Cahier des Charges

(Application de Suivi de la Santé et du Bien-être)

Contexte et Problématique

Dans un monde de plus en plus axé sur la santé et le bien-être, de nombreuses personnes cherchent à adopter des habitudes alimentaires plus saines et à améliorer leur condition physique. Le suivi de l'alimentation et de l'activité physique est devenu essentiel pour atteindre ces objectifs. Cependant, il existe une multitude d'applications et de plateformes de suivi de santé, mais elles présentent souvent des limitations qui empêchent une gestion véritablement centralisée et continue des données de l'utilisateur.

Les utilisateurs sont souvent confrontés à plusieurs problèmes :

1. **Fragmentation des données** :
   La plupart des applications de fitness et de nutrition ne permettent pas une synchronisation fluide des données entre différents appareils. De plus, de nombreuses applications exigent de l'utilisateur qu'il se réenregistre ou qu'il perde ses données si l'appareil utilisé change ou si une déconnexion se produit.
2. **Difficulté de suivi continu** :
   Bien que les utilisateurs suivent leurs progrès, il leur manque souvent des outils de visualisation et d'analyse simples et efficaces pour comprendre si leurs efforts sont en ligne avec leurs objectifs. Les graphiques et rapports manquent parfois de personnalisation et ne permettent pas une lecture rapide des progrès.
3. **Manque de flexibilité et de personnalisation** :
   Les applications de suivi alimentaire sont souvent trop rigides et ne s'adaptent pas aux régimes alimentaires spécifiques des utilisateurs, comme le végétarisme, le véganisme, ou des régimes particuliers pour des besoins médicaux. De plus, les utilisateurs veulent pouvoir suivre leurs repas, mais aussi avoir une bonne gestion de leurs activités physiques en parallèle.
4. **Sécurisation des données** :
   Les utilisateurs veulent que leurs informations personnelles (poids, taille, préférences alimentaires, etc.) et leurs données de santé soient stockées de manière sécurisée et accessibles sur plusieurs appareils. L'absence d'un système d'authentification sécurisé peut engendrer un manque de confiance.

Problématique

Le but de ce projet est de résoudre les défis rencontrés par les utilisateurs en proposant une **application de suivi de fitness et de nutrition** qui :

1. **Assure une continuité et une synchronisation des données entre plusieurs appareils** :
   Les utilisateurs doivent pouvoir se connecter depuis n'importe quel appareil iOS (iPhone, iPad) et retrouver toutes leurs informations, quel que soit le périphérique utilisé. En cas de changement de téléphone ou de déconnexion, les utilisateurs doivent retrouver toutes leurs données (activités physiques, repas, objectifs, etc.) sans perdre aucune information.
2. **Offre un suivi complet de l'alimentation et de l'activité physique** :
   L'application doit permettre aux utilisateurs d'enregistrer et suivre leur consommation alimentaire et leurs activités physiques sur une période donnée. Elle doit proposer des fonctionnalités pour enregistrer les repas, calculer automatiquement les calories et les macronutriments (protéines, glucides, lipides), ainsi que suivre le temps d'entraînement, les calories brûlées, et d'autres métriques liées à l'exercice.

3. **Fournit des analyses et des recommandations visuelles sur les progrès** :
   Il est essentiel que l'application présente des graphiques et des rapports clairs pour que l'utilisateur puisse évaluer ses progrès. Les progrès doivent être visuels et détaillés, permettant une analyse rapide : les calories consommées vs calories brûlées, la progression du poids, et les tendances d'entraînement au fil du temps.
4. **Assure la sécurité des données avec une authentification robuste** :
   L'application doit garantir la sécurité des données sensibles de l'utilisateur, telles que les informations personnelles et les statistiques de santé. Une authentification basée sur **JWT (JSON Web Token)** doit être utilisée pour assurer la sécurité et la confidentialité des échanges de données.
5. **Gère les données dans une base de données centralisée et fiable** :
   Toutes les informations liées à l'utilisateur (repas, activités, objectifs, poids, etc.) doivent être centralisées dans une **base de données MySQL**. Cette base de données sera accessible via une API développée avec **Vapor**, permettant à l'application mobile d'interagir avec le backend pour récupérer et envoyer les données de manière sécurisée et performante.
6. **Offre une expérience utilisateur intuitive et conviviale** :
   L'application doit être simple à utiliser et agréable à naviguer. Les utilisateurs ne devraient pas avoir à chercher longuement pour ajouter un repas ou une activité. Les interfaces doivent être claires, avec des options intuitives pour gérer les données personnelles, consulter les rapports, et saisir les informations quotidiennes de manière rapide.
7. **Facilite la gestion des objectifs et des rapports** :
   L'utilisateur doit pouvoir définir des objectifs précis (poids, calories quotidiennes, nombre d'entraînements, etc.) et consulter des rapports visuels pour voir si ces objectifs sont atteints. L'application devra également proposer des conseils ou des alertes lorsque les objectifs ne sont pas atteints, pour motiver l'utilisateur.
8. **Permet la personnalisation en fonction des besoins individuels** :
   L'application doit permettre à chaque utilisateur de configurer ses préférences : choix du type de régime alimentaire, objectifs personnalisés (par exemple : perdre du poids, prendre de la masse musculaire, etc.), et un historique d'activités et de repas qui reflète leurs habitudes et leurs progrès.

Résumé de la Problématique

L’objectif principal de ce projet est de créer une **application mobile iOS** permettant aux utilisateurs de suivre efficacement leur alimentation et leur activité physique, avec une **base de données centralisée** pour assurer la conservation et la synchronisation des données entre appareils. L’application doit offrir une **expérience fluide et sécurisée** via une authentification JWT et des **analyses visuelles** des progrès de l’utilisateur. La gestion des repas et des activités doit être simple et intuitive, avec des rappels et des recommandations pour améliorer les résultats de l’utilisateur.

Ainsi, l'application doit combiner plusieurs aspects : gestion des données utilisateurs, sécurité des informations sensibles, suivi détaillé et personnalisé des habitudes alimentaires et des entraînements, ainsi que la visualisation de progrès sous forme de graphiques et de rapports. Le but est de fournir aux utilisateurs les outils nécessaires pour atteindre leurs objectifs de santé, tout en assurant une expérience agréable et accessible.

L'application doit stocker toutes les données utilisateur dans une base de données centralisée (MySQL), gérée via une API Vapor. Cela garantit que même si l'utilisateur se déconnecte ou utilise un autre iPhone, ses données restent disponibles.

