

Rapport de application 4



ALAIN RABIE



SOMMAIRE

Contexte	1
Besoin et objectif	2
Analyse Fonctionnelle et Choix Technique	3
Base de donnée	4
Organisation du code	5
Presentation du code	6
Presentation de l'application	7
Annexes	8





BTS SIO 2023-2025

01

CONTEXTE

ALAIN RABIE

1. Contexte

Dans un contexte où les modes de vie deviennent de plus en plus sédentaires et où les déséquilibres alimentaires sont fréquents, les problèmes de santé liés à l'alimentation et au manque d'activité physique ne cessent d'augmenter.

Obésité, diabète, fatigue chronique, troubles cardiovasculaires ou encore perte de motivation sont autant de conséquences d'un quotidien mal équilibré. C'est face à ces constats qu'est née **NutriMotion**, une application mobile pensée pour aider chacun à retrouver un mode de vie plus sain, plus actif et plus équilibré.

NutriMotion ne se limite pas à une simple application de suivi calorique : elle propose une approche complète de la santé en combinant nutrition, activité physique, accompagnement personnalisé et outils de suivi. L'objectif est de fournir une solution intuitive, accessible et motivante pour celles et ceux qui souhaitent prendre soin de leur corps, atteindre des objectifs physiques ou tout simplement se sentir mieux dans leur quotidien.

.....

NutriMotion permet aux utilisateurs de suivre leur alimentation au quotidien, d'accéder à des programmes sportifs adaptés, et de découvrir une variété de recettes saines pour maintenir un bon équilibre nutritionnel.

L'application intègre également un accompagnement par des coachs professionnels, qui assurent un suivi personnalisé et motivant. Grâce à des outils de suivi de progression, chaque utilisateur peut visualiser ses résultats, ajuster ses efforts, et rester engagé dans l'atteinte de ses objectifs. **NutriMotion** s'impose ainsi comme une solution complète pour adopter un mode de vie plus sain et actif.



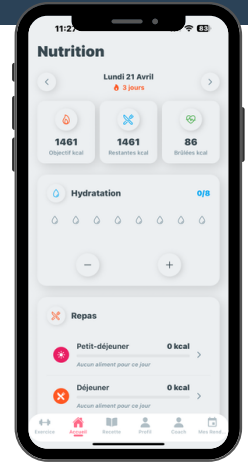
02

OBJECTIF DE L'APPLICATION

ALAIN RABIE

2. Objectif de l'application

NutriMotion a pour objectif d'accompagner les utilisateurs dans l'amélioration de leur hygiène de vie en combinant nutrition, sport et suivi personnalisé. Elle vise à encourager une alimentation équilibrée, une activité physique régulière, ainsi qu'un encadrement par des professionnels pour un accompagnement global et motivant.



1

Élaboration d'une stratégie de bien-être numérique.

Créer une plateforme simple et intuitive regroupant nutrition, sport et accompagnement pour centraliser tous les outils nécessaires à une bonne hygiène de vie.

2

Suivi personnalisé des utilisateurs

Permettre aux utilisateurs de suivre leurs calories, entraînements et évolution physique avec des outils adaptés à leurs objectifs (perte de poids, prise de masse, etc.).

3

Création de contenus adaptés

Proposer des recettes saines, des programmes d'entraînement variés et des conseils pratiques pour enrichir l'expérience et rester motivé au quotidien.

4

Mise en place d'un accompagnement humain

Offrir un suivi par des coaches professionnels pour conseiller, corriger et soutenir les utilisateurs dans leur progression.

03

ANALYSE FONCTIONNELLE ET CHOIX TECHNIQUE

ALAIN RABIE

3. Analyse Fonctionnelle et Choix Technique

L'application développée vise à offrir un suivi complet de la santé et du bien-être, avec un focus sur les calories, les programmes de sport, les recettes saines, et l'accompagnement par des coaches.

Authentification et gestion des utilisateurs

Écran de connexion pour tout les utilisateurs :

Permet aux utilisateurs de se connecter de manière sécurisée.

Écran de gestion de profil utilisateurs :

Offre la possibilité de visualiser et de modifier les informations du profil.

CRUD des utilisateurs :

Les administrateurs ont accès à l'ensemble des comptes utilisateurs. Ils peuvent créer un nouvel utilisateur, consulter les informations existantes, les modifier (comme le poids, la taille, ou les objectifs), et supprimer un compte si nécessaire.

Réinitialisation du mot de passe par mail :

En cas d'oubli de mot de passe, un administrateur peut envoyer un e-mail de réinitialisation à l'utilisateur concerné, lui permettant de choisir un nouveau mot de passe en toute sécurité.

CRUD des coaches :

Les administrateurs peuvent gérer les coaches présents sur l'application. Ils peuvent en ajouter de nouveaux, modifier leurs informations (nom, spécialité, description), consulter leur profil, ou les retirer de la plateforme si besoin.

Recettes :

Les utilisateurs ont accès à une bibliothèque de recettes variées, équilibrées et adaptées à différents objectifs (perte de poids, prise de masse, maintien, etc.). Chaque recette affiche les ingrédients nécessaires, les étapes de préparation, ainsi que le nombre de calories estimé par portion.

Des suggestions de recettes peuvent également être proposées en fonction des préférences de l'utilisateur ou de ses objectifs nutritionnels.

Gestion des Calories :

Consultation de l'historique des calories :

Permet aux utilisateurs de consulter l'historique de leurs calories consommées à partir de n'importe quelle date et d'accéder aux détails de leur alimentation passée.

Visualisation des calories du jour en cours :

Offre la possibilité de consulter les calories consommées et l'hydratation de la journée en cours, avec un récapitulatif des repas et des aliments ajoutés.

Ajout d'aliment prédéfini :

Les utilisateurs peuvent intégrer des aliments prédéfinis à leur suivi, en sélectionnant parmi une liste d'aliments courants ou enregistrés dans la base de données de l'application.

Ajout d'aliment personnalisé :

Permet aux utilisateurs d'ajouter des aliments personnalisés avec leurs calories, apport en nutriments, et toute autre information pertinente, pour un suivi précis de leur nutrition.

Gestion des coaches :

Consultation des coaches :

Les utilisateurs peuvent accéder à une liste de coaches disponibles sur la plateforme.

Pour chaque coach, les informations suivantes sont affichées : nom, spécialité (musculature, cardio, perte de poids, etc.), prix par séance, disponibilité (créneaux horaires), ainsi que les moyens de contact (adresse e-mail et numéro de téléphone). L'utilisateur peut ainsi directement contacter un coach pour prendre rendez-vous ou poser des questions via mail ou téléphone.

04

LANGAGE ET TECHNOLOGIES UTILISÉES

ALAIN RABIE

4. Langage et technologies utilisées

React - native

Ce choix m'a permis de gagner du temps dans le développement tout en assurant une expérience utilisateur fluide et moderne sur différents appareils.

React Native offre également un large écosystème de bibliothèques, une communauté très active et des outils puissants pour le debug, la navigation ou encore la gestion d'état.

FireBase

L'un des grands avantages de Firebase est qu'il est entièrement compatible avec React Native et Expo, ce qui m'a permis de l'intégrer facilement à mon projet sans trop de configuration. De plus, il permet un hébergement cloud sécurisé, avec une scalabilité automatique, ce qui est idéal même pour un projet évolutif.

Expo

J'ai choisi d'utiliser Expo avec React Native car c'est une plateforme qui simplifie énormément le développement d'applications mobiles. Expo propose un ensemble d'outils et de services qui permettent de lancer, tester et déployer une application rapidement, sans avoir à configurer manuellement Android Studio ou Xcode.

OpenMealDB

Grâce à OpenMealDB, j'ai pu enrichir l'application avec des recettes équilibrées, adaptées aux objectifs des utilisateurs (perte de poids, prise de muscle, etc.), sans avoir à construire cette base de données moi-même. L'API offre également une recherche par ingrédient ou catégorie de recette, ce qui facilite la navigation et la personnalisation de l'expérience utilisateur.



