynamiques et interactives. Il est basé sur le langage TypeScript, une surcouche de JavaScript, et est conçu pour simplifier le développement des interfaces utilisateur riches et complexes.[3]

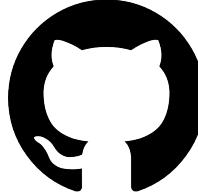


- **Axios API** : est un client HTTP basé sur les promesses pour node.js et le navigateur. Il est isomorphe (= il peut s'exécuter dans le navigateur et nodejs avec le même code source). Côté serveur, il utilise le module natif http de node.js, tandis que côté client (navigateur), il utilise des requêtes XMLHttpRequests. [4]



4.4 Environnement de développement logiciel

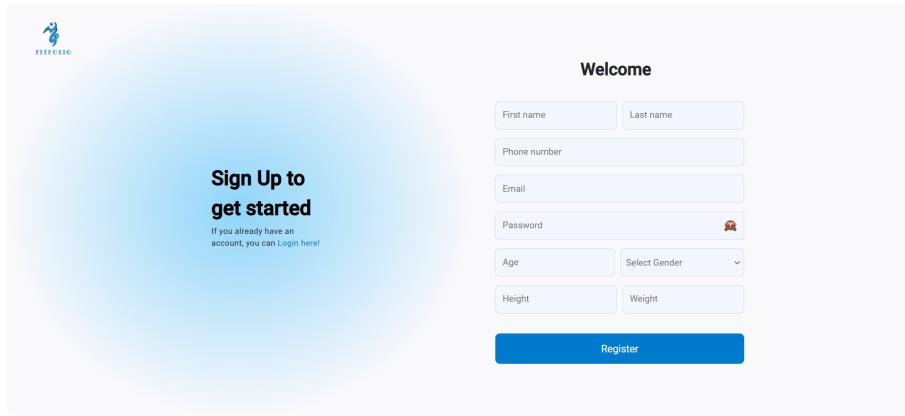
- **MySQL** : est un service de base de données entièrement géré pour déployer des applications natives du cloud en utilisant la base de données open source la plus populaire au monde. [5]
- **Postman** : est une plateforme API pour la création et l'utilisation d'API. Postman simplifie chaque étape du cycle de vie de l'API et rationalise la collaboration afin que vous puissiez créer de meilleures API, plus rapidement. [6]
- **GitHub** : est un service d'hébergement Open-Source, permettant aux programmeurs et aux développeurs de partager le code informatique de leurs projets afin de travailler dessus de façon collaborative. On peut le considérer comme un Cloud dédié au code informatique.[7]



4.5 Mise en oeuvre

4.5.1 Interface d'inscription

Cette interface permet aux nouveaux utilisateurs de s'inscrire en fournissant leurs informations personnelles, notamment le prénom, le nom, le numéro de téléphone, l'email, le mot de passe, l'âge, le genre, la taille et le poids. Elle offre également un lien pour retourner à la page de connexion en cas de besoin.



The screenshot shows the registration interface for the FITFOLIO application. At the top left is the EMSI logo. In the center, the word "Welcome" is displayed above a form with the following fields:

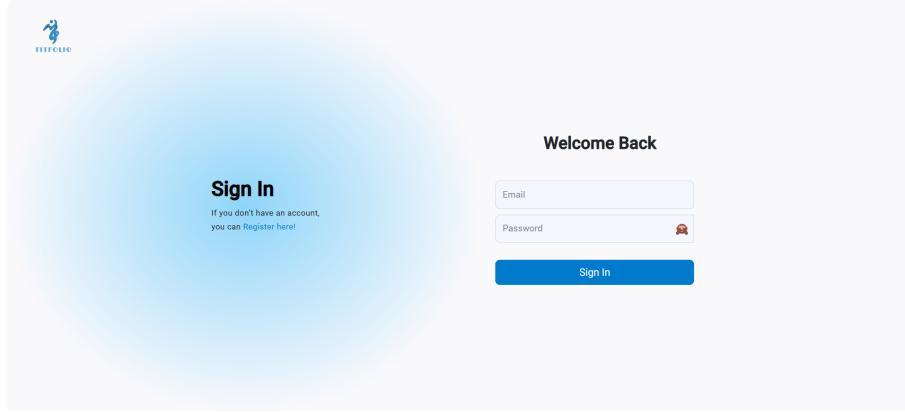
- First name and Last name (two separate input fields)
- Phone number (input field)
- Email (input field)
- Password (input field with a visibility icon)
- Age (input field)
- Select Gender (dropdown menu)
- Height (input field)
- Weight (input field)

At the bottom is a large blue "Register" button.