Objectifs

1. **Assurer une continuité des données utilisateur** :
    - Les informations personnelles, historiques d’activités, repas, objectifs, et paramètres doivent être sauvegardés dans une base de données centralisée pour être accessibles à tout moment.
2. **Offrir un suivi détaillé** :
    - Permettre un suivi journalier et hebdomadaire des repas, activités physiques, et progrès.
    - Fournir des rapports visuels pour aider l’utilisateur à interpréter ses données.
3. **Proposer une solution multiplateforme** :
    - Les données doivent être synchronisées avec le backend, accessible via une API RESTful.
    - Sécuriser les données sensibles (comme le profil utilisateur) via l’authentification JWT.
4. **Garantir une navigation simple et intuitive** :
    - L’interface utilisateur doit être ergonomique, utilisant des graphiques pour visualiser les progrès et des formulaires simplifiés pour l’entrée des données.

Fonctionnalités principales

1. Gestion des Utilisateurs

La gestion des utilisateurs est un élément central de l'application, permettant de créer un profil personnalisé, de suivre les progrès, et de garantir un accès sécurisé aux fonctionnalités. Elle englobe plusieurs actions, telles que l'inscription, la connexion, la gestion du profil, et la déconnexion. Chaque fonctionnalité est accompagnée d'une validation côté client et serveur pour assurer la sécurité et l'intégrité des données de l'utilisateur.

1.1 Inscription

Objectif : Permettre à l'utilisateur de créer un compte pour accéder à l'application.

1. **Interface Utilisateur** :
   L'écran d'inscription doit contenir un formulaire avec les champs suivants :
   - **Nom** : Champ texte pour saisir le nom de l'utilisateur.
   - **Prénom** : Champ texte pour saisir le prénom de l'utilisateur.
   - **Email** : Champ texte où l'utilisateur entre son adresse email. Il est important que cet email soit unique, c'est-à-dire que l'application doit vérifier si un autre utilisateur possède déjà cet email avant d'accepter l'inscription.
   - **Mot de passe** : Champ de mot de passe avec une condition de sécurité, imposant au moins 8 caractères. Le mot de passe doit être validé côté client avant d'être envoyé au serveur, en vérifiant la longueur et la complexité.
2. **Validation côté client** :
   - Vérification de la validité de l'email (format de l'email correct).
   - Vérification de la longueur minimale du mot de passe (8 caractères).
   - Vérification de la correspondance des champs "mot de passe" et "confirmation du mot de passe" si ce dernier est requis.
3. **Enregistrement dans la base de données** :
   - Une fois le formulaire soumis et validé côté client, l'application envoie les données au serveur via une requête HTTP POST.
   - L'API reçoit ces données et génère un identifiant unique (ID utilisateur).
   - L'email et le mot de passe sont sécurisés (le mot de passe est haché avant d'être enregistré en base de données à l'aide d'un algorithme de hachage comme bcrypt).
   - Les informations sont enregistrées dans la base de données (MySQL) avec l'ID généré, le mot de passe haché, le nom, le prénom, et l'email.
   - En cas d'échec (par exemple si l'email est déjà pris), l'API retourne une erreur appropriée.
4. **Réponse de l'API** :
   - Si l'inscription est réussie, l'API renvoie une réponse de succès, et l'utilisateur peut ensuite être redirigé vers l'écran de connexion.
   - En cas d'erreur (comme un email déjà existant), l'API renvoie un message d'erreur indiquant la cause de l'échec.

1.2 Connexion

Objectif : Permettre à l'utilisateur de se connecter à l'application en utilisant ses identifiants (email et mot de passe).

1. **Interface Utilisateur** :
    - L'écran de connexion présente un formulaire où l'utilisateur entre son **email** et son **mot de passe**.
    - Un bouton **"Se connecter"** déclenche l'authentification.
2. **Validation côté client** :
    - Avant d'envoyer les informations au serveur, le formulaire vérifie que les champs email et mot de passe sont remplis correctement (email dans le bon format, mot de passe non vide).
3. **Authentification côté serveur** :
    - Lorsque l'utilisateur soumet ses identifiants, l'application envoie une requête HTTP POST avec l'email et le mot de passe au serveur.
    - L'API compare l'email et le mot de passe avec les données enregistrées dans la base de données.
    - Si l'email est correct mais le mot de passe est incorrect, l'API renvoie un message d'erreur, et l'utilisateur est informé que ses identifiants sont invalides.
    - Si les informations sont valides, l'API génère un **JWT (JSON Web Token)** qui contient des informations d'identification , puis le renvoie à l'utilisateur.
    - Ce token est ensuite stocké côté client, généralement dans le stockage local de l'appareil .
4. **Réponse de l'API** :
    - Si la connexion est réussie, l'utilisateur reçoit un **JWT** qu'il pourra utiliser pour toutes les requêtes suivantes nécessitant une authentification.
    - Si l'authentification échoue, l'API renvoie une erreur avec un message informant l'utilisateur du problème (par exemple, "Mot de passe incorrect").

1.3 Gestion du Profil Utilisateur

Objectif : Permettre à l'utilisateur de consulter et de modifier ses informations personnelles afin de personnaliser son expérience.

1. **Interface Utilisateur** :
    - L'utilisateur peut accéder à une page de **profil** où sont affichées ses informations personnelles :
        - Nom, prénom
        - Email
        - Taille (en cm)
        - Poids (en kg)
        - Objectifs de santé (par exemple, perte de poids, prise de masse, maintien)
        - Préférences alimentaires (régime végétarien, sans lactose, etc.)
    - L'interface propose des champs de saisie pour modifier ces informations.

2. **Modification des informations personnelles** :
    - L'utilisateur peut mettre à jour son **nom**, **prénom**, **email**, et **mot de passe** (si souhaité).
    - L'utilisateur peut également mettre à jour ses **objectifs de santé** (par exemple, objectif de poids à atteindre) et ses **préférences alimentaires**.
3. **Validation côté client** :
    - Lors de la modification du mot de passe, une validation côté client s'assure que le mot de passe respecte les exigences de sécurité (longueur minimale de 8 caractères, etc.).
    - Si l'utilisateur change son email, l'application vérifie que l'email est unique et n'est pas déjà pris par un autre utilisateur.
4. **Mise à jour dans la base de données** :
    - Lorsqu'une modification est soumise, l'API met à jour les informations de l'utilisateur dans la base de données.
    - Le mot de passe est de nouveau haché avant d'être stocké en base.
    - Les autres informations (nom, prénom, taille, poids, objectifs, préférences alimentaires) sont mises à jour dans la base de données.
5. **Réponse de l'API** :
    - Si la mise à jour est réussie, l'API renvoie une confirmation.
    - En cas d'erreur (par exemple, email déjà pris), l'API renvoie une erreur appropriée.

1.4 Déconnexion

Objectif : Permettre à l'utilisateur de se déconnecter en toute sécurité.

1. **Interface Utilisateur** :
    - L'utilisateur peut accéder à un bouton **"Se déconnecter"** dans le menu de l'application.
2. **Suppression du token JWT** :
    - Lorsque l'utilisateur clique sur le bouton de déconnexion, l'application supprime le **JWT** stocké localement sur l'appareil.
    - Cela invalide la session de l'utilisateur et empêche l'accès à des ressources protégées sans se reconnecter.
3. **Réponse de l'API** :
    - La déconnexion côté serveur n'est pas nécessaire puisque la suppression du JWT côté client est suffisante.
    - Après la déconnexion, l'utilisateur est redirigé vers l’interface de connexion.