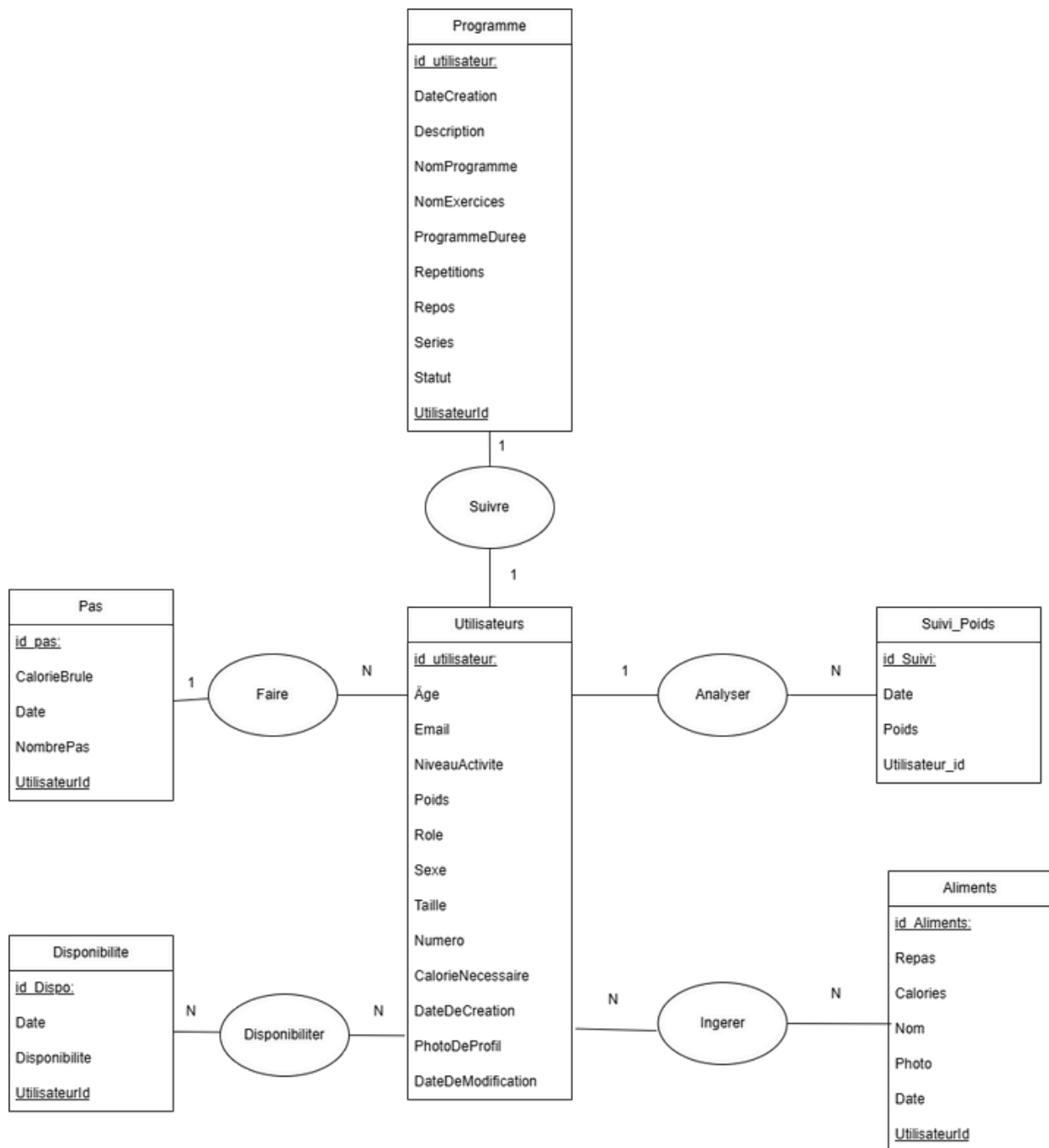
05

MA BASE DE DONNÉE FIREBASE

ALAIN RABIE

5. Base de donnée MCD

VOICI LES COLLECTIONS IMAGÉE DANS MON MCD



6. Relation MCD

Utilisateur :

Association entre les entités Utilisateurs et Programme

Cette relation traduit le fait qu'un utilisateur peut être inscrit à un ou plusieurs programmes personnalisés, conçus notamment par des coaches. Un programme peut, à son tour, être suivi par plusieurs utilisateurs.

Clés utilisées :

utilisateurId dans la table Programme

Potentiellement une table associative pour modéliser une relation plusieurs-à-plusieurs si nécessaire.

Cardinalité :

Un utilisateur peut suivre 0 à n programmes ; un programme peut être suivi par 0 à n utilisateurs.

Association entre les entités Utilisateurs et Pas

Elle permet de représenter l'activité physique journalière d'un utilisateur, en particulier le nombre de pas effectués sur une période donnée. Chaque enregistrement de pas est rattaché à un utilisateur unique.

Clé utilisée :

utilisateurId dans la table Pas

Cardinalité :

Un utilisateur peut avoir réalisé 0 à n enregistrements de pas.
(1:n)

Association entre les entités Utilisateurs et Suivi_Poids

Cette relation illustre le suivi du poids corporel par l'utilisateur. Chaque mesure (poids, date) est liée à un seul utilisateur, mais un utilisateur peut disposer de plusieurs enregistrements au fil du temps.

Clé utilisée :

utilisateurId dans la table Suivi_Poids

Cardinalité :

Un utilisateur peut disposer de 0 à n enregistrements de poids.
(1:n)

7. Relation MCD

Utilisateur :

Association entre les entités Utilisateurs et Aliments

Cette relation représente la consommation alimentaire d'un utilisateur. Chaque aliment consommé est lié à l'utilisateur concerné, avec des informations comme les calories, la date, etc.

Clé utilisée :
utilisateurId dans la table Aliments

Cardinalité :
Un utilisateur peut consommer 0 à n aliments.
(1:n)

Association entre les entités Utilisateurs et Disponibilite

Cette relation met en évidence les plages de disponibilité d'un utilisateur, notamment dans le cas des coaches. Elle permet de planifier des séances ou des suivis personnalisés.

Clé utilisée :
utilisateurId dans la table utilisateur

Cardinalité :
Un utilisateur (coach) peut avoir 0 à n disponibilités enregistrées.
(1:n)

06


ORGANISATION DU CODE

ALAIN RABIE

8. Vue

Dans le cadre du développement de cette application mobile avec **React Native** et **Expo**, j'ai structuré mon projet en centralisant tous les éléments dans le dossier `app/`, conformément aux conventions imposées par **Expo Router**.

En effet, **Expo Router** repose sur une logique de navigation basée sur le système de fichiers, ce qui signifie que chaque fichier ou dossier situé dans `app/` correspond automatiquement à une route ou un écran de l'application. Cette approche simplifie la gestion de la navigation, tout en rendant la structure du projet intuitive et cohérente.



```

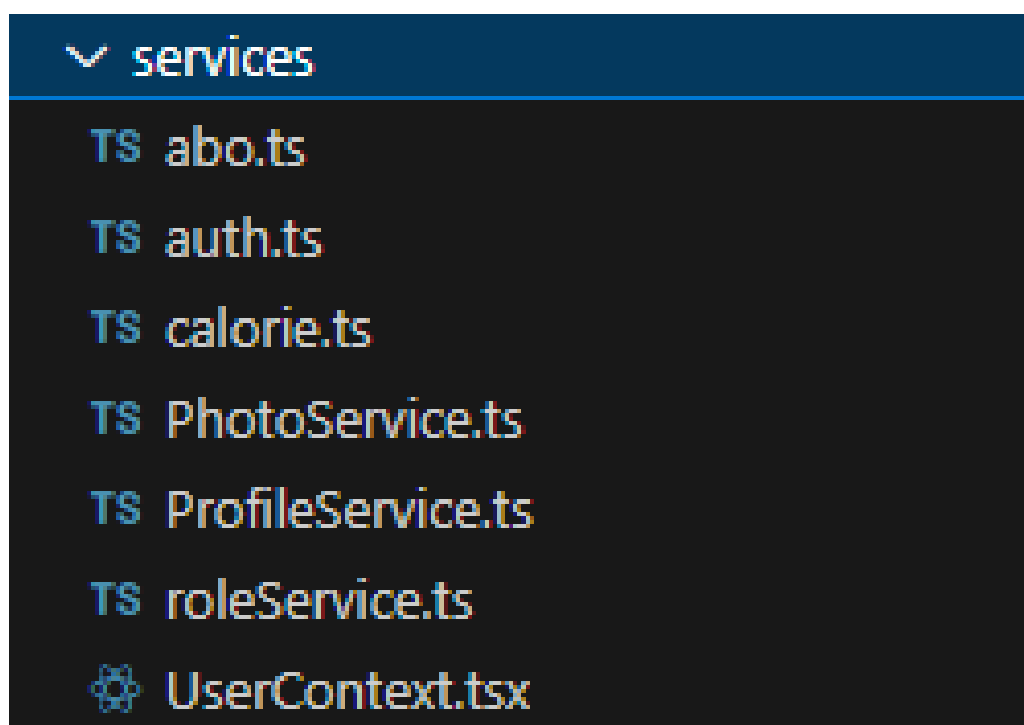
  app
  > (admin)
  > (auth)
  > (calorie)
  > (client)
  > (coach)
  > (exo)
  > (premium)
  > (reglage)
  > (tabs)
  _layout.tsx
  index.tsx

```

Les pages (ou écrans) sont donc placées directement dans le dossier `app/`, ce qui permet à **Expo Router** de les reconnaître et de générer automatiquement les routes correspondantes. Par ailleurs, l'utilisation de parenthèses dans certains noms de dossiers (par exemple `(tabs)` ou `(auth)`) est une fonctionnalité propre à **Expo Router**, qui permet de définir des groupes de routes sans affecter l'URL réelle. Cela est particulièrement utile pour organiser la navigation en groupes logiques (comme les onglets, les routes privées ou publiques) tout en gardant une arborescence claire

9. Services

Le dossier `services/` a été introduit dans l'architecture de l'application afin de centraliser toute la logique métier liée aux interactions externes ou aux fonctionnalités transversales. Dans le contexte de **React Native** avec **Expo**, ce répertoire regroupe notamment les fichiers responsables de l'accès aux APIs REST, à des SDK natifs (comme le podomètre via `expo-sensors`), ou encore aux services d'authentification. Cette séparation permet d'isoler la logique fonctionnelle du code d'interface (écrans, composants), conformément aux bonnes pratiques de séparation des responsabilités.