FIGURE 4.2 – Interface d’inscription.

4.5.2 Interface de connexion

Cette interface permet aux utilisateurs existants de se connecter à l’application en saisissant leur email et leur mot de passe. Elle offre également un lien pour accéder à la page d’inscription au cas où l’utilisateur n’aurait pas encore de compte.



The screenshot shows the login interface for the FITFOLIO application. At the top left is the EMSI logo. In the center, the words "Welcome Back" are displayed above a form with the following fields:

- Email (input field)
- Password (input field with a visibility icon)

At the bottom is a blue "Sign In" button.

FIGURE 4.3 – Interface de connexion.

4.5.3 Tableau de bord - Accueil

Cette interface sert de point d’entrée principal pour l’utilisateur. Elle affiche un résumé des informations clés telles que l’objectif du jour, le plan de repas, l’IMC, la consommation d’eau, le suivi du sommeil et les calories restantes. Elle intègre également un bouton pour télécharger un rapport PDF.

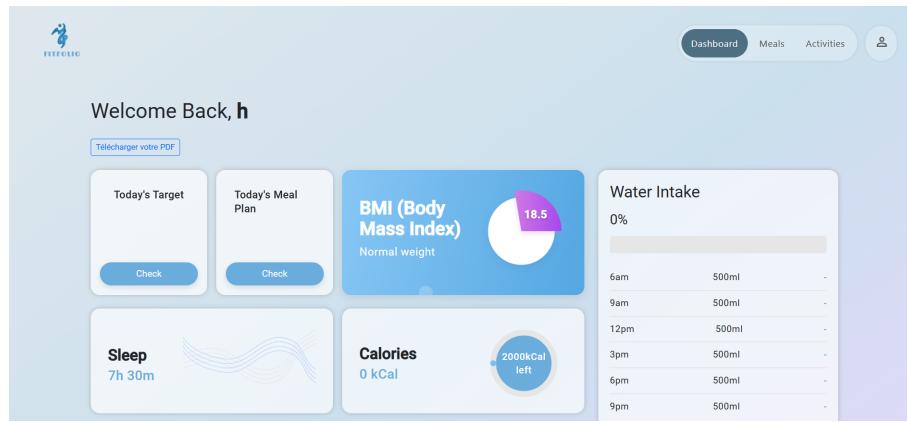


FIGURE 4.4 – Tableau de bord - Résumé.

4.5.4 Suivi de progression

Cette section du tableau de bord affiche un graphique représentant la progression hebdomadaire de l'utilisateur. Elle permet de suivre les tendances des activités physiques ou des objectifs atteints au fil de la semaine, fournissant des visualisations claires pour l'amélioration continue.

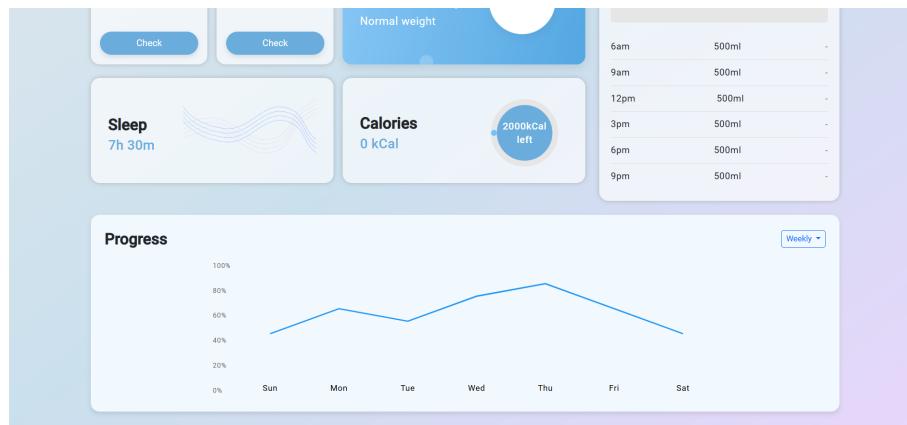


FIGURE 4.5 – Suivi de progression hebdomadaire.