2. Suivi des Activités Physiques

Le suivi des activités physiques est une fonctionnalité clé permettant à l'utilisateur de suivre son activité sportive, d'enregistrer ses séances, de consulter un historique détaillé, et d'obtenir des statistiques et des suggestions pour améliorer ses performances. Ce suivi contribue à la motivation et à l'atteinte des objectifs de santé et de fitness. L'application propose une interface intuitive permettant une gestion complète des activités physiques.

2.1 Ajout d'une Activité Physique

Objectif : Permettre à l'utilisateur d'enregistrer ses séances d'activité physique, qu'elles soient planifiées ou spontanées, et de suivre ses progrès au fil du temps.

1. **Interface Utilisateur** :
   L'utilisateur peut accéder à un formulaire pour ajouter une nouvelle activité physique. Ce formulaire contient plusieurs champs :
   - **Type d'activité** : Une liste déroulante ou une liste de choix pré-sélectionnée avec des activités courantes comme :
     - Cardio (course, vélo, natation, etc.)
     - Musculation (haltérophilie, exercices de résistance)
     - Yoga
     - Marche
   - **Durée de l'activité** : Un champ numérique en minutes pour saisir la durée totale de la séance.
   - **Calories brûlées** : L'utilisateur peut entrer une valeur estimée ou laisser ce champ vide. Si laissé vide, l'application peut automatiquement calculer une estimation des calories brûlées en fonction du type d'activité et de la durée.

2. **Calcul des calories brûlées (Fonctionnalité optionnelle)** :
   - Si l'utilisateur ne saisit pas les calories brûlées, l'application peut proposer un calcul automatique en utilisant des formules simples basées sur des données moyennes. Par exemple :
     - **Cardio (course, vélo, etc.)** : Le calcul des calories peut être basé sur la durée et l'intensité de l'exercice, avec un facteur spécifique pour chaque type d'activité.
     - **Musculation** : Calcul basé sur la durée et le type d'exercice (par exemple, exercices de poids corporel vs. exercices avec charges).
   - Ces estimations sont basées sur des moyennes et peuvent être ajustées selon des facteurs comme le poids corporel de l'utilisateur, son niveau d'intensité, etc.
3. **Validation côté client et serveur** :
   - Avant de soumettre le formulaire, une validation côté client peut être mise en place pour s'assurer que la durée de l'activité est raisonnable et que les champs obligatoires sont remplis correctement.
   - Une fois le formulaire validé, l'utilisateur soumet les données, et l'API reçoit ces informations pour les enregistrer dans la base de données.
   - L'API vérifie la validité des données (par exemple, si la durée est un nombre positif), puis les enregistre avec un identifiant unique pour chaque séance d'activité.

4. **Réponse de l'API** :
    - Si l'enregistrement de l'activité est réussi, l'API retourne une confirmation et l'activité est ajoutée à l'historique de l'utilisateur.
    - En cas d'erreur, l'API renvoie un message informant l'utilisateur du problème (par exemple, "Erreur lors de l'enregistrement des données").

2.2 Consultation de l'Historique des Activités

Objectif : Permettre à l'utilisateur de visualiser ses activités passées et de suivre ses progrès au fil du temps.

1. **Interface Utilisateur** :
    - L'utilisateur peut accéder à un écran ou une page qui affiche l'historique des activités enregistrées.
    - L'historique est présenté sous forme de **liste chronologique** des séances, avec pour chaque entrée les informations suivantes :
        - **Date de l'activité**
        - **Type d'activité**
        - **Durée**
        - **Calories brûlées** (si renseignées)
2. **Filtrage des Activités** :
    - L'utilisateur peut filtrer ou rechercher ses activités en fonction de plusieurs critères :
        - **Par type d'activité** : Par exemple, afficher uniquement les séances de cardio, de musculation ou de yoga.
        - **Par durée** : Afficher les activités ayant une durée supérieure à un certain nombre de minutes.
        - **Par date** : Visualiser les activités d'une semaine ou d'un mois spécifique.
3. **Tri des Activités** :
    - Les activités peuvent être triées par **date**, de manière croissante ou décroissante.
    - L'utilisateur peut également avoir la possibilité de trier par **type d'activité** ou **durée**.
4. **Réponse de l'API** :
    - L'API renvoie les données de l'historique en fonction des critères de filtrage ou de tri définis par l'utilisateur.
    - En cas de filtrage par date ou type d'activité, l'API envoie uniquement les résultats pertinents.

2.3 Visualisation des Performances (Fonctionnalité optionnelle)

Objectif : Fournir à l'utilisateur des graphiques et des statistiques sur ses performances sportives, afin de mieux suivre ses progrès et de se motiver.

1. **Interface Utilisateur** :
    - L'utilisateur peut consulter des graphiques interactifs sur ses activités physiques, en fonction des données enregistrées.
    - Les graphiques peuvent afficher :
        - **Calories brûlées par jour/semaine** : Un graphique en barres ou en ligne montrant la quantité de calories brûlées sur une période donnée.
        - **Nombre total de minutes d'activité par semaine** : Un graphique représentant le total des minutes d'activité physique par semaine, avec des objectifs visuels (par exemple, une barre de progression par rapport à un objectif hebdomadaire).
2. **Types de Graphiques** :
    - **Graphique en barres** pour la visualisation des calories brûlées ou des minutes d'activité par jour ou par semaine.
    - **Graphique en ligne** pour afficher l'évolution des performances sur une période donnée.
    - **Graphique en secteur** pour comparer le nombre de séances par type d'activité (par exemple, combien de séances de cardio, musculation, yoga, etc. ont été réalisées).
3. **Réponse de l'API** :
    - L'API renvoie les données nécessaires à la génération des graphiques, en fonction des critères choisis par l'utilisateur.
    - Par exemple, pour les calories brûlées, l'API calcule la somme des calories brûlées sur une période donnée et envoie ces données sous forme d'un tableau.

2.4 Suggestions d'Amélioration (Fonctionnalité optionnelle)

Objectif : Aider l'utilisateur à atteindre ses objectifs hebdomadaires en lui proposant des suggestions d'activités à réaliser pour améliorer ses performances.

1. **Calcul des Objectifs** :
    - L'application peut définir un objectif hebdomadaire pour chaque utilisateur (par exemple, **300 minutes d'activité physique par semaine** ou **2000 calories brûlées**).
    - L'utilisateur peut consulter son **objectif hebdomadaire** sur son tableau de bord ou dans ses paramètres de profil.
2. **Analyse des Activités Actuelles** :
    - L'application suit les activités enregistrées au fil de la semaine et compare le total des minutes ou des calories brûlées avec l'objectif hebdomadaire.
    - Si l'utilisateur est en retard par rapport à ses objectifs (par exemple, il a brûlé seulement 1000 calories au lieu des 2000 prévues), l'application propose des suggestions pour atteindre cet objectif.