Les pages (ou écrans) sont donc placées directement dans le dossier `app/`, ce qui permet à **Expo Router** de les reconnaître et de générer automatiquement les routes correspondantes. Par ailleurs, l'utilisation de parenthèses dans certains noms de dossiers (par exemple `(tabs)` ou `(auth)`) est une fonctionnalité propre à **Expo Router**, qui permet de définir des groupes de routes sans affecter l'URL réelle. Cela est particulièrement utile pour organiser la navigation en groupes logiques (comme les onglets, les routes privées ou publiques) tout en gardant une arborescence claire

07

PRESENTATION DU CODE

ALAIN RABIE

10. Firebase

Ce fichier joue un rôle central dans l'intégration de **Firebase** au sein de l'application. Il contient la configuration de base permettant de connecter l'application mobile à un projet **Firebase**, en utilisant les clés et identifiants fournis par la console **Firebase**.

Concrètement, **firebaseConfig.ts** exporte une constante de type objet contenant les éléments suivants :

- **apiKey** : Clé publique pour autoriser l'accès à **Firebase**.
- **authDomain** : Domaine d'authentification (souvent utilisé pour le web, mais nécessaire à l'init).
- **projectId** : Identifiant unique du projet Firebase.
- **storageBucket** : Utilisé pour le stockage de fichiers dans Firebase Storage.
- **messagingSenderId** : ID nécessaire pour le cloud messaging (notifications).
- **appId** : Identifiant unique de l'application.

TS *firebaseConfig.ts* X

Ap4 > firebase > TS *firebaseConfig.ts* > [🔍] default

```
1  import { initializeApp } from "firebase/app";
2  import { getFirestore } from "firebase/firestore";
3  import { getAuth } from "firebase/auth";
4
5  const firebaseConfig = {
6    apiKey: "AIzaSyD0tG1_tkj8QxwcUmKi6twRfz0lhIa3euo",
7    authDomain: "callapp-e73e5.firebaseio.com",
8    projectId: "callapp-e73e5",
9    storageBucket: "callapp-e73e5.firebaseio.com",
10   messagingSenderId: "76884156975",
11   appId: "1:76884156975:web:7b322a890d1c935cf49cda"
12 };
13
14 const app = initializeApp(firebaseConfig);
15 export const auth = getAuth(app);
16 export const db = getFirestore(app);
17 export default app;
```

11. Login

La fonction **login** permet d'authentifier un utilisateur en utilisant son email et son mot de passe via **Firestore Authentication**.

Lorsque la connexion est réussie, elle récupère les informations de l'utilisateur à partir de **Firestore** en accédant à son document correspondant grâce à son uid. Si ce document existe, la fonction met à jour la date de la dernière connexion de l'utilisateur, en utilisant l'option { merge: true } pour ne pas écraser les données existantes.

Cela permet de conserver toutes les informations de l'utilisateur tout en ajoutant la nouvelle date de connexion.

Les informations de l'utilisateur authentifié et ses données personnelles sont ensuite renvoyées pour être utilisées dans l'application.

En cas d'échec, une erreur est capturée et un message d'erreur explicite est renvoyé, permettant de diagnostiquer rapidement le problème. Cette approche garantit une gestion fluide de l'authentification et des données utilisateur au sein de l'application.

```
export const login = async (email: string, password: string) => {  
  try {  
    const userCredential = await signInWithEmailAndPassword(auth, email, password);  
    const user = userCredential.user;  
    console.log('Utilisateur connecté avec succès :', user);  
  
    const userDocRef = doc(db, 'utilisateurs', user.uid);  
    const userDoc = await getDoc(userDocRef);  
  
    if (!userDoc.exists()) {  
      throw new Error("Données utilisateur introuvables");  
    }  
  
    const userData = userDoc.data();  
  
    await setDoc(userDocRef, { derniereConnexion: new Date() }, { merge: true });  
  
    return {  
      authUser: user,  
      userData: userData  
    };  
  } catch (error: any) {  
    console.error('Erreur lors de la connexion :', error.message);  
    throw new Error(error.message);  
  }  
};
```

12. Register

La fonction `register` permet d'enregistrer un nouvel utilisateur en utilisant son email et son mot de passe via **Firestore Authentication**.

Une fois l'utilisateur créé, la fonction récupère l'objet utilisateur renvoyé par **Firestore**.

Ensuite, elle crée un document dans la collection utilisateurs de **Firestore** en utilisant l'uid de l'utilisateur comme identifiant unique.

Ce document contient plusieurs informations importantes, telles que l'email, le nom complet, un rôle par défaut ("utilisateur"), la date de création du compte, la dernière connexion et un champ pour l'avatar qui est initialisé à une chaîne vide.

Ces informations permettent de personnaliser l'expérience de l'utilisateur et de gérer son compte.

En cas d'erreur lors de l'inscription, celle-ci est capturée et un message d'erreur est renvoyé pour faciliter le diagnostic. Ainsi, cette fonction gère à la fois l'inscription de l'utilisateur dans **Firestore Authentication** et l'enregistrement de ses informations dans **Firestore**, assurant une intégration complète des données utilisateur.

```
export const register = async (email: string, password: string, nomComplet: string, departement: string) => {
  try {
    const userCredential = await createUserWithEmailAndPassword(auth, email, password);
    const user = userCredential.user;

    console.log('Utilisateur inscrit avec succès :', user);

    await setDoc(doc(db, 'utilisateurs', user.uid), {
      id: user.uid,
      email: user.email,
      nomComplet: nomComplet,
      role: 'utilisateur',
      dateCreation: new Date(),
      derniereConnexion: new Date(),
      urlAvatar: '',
    });

    return user;
  } catch (error: any) {
    console.error('Erreur lors de l\'inscription :', error.message);
    throw new Error(error.message);
  }
};
```

13. Ajouter

La fonction **addMeal** permet d'ajouter un repas dans la base de données **Firestore** pour un utilisateur connecté.

Elle commence par vérifier si un utilisateur est authentifié via **Firestore Authentication**.

Si ce n'est pas le cas, elle bloque l'ajout et renvoie une erreur. Ensuite, elle construit un objet **newMeal** à partir des données reçues, incluant le nom du repas, les calories, le type de repas (comme petit-déjeuner ou dîner), l'URL de la photo, l'identifiant de l'utilisateur et la date du repas au format ISO.

Une fois ces informations prêtes, elles sont envoyées dans la collection aliments de **Firestore** à l'aide de la fonction **addDoc**.

Ce système permet à chaque utilisateur de suivre ses repas au quotidien de manière personnalisée et sécurisée. En cas de problème, une erreur est capturée et un message explicite est renvoyé.

```
export const addMeal = async (mealData: MealData): Promise<void> => {
  const user = auth.currentUser;

  if (!user) {
    throw new Error("Vous devez être connecté pour ajouter un repas.");
  }

  try {
    const newMeal = {
      nom: mealData.nom,
      calories: mealData.calories,
      Repas: mealData.Repas,
      urlPhoto: mealData.urlPhoto,
      utilisateurId: user.uid,
      date: mealData.date.toISOString(),
    };

    await addDoc(collection(db, "aliments"), newMeal);
  } catch (error) {
    console.error("Erreur lors de l'ajout du repas :", error);
    throw new Error("Impossible d'ajouter le repas.");
  }
}
```

14. Podomètre

Dans mon application de gestion de calories, j'ai intégré un podomètre afin de permettre aux utilisateurs de suivre leur activité physique quotidienne. Pour cela, j'ai utilisé le module **Pedometer** de la bibliothèque **expo-sensors** fournie par **Expo**.