4.5.5 Rapport de suivi de santé

Cette interface génère un rapport de suivi de santé personnalisé pour l'utilisateur, incluant des informations essentielles telles que le poids, la taille, l'indice de masse corporelle (IMC), l'objectif, et un résumé des calories consommées et brûlées. Elle affiche également un graphique montrant l'évolution hebdomadaire des performances physiques.

Rapport de suivi de santé

Date: 12/16/2024

Full Name: h h

Age: 20 years old

Gender: Male

Objectif: Lean & Tone

Weight: 60 kg

Height: 1.8 m

BMI: 18.5 : Normal weight

Moyenne de calories

0 kCal 0 kCal

Consommées Brûlées

Workout Progress (Weekly)



FIGURE 4.6 – Rapport de suivi de santé.

4.5.6 Plan de repas personnel

Cette interface permet à l'utilisateur de visualiser son plan de repas personnel pour une journée ou une semaine donnée. Les repas sont organisés en fonction des moments de la journée : petit-déjeuner, déjeuner, dîner et souper. Chaque carte de repas inclut une image et un titre descriptif.

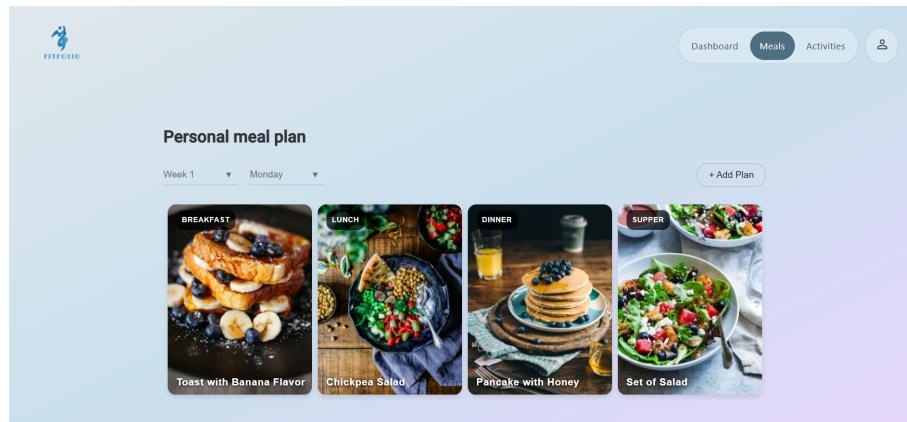


FIGURE 4.7 – Interface du plan de repas personnel.

4.5.7 Sélection des repas

Cette interface permet à l'utilisateur de personnaliser son plan de repas en sélectionnant des plats spécifiques pour chaque moment de la journée. Elle inclut également des détails sur les repas, tels que les ingrédients, le temps de préparation, et une description.

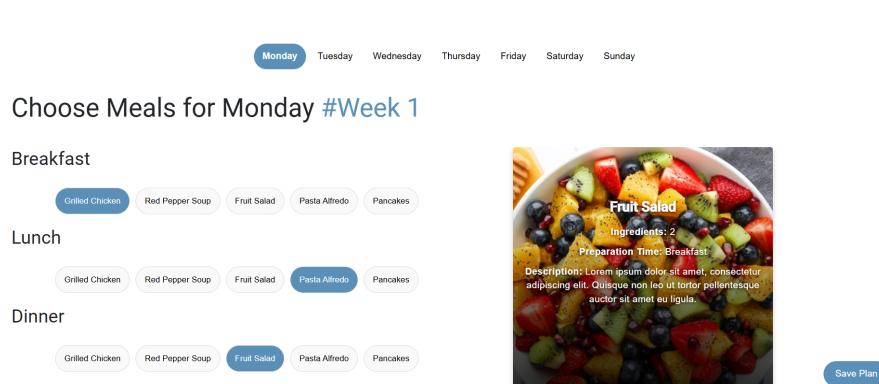


FIGURE 4.8 – Interface de sélection des repas.

4.5.8 Planning des repas

Cette interface permet à l'utilisateur de planifier ses repas pour chaque jour de la semaine. L'utilisateur peut sélectionner des plats pour le petit-déjeuner, le déjeuner et le dîner à partir d'une liste prédéfinie. Une fois les sélections effectuées, le planning peut être enregistré pour une gestion pratique et personnalisée des repas.