3. **Suggestions Personnalisées** :
    - L'application peut envoyer une **notification** ou proposer dans l'interface des **suggestions personnalisées** en fonction des données recueillies, par exemple :
        - "Ajoutez 30 minutes de marche pour atteindre votre objectif hebdomadaire de calories."
        - "Il vous manque 45 minutes de musculation pour atteindre votre objectif de temps d'activité."
    - Ces suggestions peuvent être ajustées selon le type d'activités déjà enregistrées et l'intensité des exercices.
4. **Réponse de l'API** :
    - L'API calcule les progrès de l'utilisateur par rapport à son objectif et renvoie des recommandations d'activités si nécessaire.

3. Suivi de la Nutrition (Repas et Aliments)

Le suivi de la nutrition est un élément essentiel pour aider l'utilisateur à contrôler ses apports alimentaires et à atteindre ses objectifs de santé et de bien-être. En enregistrant les repas consommés, les utilisateurs peuvent suivre l'évolution de leurs apports en calories, macronutriments (protéines, glucides, lipides) et micronutriments. Ce suivi est essentiel pour ceux qui cherchent à maintenir une alimentation équilibrée ou à respecter des objectifs spécifiques tels qu'une perte de poids, une prise de muscle, ou une gestion de la santé.

3.1 Ajout d'un Repas

Objectif : Permettre à l'utilisateur d'enregistrer ses repas en précisant les aliments consommés, leur quantité et de calculer automatiquement les valeurs nutritionnelles de chaque repas.

1. **Interface Utilisateur** :
   L'utilisateur peut accéder à un formulaire pour ajouter un repas. Le formulaire comprend plusieurs champs obligatoires et optionnels :
   - **Type de repas** : L'utilisateur choisit parmi une liste de types de repas :
     - Petit-déjeuner
     - Déjeuner
     - Dîner
     - Collation
   - **Aliments consommés** : Une recherche dans une base de données d'aliments pré-remplie permet à l'utilisateur de sélectionner les aliments qu'il a consommés. La base de données peut inclure une grande variété d'aliments courants (par exemple, fruits, légumes, viandes, céréales, etc.). Si l'aliment n'est pas disponible, l'utilisateur peut ajouter un aliment personnalisé.
   - **Quantité consommée** : L'utilisateur saisit la quantité consommée de chaque aliment, en grammes (g) ou en unités (par exemple, 2 œufs, 1 tranche de pain, etc.).
2. **Calcul des Valeurs Nutritionnelles** :
   - Pour chaque aliment, l'application calcule automatiquement les calories et les valeurs nutritionnelles (protéines, glucides, lipides) en fonction de la quantité consommée.
   - **Base de données nutritionnelle** : Chaque aliment de la base de données est accompagné de données nutritionnelles précises. Ces informations peuvent provenir de sources fiables comme des bases de données gouvernementales ou des ressources spécialisées (par exemple, la base de données USDA ou Open Food Facts).
     - Exemple : Pour 100g de poulet rôti, l'application affichera 165 calories, 31g de protéines, 3.6g de lipides, et 0g de glucides.
   - L'application additionne ensuite les valeurs nutritionnelles de chaque aliment pour obtenir un total pour l'ensemble du repas.

3. **Validation côté client et serveur** :
    - Une validation côté client peut être effectuée pour vérifier que les quantités et aliments sont corrects avant de soumettre le repas. Par exemple, l'application peut avertir si l'utilisateur saisit une quantité incorrecte (par exemple, une valeur négative).
    - Après la validation, l'API reçoit les données et enregistre le repas dans la base de données, avec un identifiant unique pour chaque entrée de repas.
    - L'API effectue une vérification des données (par exemple, assure que les valeurs nutritionnelles sont cohérentes avec les aliments enregistrés) avant d'enregistrer le repas.
4. **Réponse de l'API** :
    - Si l'enregistrement est réussi, l'API retourne une confirmation avec les informations nutritionnelles totalisées pour ce repas.
    - Si une erreur survient (par exemple, des données manquantes ou incorrectes), l'API renvoie un message d'erreur informant l'utilisateur du problème.

3.2 Consultation des Repas Enregistrés

Objectif : Permettre à l'utilisateur de consulter la liste de ses repas passés et de visualiser ses apports nutritionnels pour une journée donnée.

1. **Interface Utilisateur** :
   L'utilisateur peut accéder à une page qui affiche la liste de ses repas enregistrés pour une journée donnée. Cette liste inclut :
        - **Type de repas** : Petit-déjeuner, Déjeuner, Dîner, Collation.
        - **Aliment consommé** : Liste des aliments avec leur quantité et les valeurs nutritionnelles (calories, protéines, glucides, lipides).
        - **Calories totales du repas** : Le total des calories pour chaque repas est affiché.
2. **Visualisation des Calories Totales et Apports Nutritionnels** :
    - **Calcul des calories totales de la journée** : L'application calcule la somme des calories pour tous les repas de la journée et affiche cette information sur un tableau de bord.
    - **Répartition des macronutriments** : Les quantités totales de protéines, glucides et lipides sont également calculées et affichées pour chaque repas ainsi que pour la journée complète.
3. **Réponse de l'API** :
    - L'API renvoie les repas enregistrés pour la date spécifiée ainsi que les informations nutritionnelles totales pour chaque repas et pour la journée dans son ensemble.
    - En cas de filtrage ou de recherche, l'API renvoie uniquement les repas correspondant aux critères spécifiés (par exemple, uniquement les repas du déjeuner ou uniquement ceux qui contiennent plus de 500 calories).

3.3 Visualisation des Apports Journaliers

Objectif : Permettre à l'utilisateur de suivre ses apports nutritionnels quotidiens et de les comparer avec ses objectifs personnels.

1. **Interface Utilisateur** :
    - Sur le tableau de bord ou la page des repas, l'utilisateur peut voir un récapitulatif des calories consommées et des apports nutritionnels de la journée.
    - **Objectif quotidien** : L'utilisateur peut définir un objectif calorique quotidien et des objectifs pour chaque macronutriment (par exemple, 2000 calories, 150g de protéines, 250g de glucides, 70g de lipides).
    - **Comparaison avec les objectifs** : L'application compare les calories et les macronutriments consommés aux objectifs définis. Un graphique en temps réel montre les progrès de l'utilisateur par rapport à ses objectifs.
2. **Graphiques et Visualisation (Fonctionnalité optionnelle)** :
    - **Graphique en anneau** : Un graphique en anneau (ou à barres) peut afficher la répartition des macronutriments (protéines, glucides, lipides) dans l'alimentation de l'utilisateur pour la journée. Par exemple, un graphique pourrait montrer que 40% des calories proviennent des glucides, 30% des protéines, et 30% des lipides.
    - **Graphique de calories totales** : Un graphique à barres ou en ligne peut illustrer les calories consommées au cours de la journée et la comparaison avec l'objectif calorique.
3. **Réponse de l'API** :
    - L'API renvoie les apports nutritionnels totaux de la journée et les compare avec l'objectif de l'utilisateur.
    - En cas de dépassement de l'objectif calorique ou d'un macronutriment, l'API peut renvoyer une alerte ou une recommandation.