Grâce à ce module, l'application vérifie d'abord si le podomètre est disponible sur l'appareil via la méthode **Pedometer.isAvailableAsync()**. Si c'est le cas, elle récupère ensuite le nombre total de pas effectués depuis le début de la journée jusqu'à l'instant actuel à l'aide de **Pedometer.getStepCountAsync(start, end)**. En parallèle, l'application met en place une surveillance en temps réel des pas grâce à **Pedometer.watchStepCount()**, qui met automatiquement à jour l'état avec le nombre de nouveaux pas effectués. L'ensemble de cette logique est géré dans un **useEffect** pour que le suivi commence dès que le composant est monté. Cette fonctionnalité enrichit l'expérience utilisateur en fournissant des données en temps réel sur leur activité, ce qui permet d'adapter les conseils nutritionnels et sportifs proposés par l'application.

```
useEffect(() => {
  const startPedometer = async () => {
    try {
      const available = await Pedometer.isAvailableAsync();
      setIsPedometerAvailable(available);

      if (available) {
        const end = new Date();
        const start = new Date();
        start.setHours(0, 0, 0, 0);

        try {
          const pastStepData = await Pedometer.getStepCountAsync(start, end);
          if (pastStepData) {
            setSteps(pastStepData.steps);
          }

          const subscription = Pedometer.watchStepCount(result => {
            setSteps(currentSteps => currentSteps + result.steps);
          });

          return () => {
            if (subscription) {
              subscription.remove();
            }
          };
        } catch (err) {
          console.error("Erreur lors de l'accès au podomètre:", err);
        }
      } else {
        console.log("Le podomètre n'est pas disponible sur cet appareil");
      }
    } catch (error) {
      console.error("Erreur d'initialisation du podomètre:", error);
    }
  };

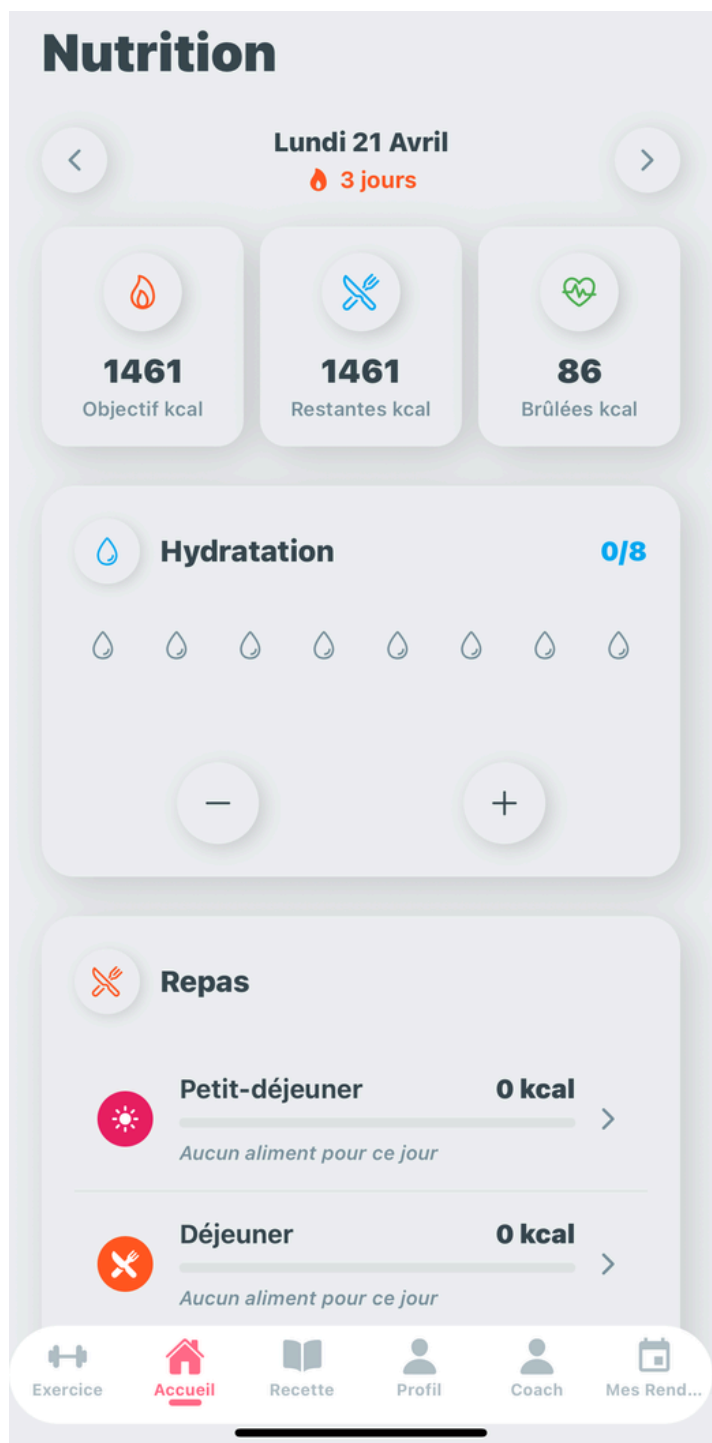
  startPedometer();
}, []);
```


08

PRESENTATION DE L'APPLICATION (RÔLE UTILISATEUR)

ALAIN RABIE

15. Accueil



1

Résumé calorique

Cette section affiche l'objectif calorique quotidien, les calories restantes, et celles brûlées. Les calories brûlées sont mises à jour en temps réel grâce au podomètre intégré. Elle permet à l'utilisateur de visualiser rapidement son équilibre énergétique.

2

Hydratation

L'utilisateur peut suivre sa consommation d'eau quotidienne via un compteur visuel. Chaque goutte représente un verre, avec un objectif fixé à 8 verres par jour. Des boutons permettent d'ajouter ou retirer facilement les verres consommés.

3

Suivi des repas

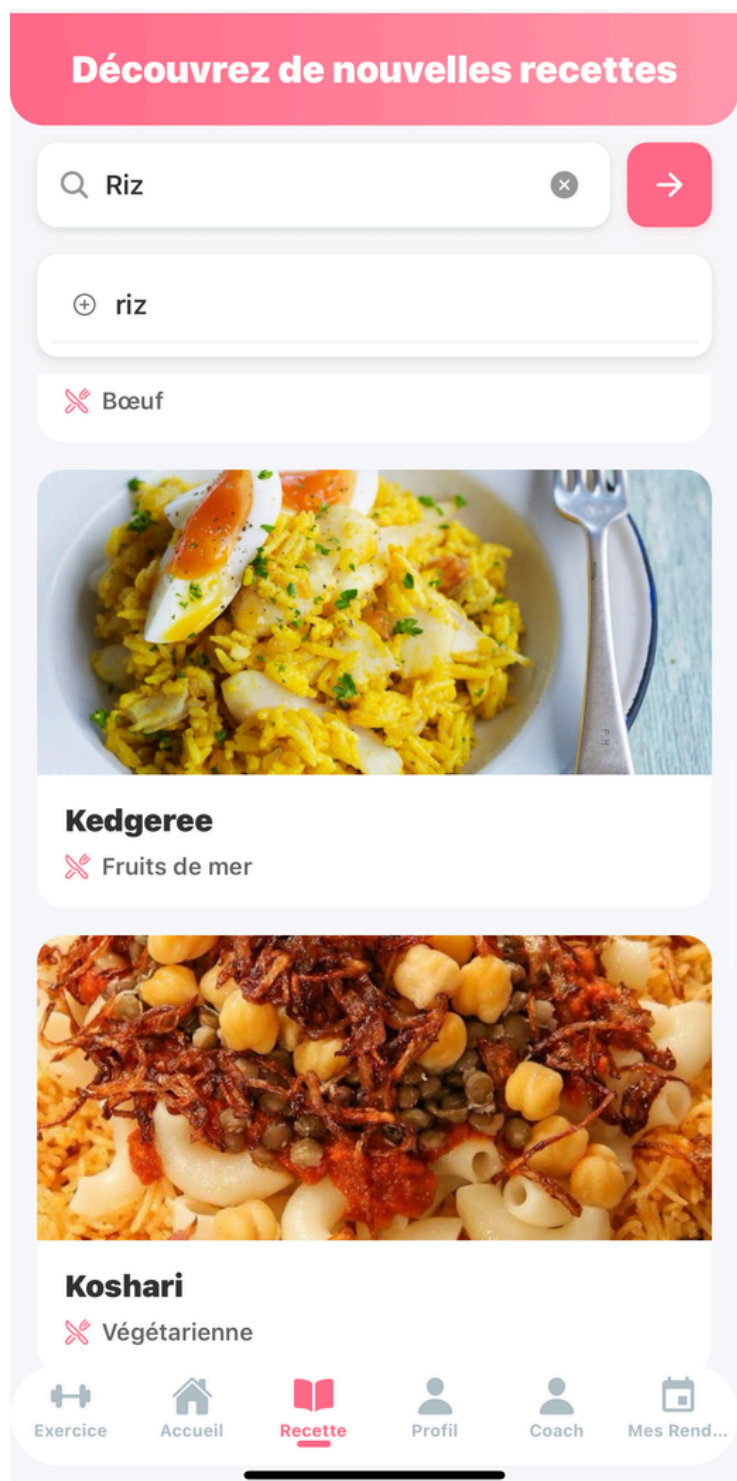
Proposer des recettes saines, des programmes d'entraînement variés et des conseils pratiques pour enrichir l'expérience et rester motivé au quotidien.