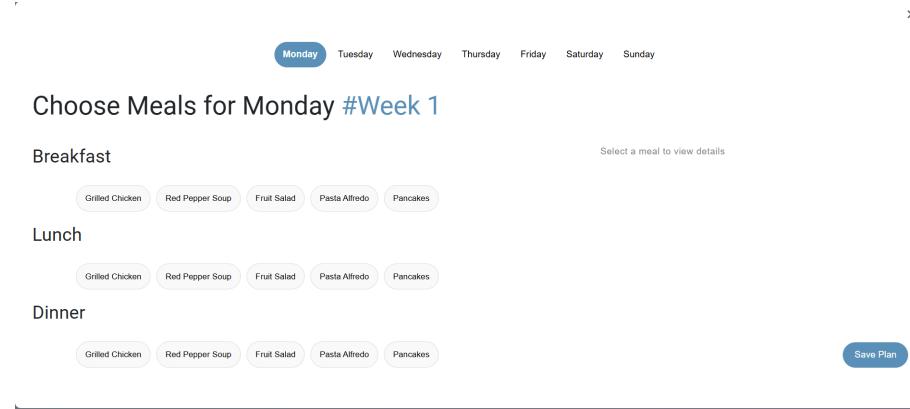


FIGURE 4.9 – Interface de planning des repas.

4.5.9 Choix d'une activité

Cette interface permet à l'utilisateur de choisir une activité physique pour la journée. Une fois sélectionnée, l'activité est affichée avec des détails tels que la description, la vitesse et les calories brûlées.

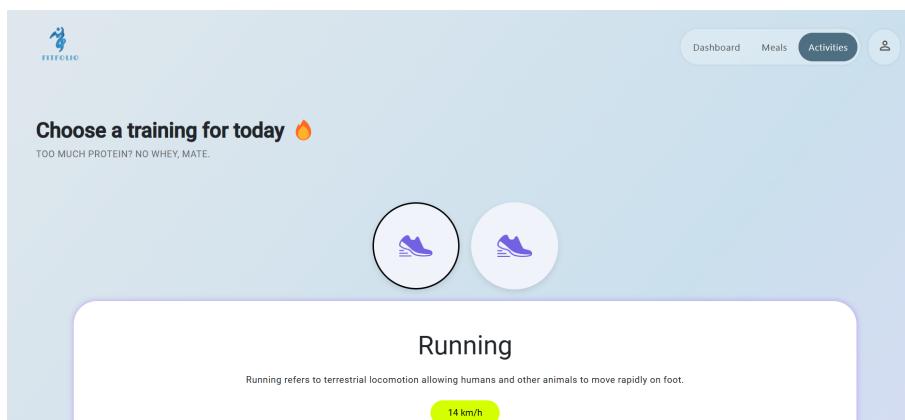


FIGURE 4.10 – Interface de choix d'une activité.

4.5.10 Activité sélectionnée

Cette interface affiche les détails de l'activité choisie par l'utilisateur, y compris le nom de l'activité, une description, la vitesse, et les calories associées. Elle permet à l'utilisateur de valider son choix ou d'en sélectionner une autre.

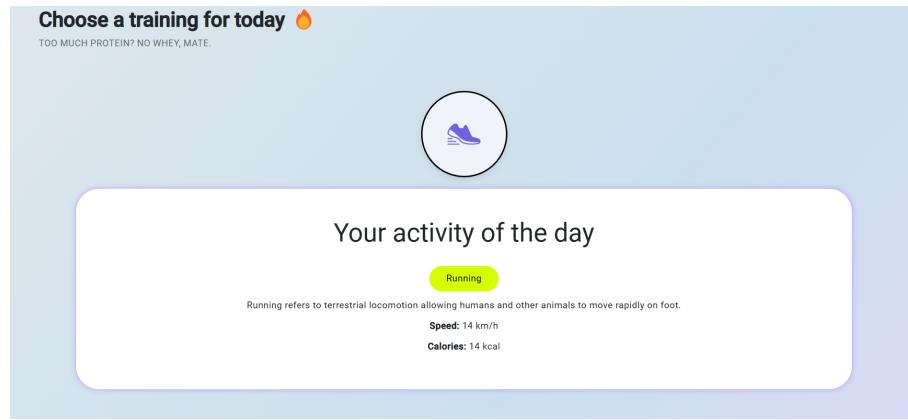


FIGURE 4.11 – Interface de l’activité sélectionnée.