3.4 Alertes et Recommandations (Fonctionnalité optionnelle)

Objectif : Fournir des alertes et des recommandations pour aider l'utilisateur à respecter ses objectifs nutritionnels.

1. **Alertes** :
    - Si l'utilisateur dépasse son **objectif calorique quotidien**, l'application affiche une alerte : "Vous avez dépassé votre objectif de calories pour aujourd'hui de X calories".
    - Des alertes peuvent aussi être affichées si un macronutriment spécifique est trop faible ou trop élevé par rapport à l'objectif. Par exemple, "Vous n'avez pas atteint votre objectif de protéines pour aujourd'hui".
2. **Recommandations Alimentaires** :
    - L'application peut fournir des recommandations spécifiques pour aider l'utilisateur à atteindre ses objectifs nutritionnels. Par exemple :
        - **"Ajoutez 20g de protéines pour atteindre votre objectif nutritionnel."**

        - **"Consommez 10g de glucides supplémentaires pour atteindre votre quota."**
    - Ces recommandations peuvent être personnalisées en fonction des habitudes alimentaires de l'utilisateur et de ses objectifs personnels (par exemple, perte de poids, prise de muscle).
3. **Réponse de l'API** :
    - L'API analyse les données nutritionnelles et compare les apports aux objectifs définis par l'utilisateur. Elle génère alors des alertes ou des recommandations, selon le cas, et les renvoie à l'interface utilisateur.

4. Gestion des Objectifs

Les objectifs sont essentiels pour aider les utilisateurs à structurer et à suivre leur programme de santé. Qu'il s'agisse de perte de poids, de gain musculaire, de performance physique, ou de gestion d'une alimentation saine, les objectifs permettent aux utilisateurs de se fixer des repères clairs et mesurables. Ces objectifs peuvent être personnalisés selon les besoins individuels et aider à maintenir une motivation constante tout au long du parcours de santé.

4.1 Définition des Objectifs

Objectif : Permettre à l'utilisateur de définir des objectifs personnalisés en matière de poids, de calories, et d'activités physiques. Ces objectifs sont cruciaux pour que l'utilisateur puisse suivre ses progrès et ajuster son programme en fonction de ses besoins.

1. **Objectif de Poids (Fonctionnalité optionnelle)** :
    - L'utilisateur définit un **poids cible** qu'il souhaite atteindre.
    - **Délai** : L'utilisateur spécifie un délai pour atteindre cet objectif (par exemple, "Perdre 5 kg en 3 mois").
    - L’application peut suggérer un objectif basé sur des recommandations standard (par exemple, une perte de poids de 0,5 à 1 kg par semaine), ou permettre à l’utilisateur de personnaliser cet objectif en fonction de ses besoins.
    - **Calcul automatique** : En fonction de l’objectif de poids, l'application peut proposer un **objectif calorique journalier** adapté pour atteindre cet objectif (par exemple, en calculant un déficit calorique de 500 calories par jour pour une perte de poids).
2. **Objectif Calorique Journalier** :
   L'application peut offrir deux options pour l’objectif calorique journalier :
        - **Calcul automatique** : L’application calcule un objectif calorique basé sur le métabolisme basal (BMR) et le niveau d'activité physique de l'utilisateur. Ce calcul tient compte des informations personnelles telles que l’âge, le sexe, la taille, le poids, et l’activité physique.

        - **Personnalisation de l’utilisateur** : L'utilisateur peut saisir son propre objectif calorique en fonction de ses préférences et de ses objectifs spécifiques (par exemple, une prise de masse musculaire nécessitant un surplus calorique).
3. **Objectifs d'Activité Physique** :
    - L’utilisateur peut définir des objectifs spécifiques en matière d’activité physique, comme :
        - **Fréquence d'entraînement** : Par exemple, "3 entraînements par semaine".
        - **Calories brûlées** : Par exemple, "500 calories brûlées par jour" ou "300 calories par séance d’entraînement".
        - **Durée des séances** : Par exemple, "30 minutes d’activité cardio par jour".
    - Ces objectifs d'activité physique peuvent être adaptés en fonction des types d'activités choisies par l’utilisateur (cardio, musculation, yoga, etc.), et peuvent inclure des éléments de progression (par exemple, "Augmenter la durée de la séance de 10 minutes chaque semaine").

4.2 Suivi des Progrès (Optionnel pour une Version Avancée)

Objectif : Offrir à l'utilisateur des outils de suivi visuels et des données qui lui permettent d’observer son évolution par rapport à ses objectifs. Les progrès sont suivis de manière détaillée et visuelle pour favoriser la motivation et aider à ajuster les actions en fonction des résultats.

1. **Graphiques montrant l’évolution du poids** :
    - L’application suit l’évolution du **poids** de l’utilisateur au fil du temps. L'utilisateur peut entrer son poids à intervalles réguliers (par exemple, chaque semaine ou chaque mois).
    - Un graphique linéaire montre l’évolution du poids de l’utilisateur sur une période donnée (par exemple, une courbe représentant la variation du poids sur plusieurs semaines).
    - **Comparaison avec l’objectif de poids** : L’application affiche la progression vers l'objectif de poids. Si l’utilisateur est proche de son objectif ou en retard, des indicateurs visuels (par exemple, un pourcentage atteint) sont fournis pour donner un aperçu rapide de la progression.
2. **Progression vers les objectifs hebdomadaires/mensuels** :
    - L'application permet de suivre l’objectif calorique et d’activité physique sur une base hebdomadaire ou mensuelle. Par exemple :
        - **Calories consommées vs calories brûlées** : Un graphique montre la différence entre les calories consommées et les calories brûlées par l’utilisateur chaque jour, chaque semaine ou chaque mois.
        - **Atteinte des objectifs d'activité physique** : Le suivi des objectifs d'entraînement (nombre de séances ou calories brûlées) est également suivi. Par exemple, un graphique à barres montre si l’utilisateur a atteint son objectif d’entraînements hebdomadaires.
3. **Réponse de l'API** :

- L’API calcule l’évolution du poids, des calories consommées et des calories brûlées en fonction des données saisies par l'utilisateur. Ces données sont ensuite renvoyées sous forme de graphiques interactifs ou de rapports.
- L'API fournit également un calcul dynamique de la progression par rapport aux objectifs d’activité physique, en prenant en compte la fréquence et l'intensité des séances d'entraînement.