4

Barre de navigation

La barre de navigation permet d'accéder aux différentes sections de l'application. Elle comprend des onglets comme Exercice, Recette, Coach, et Profil. Elle facilite la navigation rapide entre les fonctionnalités principales.

16. Recette



1

Recherche de recettes

L'utilisateur peut rechercher des recettes en tapant un ingrédient ou un mot-clé.

Un champ de recherche avec un bouton permet de lancer la requête facilement.

Cela permet de filtrer les recettes en fonction des préférences alimentaires.

2

Résultat de recherche

Les résultats s'affichent sous forme de cartes avec une image, un nom et une catégorie.

On retrouve ici des recettes contenant "riz", comme "Poulet Congee" ou "Fatteh".

Cela permet de trouver rapidement des idées de plats à base d'un ingrédient.

3

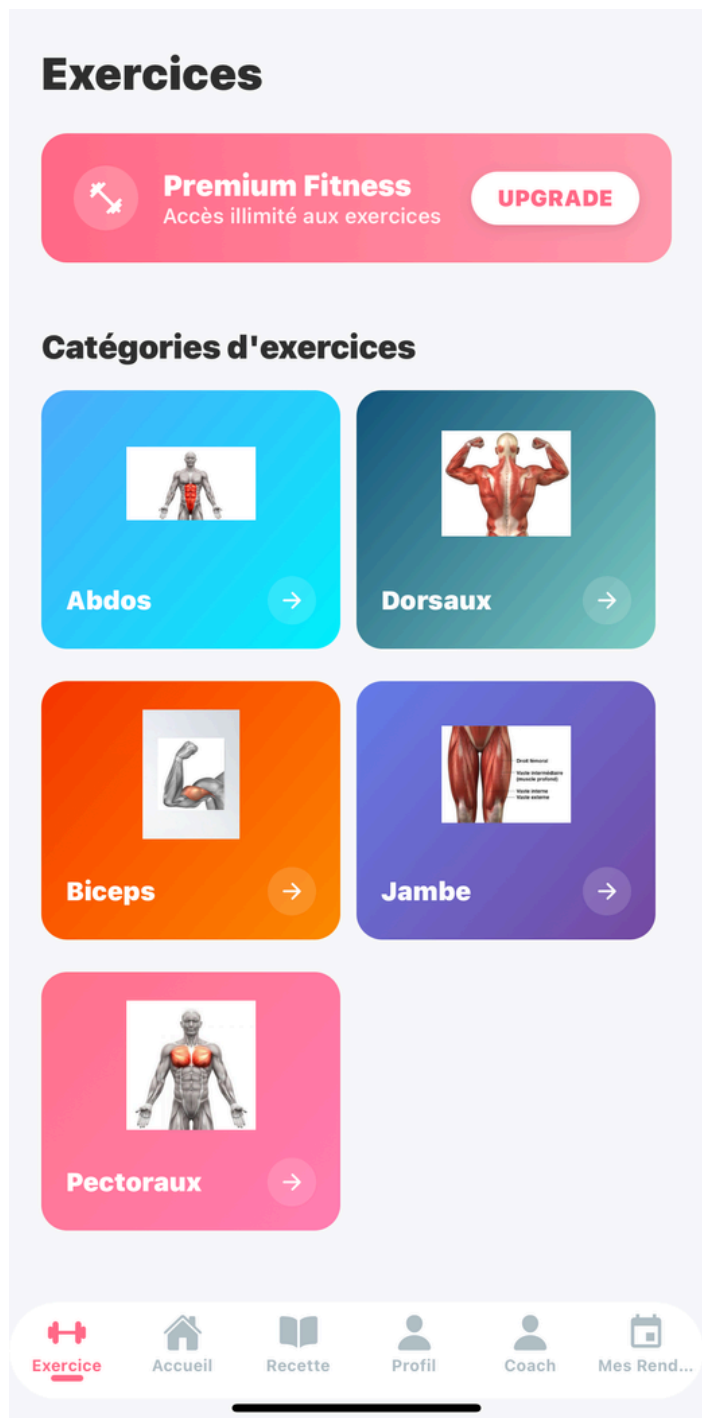
Affichage des recettes

Chaque recette est présentée avec une photo appétissante et son nom.

Les noms comportent des caractères encodés (%20) qu'il serait utile de nettoyer.

Chaque carte est cliquable pour accéder aux détails de la recette.

17. Exercices



1

Organisation des exercices

Sous la bannière, le titre "Catégories d'exercices" introduit le cœur de la page. Les exercices sont répartis par groupe musculaire : abdos, dorsaux, biceps, jambes, pectoraux, etc. Chaque catégorie est affichée dans un carré coloré, avec une image anatomique du muscle correspondant, ce qui rend l'interface intuitive.

2

Navigation vers les exercices

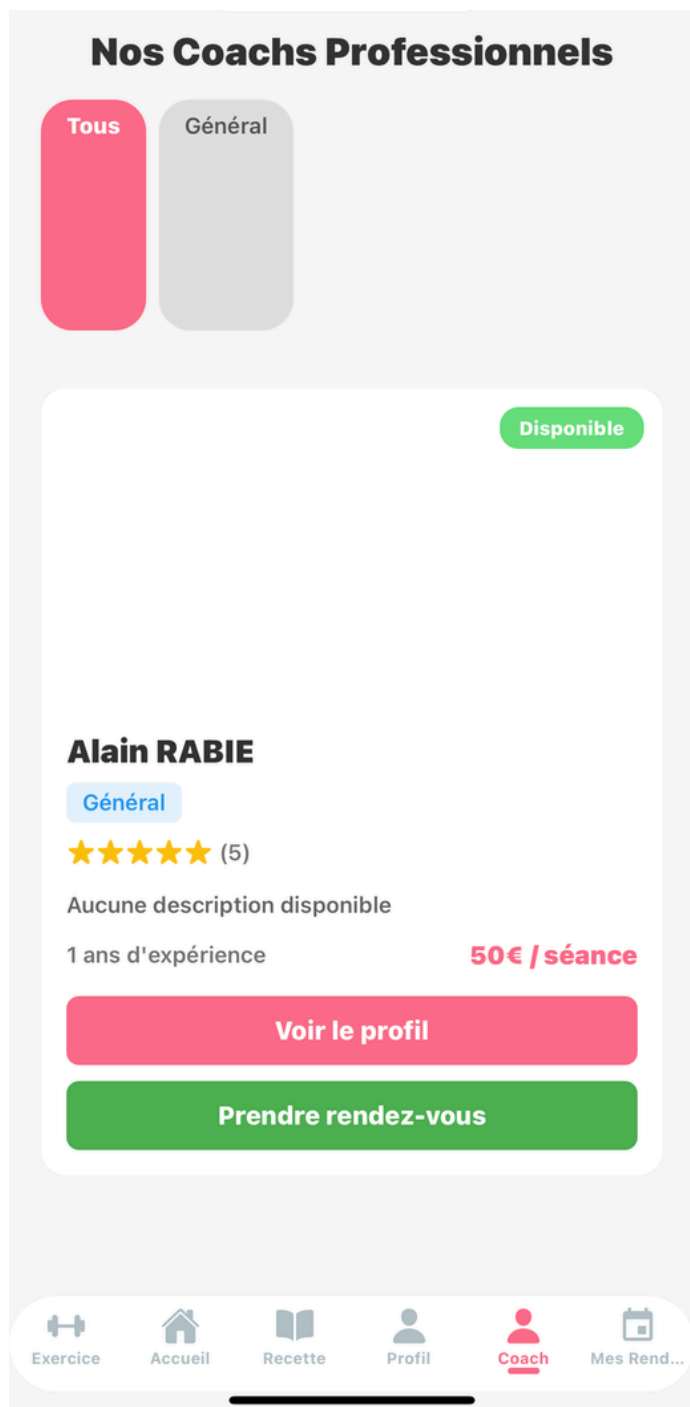
En cliquant sur une des cartes (ex : "Abdos"), l'utilisateur est redirigé vers une page dédiée listant plusieurs exercices ciblant ce muscle. Chaque exercice peut inclure des explications, des vidéos ou images, et des variantes selon le niveau ou le matériel disponible.