4.5.11 Historique des activités

Cette interface affiche une liste des activités passées de l’utilisateur avec des détails tels que la date et une brève description de chaque activité. Elle permet à l’utilisateur de suivre son évolution physique et ses performances.

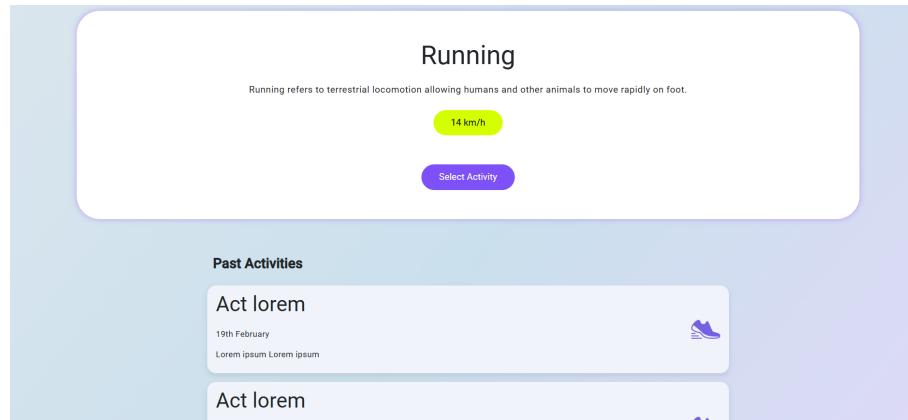


FIGURE 4.12 – Interface de l’historique des activités.

4.6 Conclusion

Au cours de ce chapitre, nous avons abordé tout ce qui concerne la réalisation de notre site web. Nous avons présenté les technologies adoptées lors de l’implémentation de notre application ainsi que les captures d’écran des interfaces.

Conclusion et Perspectives

En conclusion, le projet de développement d'une plateforme web de suivi de santé a été mené à bien, répondant aux objectifs initiaux avec succès. Grâce à une analyse approfondie des besoins fonctionnels et non fonctionnels, une conception méthodique basée sur des diagrammes UML, et une implémentation utilisant des technologies modernes telles que **Spring Boot**, **Angular**, **MySQL**, et **GitHub**, nous avons créé un produit fonctionnel, convivial et adapté aux attentes des utilisateurs.

Les tâches réalisées incluent la mise en place d'un système sécurisé d'inscription et d'authentification, le développement d'une interface utilisateur intuitive permettant de suivre les données de santé telles que le poids, les activités physiques et les apports nutritionnels, ainsi que la création de graphiques et de rapports personnalisés. En outre, la plateforme propose des analyses détaillées et des recommandations adaptées pour permettre aux utilisateurs d'atteindre leurs objectifs de santé.

Pour les perspectives d'avenir, nous envisageons l'intégration de fonctionnalités supplémentaires telles que l'utilisation de l'intelligence artificielle pour fournir des conseils encore plus personnalisés, l'ajout de rappels intelligents pour maintenir la motivation des utilisateurs, et l'expansion de la plateforme vers les appareils mobiles pour une accessibilité accrue. De plus, des efforts seront déployés pour intégrer des capteurs connectés, tels que des montres ou des bracelets de fitness, afin d'automatiser davantage le suivi des données de santé.

En somme, ce projet vise à offrir aux utilisateurs une solution complète et moderne pour surveiller leur santé et atteindre leurs objectifs de bien-être. La réussite de ce projet témoigne de notre engagement à fournir des solutions technologiques innovantes et efficaces pour améliorer la qualité de vie des individus. Nous restons déterminés à poursuivre le développement et l'amélioration de cette plateforme afin de répondre aux besoins évolutifs des utilisateurs.

Bibliographie

- [1] *UML*: <https://www.futura-sciences.com/tech/definitions/informatique-uml-3979/>.
- [2] *SpringBoot*: <https://spring.io/projects/spring-framework>.
- [3] *Angular*: <https://v17.angular.io/guide/what-is-angular>.
- [4] *Axios*: <https://axios-http.com/docs/intro>.
- [5] *MySQL*: <https://www.mysql.com/fr/>.
- [6] *Postman*: <https://www.postman.com/>.
- [7] *GitHub*: <https://datascientest.com/github-tout-savoir>.