4.3 Notifications Internes

Objectif : Fournir des rappels et des messages motivants pour encourager l’utilisateur à respecter ses objectifs et à maintenir une bonne régularité.

1. **Rappels de Progrès** :
   - L'application peut envoyer des rappels réguliers pour aider l'utilisateur à rester sur la bonne voie. Par exemple :
     - **Rappel d'entraînement** : "N'oubliez pas votre séance de cardio aujourd'hui ! Vous êtes à 80 % de votre objectif hebdomadaire d'activité".
     - **Rappel de poids** : "Avez-vous pesé aujourd'hui ? Vous êtes sur la bonne voie pour atteindre votre objectif de poids".
2. **Messages Motivants** :
   - En fonction des progrès de l'utilisateur, l'application peut envoyer des messages positifs pour renforcer la motivation. Par exemple :
     - **"Bravo ! Vous avez atteint 90 % de votre objectif calorique pour cette semaine !"**
     - **"Félicitations ! Vous avez perdu 1 kg ce mois-ci, continuez comme ça !"**
   - Des messages peuvent aussi être envoyés pour encourager l'utilisateur en cas de difficulté, comme :
     - **"Ne vous découragez pas ! Vous êtes à seulement 200 calories de votre objectif quotidien."**
     - **"Un petit écart ne fait pas tout un échec. Reprenez demain avec encore plus d'énergie !"**
3. **Réponse de l'API** :
   - L'API envoie des notifications sur la base des progrès de l'utilisateur. Par exemple, si l'utilisateur atteint 80 % de son objectif d'activité physique, une notification de rappel peut être générée automatiquement.
   - Ces notifications peuvent être configurées pour apparaître à des moments spécifiques de la journée ou selon un planning défini par l'utilisateur (par exemple, tous les soirs à 18h).

5. Historique et Rapports

L'historique et les rapports permettent à l'utilisateur de consulter et d'analyser ses données de manière détaillée, afin de suivre ses progrès, d'identifier des tendances, et de prendre des décisions éclairées pour optimiser son programme de santé. Ces fonctionnalités offrent une vue d'ensemble sur les habitudes de l'utilisateur en matière d'activité physique et de nutrition, et permettent également un suivi approfondi de ses objectifs.

5.1 Historique des Activités et Repas

Objectif : Fournir un moyen d'accéder facilement à toutes les données enregistrées, permettant à l'utilisateur de voir l'évolution de ses habitudes et de son engagement au fil du temps.

1. **Vue mensuelle ou hebdomadaire** :
    - L'utilisateur peut choisir la période d'affichage des données sous forme de **vue mensuelle** ou **vue hebdomadaire**, ce qui lui permet de visualiser ses activités physiques et repas enregistrés sur la durée qui l'intéresse.
    - La **vue mensuelle** offre un aperçu global, avec des résumés des calories brûlées, du nombre d'entraînements effectués, des repas consommés, et des calories totales par jour pour chaque mois.
    - La **vue hebdomadaire** présente les activités et repas au jour le jour, avec un suivi plus détaillé des habitudes et des progrès à court terme.
2. **Possibilité de rechercher par date** :
    - L'utilisateur peut rechercher ses activités ou repas enregistrés sur des dates spécifiques ou des plages de dates personnalisées (par exemple, "du 1er au 7 novembre").
    - Un champ de recherche ou un calendrier interactif permet de sélectionner une date précise pour afficher l'historique des données de cette journée.
    - La recherche par date est utile pour revenir sur un événement particulier (par exemple, "Quel a été mon repas le 15 octobre ?") ou pour analyser des périodes spécifiques d'entraînement ou de nutrition.
3. **Consultation détaillée des repas et activités** :
    - Pour chaque journée sélectionnée, l'utilisateur peut consulter les **détails des repas** enregistrés (types de repas, aliments consommés, calories et macronutriments) ainsi que les **détails des activités** (type d'activité, durée, calories brûlées, etc.).
    - Ces données sont présentées sous forme de liste ou de tableau détaillé, permettant une lecture facile et rapide.

5.2 Rapports Visuels (Fonctionnalité optionnelle)

Objectif : Fournir des statistiques visuelles sur les progrès de l'utilisateur à travers des graphiques clairs, permettant une meilleure compréhension des données et une analyse facile de l'évolution dans le temps.

1. **Statistiques Globales sur une Période Donnée** :
    - L'utilisateur peut consulter des **statistiques globales** sur des périodes spécifiques : hebdomadaire, mensuelle ou annuelle. Ces rapports agrègent les données de manière claire et condensée, offrant un aperçu complet des efforts de l'utilisateur.
        - **Calories consommées vs calories brûlées** : Un graphique compare les calories consommées au cours de la période avec celles brûlées lors des activités physiques. Cela permet à l'utilisateur de vérifier s'il respecte son objectif calorique ou s'il se trouve dans un déficit ou excédent.
        - **Moyenne hebdomadaire de temps d'entraînement** : Ce rapport présente la moyenne du temps passé à faire de l'exercice chaque semaine, ce qui permet à l'utilisateur de vérifier sa constance dans ses efforts physiques.
        - **Progrès vers les objectifs** : Un graphique montre la progression de l'utilisateur vers ses objectifs de poids, de calories brûlées et d'activité physique. Par exemple, un graphique linéaire ou à barres peut être utilisé pour visualiser l'évolution du poids au fil du temps.
2. **Graphiques linéaires, en barres ou circulaires** (Optionnel pour une version avancée) :
    - Des **graphes linéaires** peuvent être utilisés pour visualiser des tendances sur de longues périodes, par exemple l'évolution du poids ou des calories brûlées au fil des semaines ou des mois.
    - Des **graphes en barres** permettent une comparaison de différentes périodes (par exemple, calories brûlées par semaine ou par mois).
    - Des **graphes circulaires** (ou en anneau) peuvent être utilisés pour visualiser la répartition des macronutriments dans les repas, indiquant par exemple le pourcentage de protéines, de glucides et de lipides consommés durant la journée ou la semaine.
3. **Consultation des données nutritionnelles et d'activité dans les rapports** :
    - Les rapports visuels affichent également les données détaillées sur les **repas** (calories, macronutriments) et les **activités physiques** (durée, calories brûlées). Ces informations sont affichées sous forme de graphiques à secteurs ou à barres pour une meilleure compréhension.
    - **Objectifs alimentaires et activités physiques** : Les graphiques montrent si l'utilisateur a atteint ses objectifs nutritionnels ou d'entraînement, ou s'il reste encore des efforts à fournir.

5.3 Téléchargement des Rapports (Fonctionnalité optionnelle)

Objectif : Permettre à l'utilisateur de télécharger des rapports personnalisés pour une consultation en dehors de l'application ou pour un archivage personnel.