3

Affichage des exercices

La page d'exercices permet à l'utilisateur de choisir des entraînements selon différentes parties du corps comme les abdos, les jambes ou les pectoraux. Chaque catégorie est illustrée pour faciliter la compréhension. Elle aide à structurer les séances et à accompagner le suivi calorique de manière cohérente.

18. Coach



1

Profil détaillé du coach

Cette page met en avant les coachs disponibles, affichant leur nom, leur catégorie, leur expérience et leurs tarifs. Les utilisateurs peuvent ainsi consulter les avis des autres membres à travers le système de notation par étoiles, leur permettant de choisir le professionnel qui correspond le mieux à leurs attentes.

2

Vérification de la disponibilité

En haut de la page, le statut du coach indique s'il est disponible pour des sessions. Cette information permet aux utilisateurs de savoir immédiatement s'ils peuvent réserver un rendez-vous ou s'ils doivent choisir un autre créneau horaire.

3

Accès rapide aux options

Deux boutons facilitent l'interaction avec le coach : « Voir le profil » permet d'accéder aux détails complets du coach, incluant ses tarifs et ses évaluations, tandis que « Prendre rendez-vous » redirige directement vers la page de planification des séances. Ces fonctionnalités garantissent une expérience fluide et intuitive pour les utilisateurs.

19. Profil



1

Profil Utilisateur Personnalisé

Sur cet écran, l'utilisateur peut consulter ses informations personnelles telles que son nom complet, son email, son poids, sa taille et son âge. Toutes ces informations sont présentées de manière claire et intuitive. Le bouton Déconnexion est également disponible pour permettre à l'utilisateur de se déconnecter rapidement et en toute sécurité de l'application.

2

Suivi de la Forme Physique

L'utilisateur retrouve ici son Indice de Masse Corporelle (IMC) automatiquement calculé à partir de ses données de poids et de taille. Cet indice est affiché avec un diagnostic (par exemple ici "Insuffisance pondérale") et une visualisation sur une échelle colorée, pour lui permettre de comprendre instantanément son état de santé physique. Cette fonction donne une première indication sur la nécessité éventuelle d'adapter son mode de vie.

3

Gestion des Données Personnelles

Depuis cet écran, l'utilisateur peut modifier son poids, sa taille et son âge à tout moment en cliquant sur les sections correspondantes. À partir de ces données mises à jour, l'application recalcule automatiquement les besoins caloriques journaliers de l'utilisateur, qui s'affichent en bas de l'écran. Cela garantit un suivi précis et personnalisé, en fonction de son évolution physique et de ses objectifs.

20. Reglage



1

Suivi quotidien des pas

Le podomètre permet aux utilisateurs de suivre leur activité physique au jour le jour. Grâce à un affichage clair et précis, ils peuvent voir le nombre de pas effectués et ainsi ajuster leur routine en fonction de leurs objectifs de santé et de bien-être.

2

Une opportunité pour les professionnels

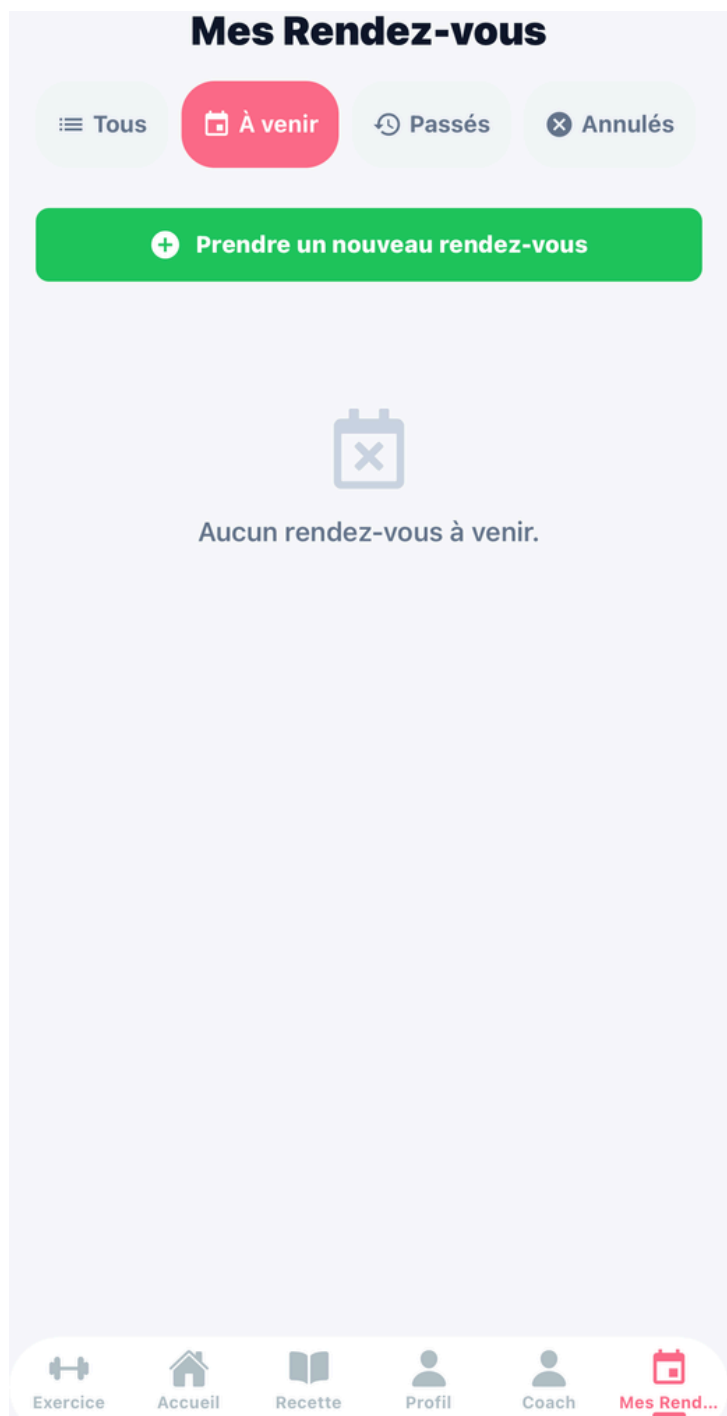
La fonctionnalité « Devenir Coach » offre aux experts en nutrition et en fitness la possibilité de rejoindre la plateforme. En soumettant leur candidature, ils peuvent partager leurs connaissances et aider les utilisateurs à atteindre leurs objectifs de manière efficace et personnalisée.

3

Processus de validation simplifié

Lorsqu'un utilisateur fait une demande pour devenir coach, un e-mail est envoyé à un administrateur avec son ID et sa requête. Après validation, son statut est mis à jour, lui permettant d'accéder aux outils dédiés aux coaches et de proposer ses services en toute liberté.

21. Rendez-vous



1

Suivi des rendez-vous

Cette page permet aux utilisateurs de consulter l'ensemble de leurs rendez-vous, classés en différentes catégories : à venir, passés et annulés. Avec cette organisation claire, chaque utilisateur peut rapidement visualiser son planning et gérer ses engagements en toute simplicité.

2

Gestion intuitive

Lorsqu'un utilisateur n'a aucun rendez-vous prévu, l'application affiche un message explicite et offre une solution immédiate grâce à un bouton dédié. Ce dernier permet de prendre un nouveau rendez-vous en quelques instants, garantissant une expérience fluide et accessible.

3

Accès rapide à la planification

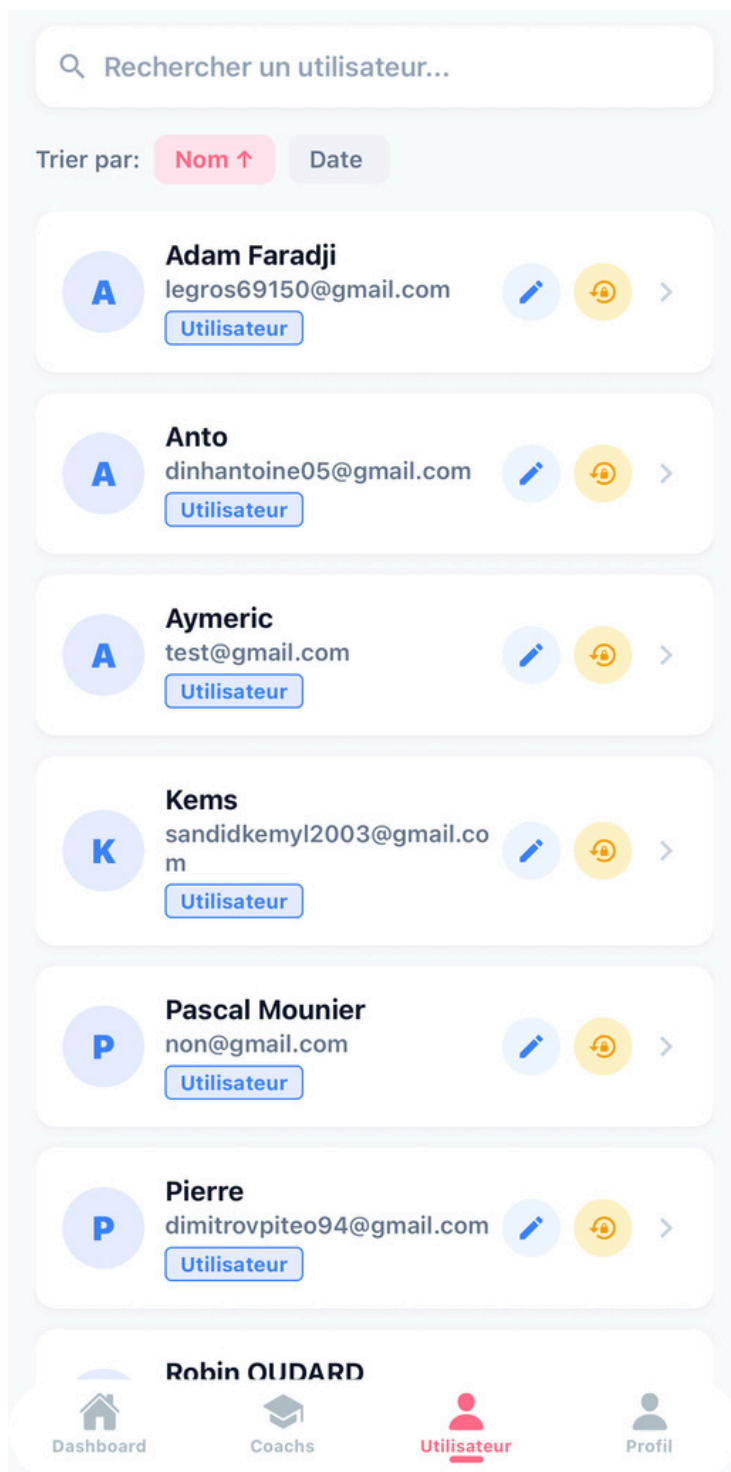
Le bouton « Prendre un nouveau rendez-vous » redirige directement vers la page de réservation, facilitant ainsi la prise de rendez-vous avec un coach. Grâce à cette fonctionnalité, les utilisateurs peuvent organiser leurs séances en toute simplicité et bénéficier d'un accompagnement personnalisé.

08

PRESENTATION DE L'APPLICATION (RÔLE ADMINISTRATEUR)

ALAIN RABIE

22. Utilisateur



1

Gestion des Utilisateurs

Dans cet écran, tous les utilisateurs de la plateforme sont listés de manière claire et organisée. Un champ de recherche en haut de la page permet de retrouver rapidement un utilisateur en saisissant son nom ou son email. Il est également possible de trier les utilisateurs soit par Nom soit par Date d'inscription, facilitant ainsi la gestion et la navigation dans une grande base d'utilisateurs.

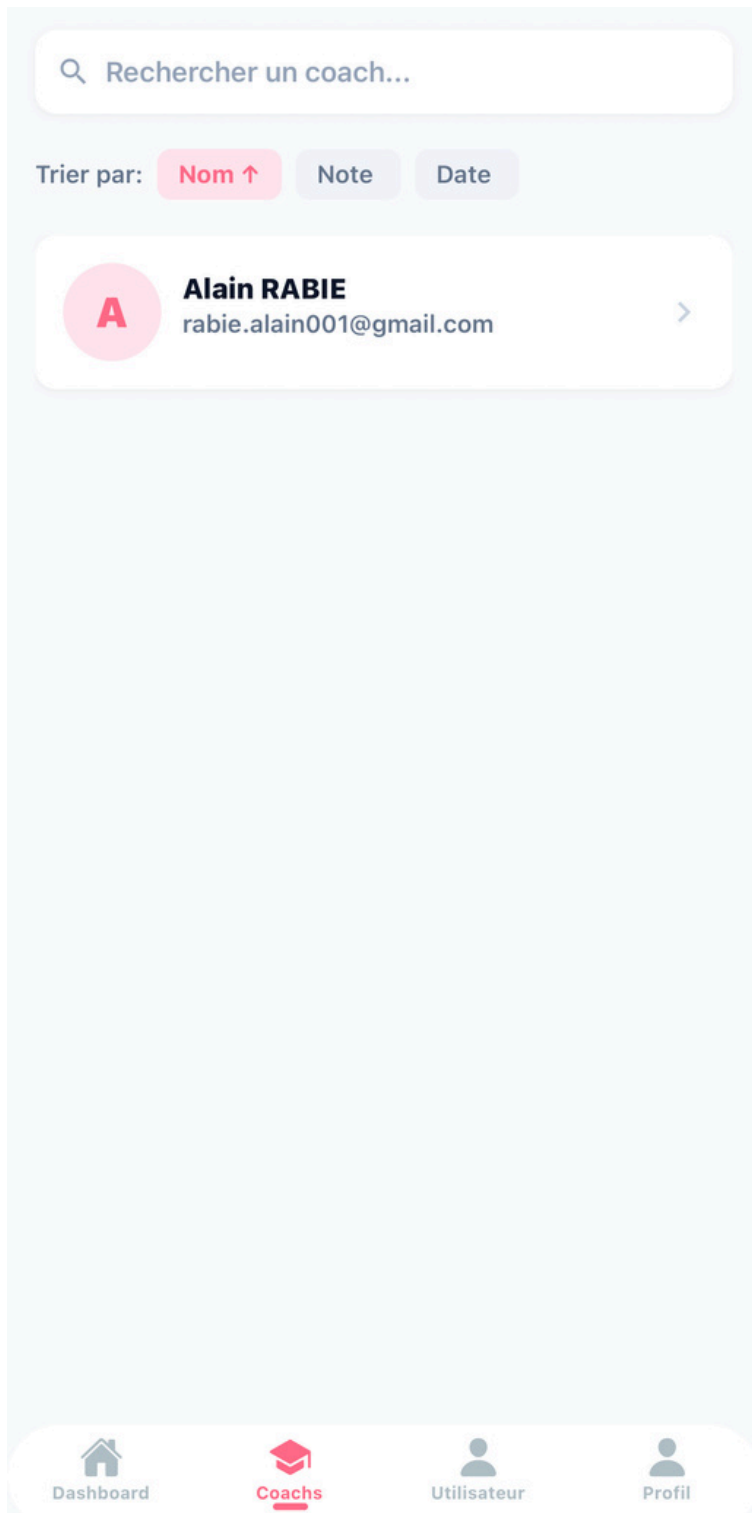
2

Actions Rapides sur les Utilisateurs

Chaque utilisateur possède deux actions principales à droite de son encadré : un bouton en forme de stylo pour modifier ses informations (nom, email, statut, etc.), et un bouton jaune représentant une flèche circulaire, qui sert à envoyer directement un mail de réinitialisation de mot de passe.

Cette organisation permet d'agir rapidement en cas de besoin de mise à jour ou d'assistance d'un utilisateur sans passer par plusieurs menus.

23. Coach



1

Planning

Une personne ayant le rôle utilisateur peut consulter le profil d'un coach et pouvoir prendre rendez-vous à l'aide d'un système de créneaux horaires taxés à un prix que le coach aura choisi. L'utilisateur pourra également consulter les rendez-vous prises et pour quelle coach.

2

Disponibilité

Une personne ayant le rôle coach aura une rubrique dédiée pour afficher les disponibilités ainsi que son tarif horaire.

Une fois un rendez-vous pris par un utilisateur, le rendez-vous s'affiche dans la rubrique rendez-vous dédiée et une fois la date du créneau horaire passée, l'application attribue l'état du rendez-vous comme terminé.