1. **Export des données sous forme de fichier (PDF ou CSV)** :
    - L'utilisateur peut télécharger un rapport détaillé de ses **activités physiques** et **repas** sous forme de fichier **PDF** ou **CSV** (format tableur).
        - **Format PDF** : Un fichier PDF bien structuré qui présente les statistiques visuelles (graphes) ainsi que les données textuelles sous forme de résumé. Cela permet à l'utilisateur de conserver un historique imprimable ou à partager avec un professionnel de la santé.
        - **Format CSV** : Un fichier CSV contenant les données brutes sur les repas et les activités physiques. Ce format est utile pour ceux qui souhaitent analyser les données avec des outils externes (comme un tableur Excel).
    - **Personnalisation des rapports** : L'utilisateur peut choisir la période à exporter (par exemple, "Rapport de novembre 2024" ou "Rapport des 3 derniers mois"). Il peut également choisir de n'exporter que certaines catégories de données (par exemple, uniquement les repas ou les activités).
2. **Partage des rapports** :
    - Une fois téléchargé, l'utilisateur peut facilement partager ces rapports par email ou via des plateformes de stockage cloud (Google Drive, Dropbox, etc.) pour les consulter sur d'autres appareils ou les partager avec un nutritionniste, un coach sportif ou un médecin.
3. **Utilisation des données exportées** :
    - L'exportation de données permet à l'utilisateur d'effectuer une analyse plus poussée de ses habitudes et d'adapter son programme de santé en fonction de ses résultats. Par exemple, il pourrait examiner les tendances sur plusieurs mois ou analyser les moments où il a atteint ou dépassé ses objectifs.

6. Administration (Fonctionnalité optionnelle)

L'interface d'administration permet à un administrateur de gérer l'ensemble des données globales de l'application. Bien que cette fonctionnalité soit facultative, elle est utile pour un contrôle centralisé sur les utilisateurs, les repas, les exercices, et pour obtenir des statistiques globales. Elle permet également de maintenir la base de données à jour, d'analyser l'engagement des utilisateurs, et d'assurer la conformité avec les règles de l'application.

6.1 Gestion des Utilisateurs

Objectif : Offrir un contrôle total sur les utilisateurs inscrits, avec des capacités d'interaction pour gérer leur accès et leur comportement sur l'application.

1. **Consulter la liste des utilisateurs** :
    - L'administrateur peut consulter une **liste complète des utilisateurs** inscrits sur l'application, avec des informations de base comme le nom, l'email, le statut (actif ou inactif), et les objectifs définis par chaque utilisateur.
    - Cette liste peut être filtrée et triée par différents critères, comme le nombre d'activités enregistrées, les objectifs atteints, ou la date d'inscription, pour une gestion plus efficace.
2. **Bloquer ou suspendre un utilisateur** :
    - L'administrateur peut choisir de **bloquer temporairement** ou **suspendre définitivement** l'accès à un utilisateur pour diverses raisons (comportement inapproprié, violation des conditions d'utilisation, etc.).
    - Lorsqu'un utilisateur est bloqué, il ne peut plus accéder à son compte, mais les données restent conservées dans la base de données pour une consultation ultérieure ou une réactivation si nécessaire.
3. **Supprimer un utilisateur** :
    - L'administrateur a la possibilité de **supprimer définitivement** un utilisateur et toutes ses données associées (activités, repas, objectifs). Cette option peut être utilisée en cas de demandes de suppression de compte ou pour gérer les utilisateurs inactifs.
    - La suppression d'un utilisateur est irréversible, et une confirmation est demandée avant d'effectuer cette action pour éviter les erreurs.
4. **Rechercher et analyser les utilisateurs** :
    - Des outils de **recherche avancée** permettent de filtrer les utilisateurs par critères spécifiques (par exemple, utilisateurs ayant un objectif de perte de poids, ayant dépassé leur objectif calorique, etc.).
    - Un **tableau de bord** peut afficher des informations statistiques détaillées sur les utilisateurs (par exemple, nombre total d'utilisateurs actifs, répartition par objectifs, fréquence des connexions).

6.2 Gestion des Aliments et Exercices

Objectif : Permettre à l'administrateur de maintenir à jour les bases de données des aliments et des exercices pour garantir la pertinence des informations fournies aux utilisateurs.

1. **Gestion des aliments** :
    - L'administrateur peut **ajouter, modifier ou supprimer des aliments** dans la base de données de l’application.
        - **Ajouter un aliment** : Lors de l'ajout, l’administrateur doit renseigner des informations comme le nom de l’aliment, les valeurs nutritionnelles (calories, protéines, glucides, lipides, etc.), ainsi que des catégories (ex : légumes, viandes, fruits, etc.).
        - **Modifier un aliment** : L'administrateur peut mettre à jour les informations nutritionnelles des aliments existants si elles changent ou si une erreur a été commise lors de l’ajout.
        - **Supprimer un aliment** : L’administrateur peut également supprimer un aliment de la base de données si celui-ci n'est plus pertinent ou doit être retiré.
2. **Gestion des exercices physiques** :
    - L’administrateur peut **ajouter, modifier ou supprimer des types d'exercices** dans la base de données. Ces exercices peuvent être associés à des catégories (ex : Cardio, Musculation, Yoga, etc.).
        - **Ajouter un exercice** : Cela inclut l’ajout du nom de l'exercice, des informations sur les calories brûlées par minute, des groupes musculaires ciblés, et d'autres détails spécifiques comme le niveau de difficulté.
        - **Modifier un exercice** : L’administrateur peut ajuster les informations d’un exercice, par exemple en modifiant les calories brûlées ou en mettant à jour des instructions.
        - **Supprimer un exercice** : Cette option permet de retirer des exercices de la base de données, notamment si un exercice devient obsolète ou n'est plus proposé.
3. **Maintien de la cohérence de la base de données** :
    - L’administrateur doit également veiller à la cohérence des données (par exemple, les valeurs nutritionnelles des aliments et les données relatives aux exercices doivent être exactes et mises à jour régulièrement).
    - Des **outils de validation** peuvent être fournis pour s'assurer que les informations ajoutées respectent des standards de qualité (par exemple, vérifier que les calories d’un aliment sont réalistes).

6.3 Tableau de Bord Administrateur

Objectif : Offrir à l'administrateur une vue d'ensemble des performances globales des utilisateurs et des statistiques liées à l’application.

1. **Statistiques globales sur les utilisateurs actifs** :
    - Le tableau de bord affiche des **statistiques sur les utilisateurs actifs**, comme le nombre total d’utilisateurs inscrits, le nombre d’utilisateurs ayant enregistré des activités ou des repas, et les utilisateurs qui ont atteint leurs objectifs (poids, activité physique, etc.).
    - L’administrateur peut obtenir des **rapports démographiques** sur les utilisateurs (par exemple, répartition des utilisateurs par âge, sexe, objectif de santé) pour comprendre la composition de l'audience de l’application.
2. **Statistiques sur les repas enregistrés** :
    - Le tableau de bord inclut des **données sur les repas enregistrés**, telles que le nombre total de repas ajoutés par les utilisateurs dans une période donnée (par exemple, par semaine, mois, ou année).
    - Ces statistiques peuvent être segmentées en fonction des types de repas (petit-déjeuner, déjeuner, dîner, etc.) ou des catégories alimentaires les plus fréquemment consommées.
3. **Performance des exercices et activités** :
    - Des statistiques peuvent aussi être présentées sur les **activités physiques enregistrées**, avec des graphiques montrant la répartition des types d'exercices (cardio, musculation, yoga, etc.), ainsi que des données sur les calories brûlées par les utilisateurs.
    - Ces rapports permettent à l’administrateur de suivre les tendances dans les comportements d’exercice et d’identifier les activités les plus populaires.
4. **Suivi des objectifs des utilisateurs** :
    - Le tableau de bord présente également des **données sur les objectifs atteints** par les utilisateurs, telles que le nombre d’utilisateurs ayant atteint leur objectif de poids ou d’activité physique, ou ayant respecté leurs objectifs caloriques.
    - Cela permet à l’administrateur de vérifier l'efficacité de l'application en matière de suivi des objectifs et d'adapter les fonctionnalités en fonction des résultats.

6.4 Fonctionnalités Avancées (optionnel)

1. **Gestion des feedbacks utilisateurs** :
    - L’administrateur peut consulter les **avis et retours des utilisateurs**, ce qui permet de prendre des mesures correctives ou d'ajuster les fonctionnalités en fonction des suggestions des utilisateurs.
2. **Gestion des notifications et messages** :
    - L’administrateur peut envoyer des **notifications globales** aux utilisateurs, par exemple, pour annoncer une mise à jour de l’application ou pour rappeler des événements spécifiques (par exemple, des challenges mensuels).
3. **Audits de sécurité et de conformité** :
    - Un outil d’audit peut être inclus pour vérifier la sécurité des données utilisateurs (comme la protection des données personnelles) et garantir la

conformité avec les lois et réglementations locales, telles que le RGPD (Règlement Général sur la Protection des Données).

Contraintes Techniques

1. Architecture et Conception

- **Architecture MVVM** : L'application doit suivre le modèle **MVVM (Model-View-ViewModel)** pour séparer la logique métier, l'affichage de l'interface utilisateur et la gestion des interactions. Cela facilitera la maintenance et l'extensibilité du code.
- **Utilisation de SwiftUI** : L'interface utilisateur doit être développée en utilisant **SwiftUI** pour garantir une expérience moderne et fluide sur les appareils iOS. L'application doit être réactive et capable de se mettre à jour dynamiquement en fonction des changements d'état.

2. Base de Données

- **Base de données relationnelle** : L'application doit utiliser une **base de données relationnelle MySQL** pour stocker les données utilisateurs, les repas, les activités physiques et les objectifs. Il sera nécessaire d'implémenter des relations entre les différentes entités à l'aide de **clés primaires** et **clés étrangères**.
- **Synchronisation des données** : Les données doivent être stockées de manière centralisée et accessible sur tous les appareils de l'utilisateur. En cas de déconnexion ou de changement d'appareil, l'utilisateur doit pouvoir retrouver ses données intactes.
- **Sécurisation des données sensibles** : Les données personnelles des utilisateurs (mots de passe, préférences alimentaires, etc.) doivent être stockées de manière sécurisée. Utilisation d'un chiffrement pour les données sensibles.
- **Normalisation des données** : La base de données doit être normalisée pour minimiser la redondance et garantir l'intégrité des données.

3. Backend et API

- **API RESTful avec Vapor** : L'application mobile doit interagir avec un backend via une API **RESTful**. L'API sera développée avec **Vapor**, et permettra de gérer les utilisateurs, leurs repas, leurs activités physiques, et leurs objectifs. L'API doit être performante, sécurisée et capable de gérer des appels fréquents.
- **Authentification avec JWT** : L'authentification doit se faire via **JSON Web Token (JWT)** pour garantir la sécurité des échanges et des sessions utilisateur. L'utilisateur devra se connecter avec un compte et obtenir un token d'authentification pour chaque session.
- **Gestion des CORS** : L'API devra gérer correctement les **Cross-Origin Resource Sharing (CORS)** pour permettre la communication entre l'application mobile et le backend, tout en respectant les bonnes pratiques de sécurité.

4. Sécurité

- **Protection des données sensibles** : Les données personnelles des utilisateurs (par exemple, les informations de compte et les habitudes de santé) doivent être stockées et transmises de manière sécurisée. Les mots de passe doivent être **hachés** à l'aide d'un algorithme sécurisé avant d'être stockés.
- **HTTPS pour toutes les communications (optionnel pour une version avancée)** : Toutes les communications entre l'application mobile et l'API backend doivent être chiffrées via **HTTPS** pour garantir la sécurité des échanges.
- **Contrôle d'accès et gestion des sessions** : L'application doit vérifier l'identité de l'utilisateur à chaque requête via des tokens JWT. En cas de déconnexion ou d'expiration du token, l'accès aux données sensibles doit être interdit.

5. Performance et Scalabilité

- **Optimisation des requêtes SQL** : Les requêtes à la base de données doivent être optimisées pour garantir des performances rapides, même lorsque le nombre d’utilisateurs et de données augmente.
- **Mise en cache des données (optionnel pour une version avancée)** : Il est recommandé de mettre en place des mécanismes de **mise en cache** pour les données fréquemment consultées, comme les informations relatives aux repas ou aux exercices, afin de réduire la charge sur le serveur et améliorer les temps de réponse.
- **Réduction des appels à l'API (optionnel pour une version avancée)** : L'application mobile doit minimiser le nombre de requêtes API en utilisant des stratégies comme le stockage local de certaines données temporaires ou la récupération groupée des informations nécessaires.

Livrables Exacts Attendues

1. **Conception :**
    - Dictionnaire des données.
    - Modèle Conceptuel de Données.
    - Modèle Logique de Données.
    - Modèle Physique de Données
    - Diagramme de classes.
    - Diagramme de cas d'utilisation.
    - Maquette Figma : représentation visuelle des interfaces utilisateur, conforme aux exigences fonctionnelles.
2. **Base de Données :**
    - Script SQL complet pour créer la base et ses relations.
    - Tables remplies avec des données fictives pour tester.
3. **Backend :**
    - API Vapor avec endpoints documentés.
    - Authentification JWT.
    - Implémentation Fluent et quelques requêtes SQL en dur.
    - CORS.
4. **Application iOS (SwiftUI) :**
    - Application fonctionnelle avec synchronisation complète des données.
    - Écrans requis :
        - Connexion/Inscription.
        - Tableau de bord (calories consommées, activités, graphiques).
        - Suivi des repas.
        - Suivi des activités physiques.
        - Gestion des objectifs et paramètres utilisateur.