3

État des rendez-vous

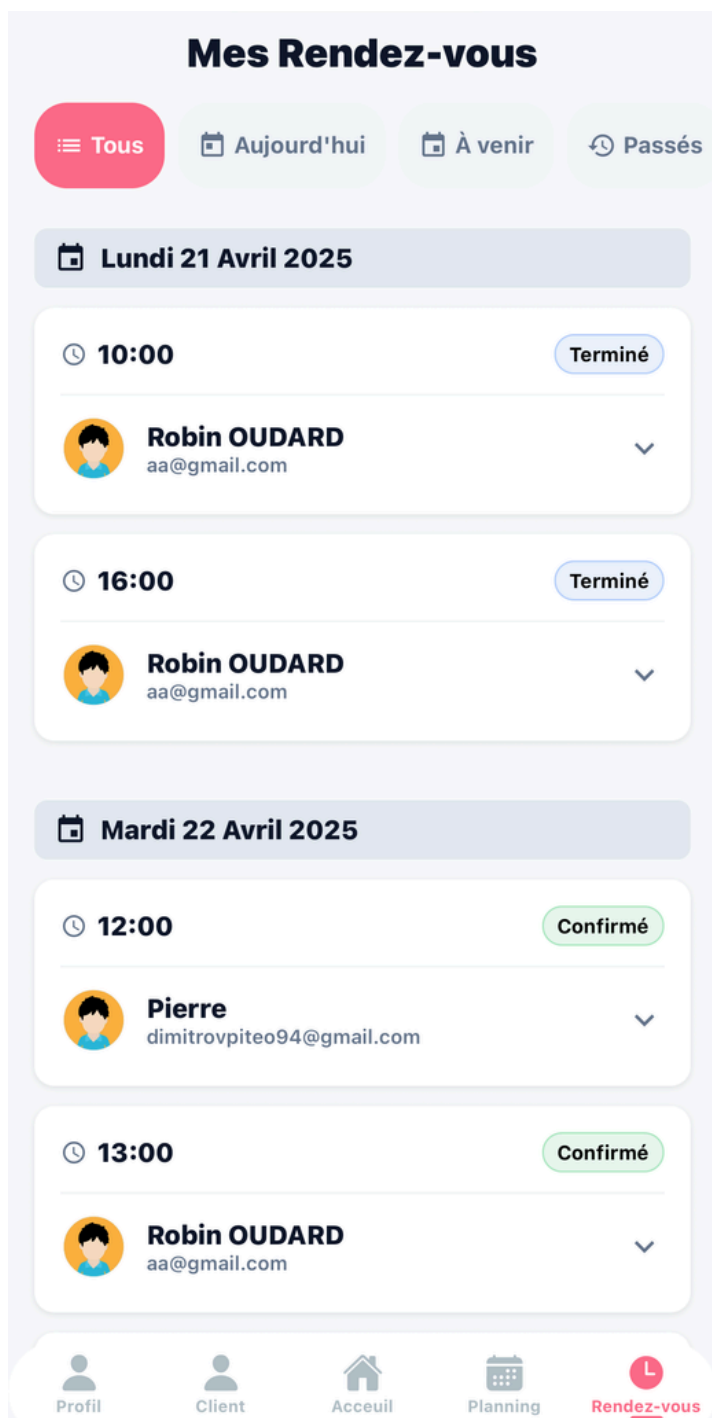
Le coach peut annuler un rendez-vous qui aura un impact sur les créneaux disponibles.

09

PRESENTATION DE L'APPLICATION (RÔLE COACH)

ALAIN RABIE

24. Rendez-vous



Coach

Une personne ayant le rôle utilisateur peut consulter le profil d'un coach et pouvoir prendre rendez-vous à l'aide d'un système de créneaux horaires taxés à un prix que le coach aura choisi. L'utilisateur pourra également consulter les rendez-vous pris et pour quel coach.

Disponibilité

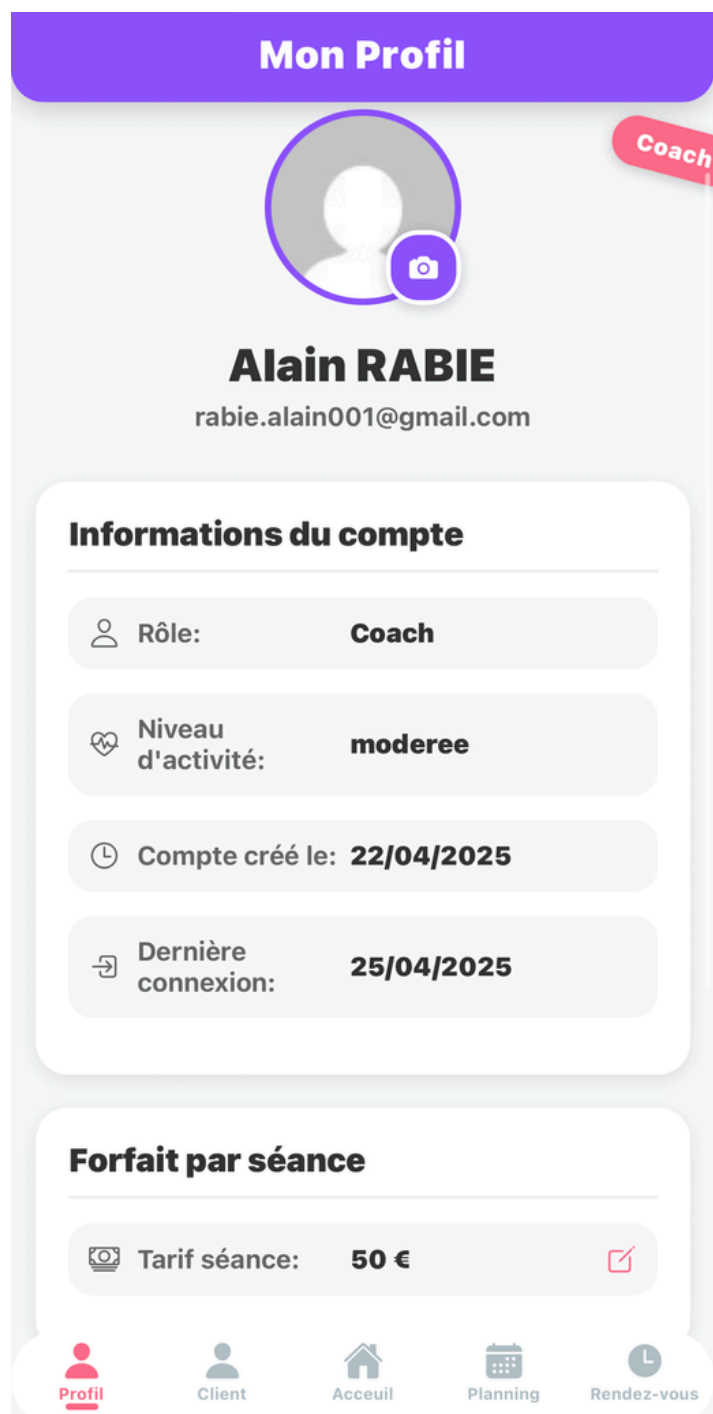
Une personne ayant le rôle coach aura une rubrique dédiée pour afficher les disponibilités ainsi que son tarif horaire.

Une fois un rendez-vous pris par un utilisateur, le rendez-vous s'affiche dans la rubrique rendez-vous dédiée et une fois la date du créneau horaire passée, l'application attribue l'état du rendez-vous comme terminé.

État des rendez-vous

Le coach peut annuler un rendez-vous qui aura un impact sur les créneaux disponibles.

25. Profil



1

Personnalisation du profil

Chaque coach peut modifier son nom et adapter ses tarifs directement depuis son espace personnel. Grâce à une interface intuitive, ces ajustements se font rapidement et en toute simplicité, offrant une gestion flexible de leur activité.

2

Transparence et accessibilité

Une personne ayant le rôle coach aura une rubrique dédiée pour affichée les disponibilitée ainsi que c'est tarif honoraire.

Les informations mises à jour sont immédiatement visibles par les utilisateurs, garantissant une transparence totale et une meilleure communication avec les clients. Cette fonctionnalité permet aux coaches de gérer leur profil de manière efficace et professionnelle.

3

Un outil de gestion adapté

La page de profil est conçue pour faciliter le quotidien des coaches en leur offrant un espace dédié à l'optimisation de leurs prestations. Ils peuvent ainsi ajuster leurs offres en fonction de leur évolution et assurer un suivi optimal de leur activité.

26. Planning

Planning de Disponibilités

< Précédente

Aujourd'hui

Suivante >

avril 2025

Horaire	lun. 21	mar. 22	mer. 23	jeu. 24
8:00	✓	✓		
9:00	✓	✓		
10:00	✗	✓		
11:00	✓	✓		
12:00	✓	✗		
13:00	✓	✗		
14:00	✓	✓		
15:00	✓	✓		
16:00	✗	✗		
17:00	✓	✗		
18:00	✓	✓		

Mon défini

Disponible

Occupé

Profil

Client

Accueil

Planning

Rendez-vous

1

Gestion des disponibilités

Grâce à cette fonctionnalité, chaque coach peut afficher ses disponibilités pour différentes journées et heures. En quelques clics, il peut organiser son emploi du temps pour proposer des créneaux adaptés à ses clients, tout en gardant une vue d'ensemble sur ses rendez-vous.

2

Blocage automatique des créneaux

Lorsqu'un utilisateur réserve un créneau horaire, celui-ci est automatiquement bloqué, évitant ainsi toute double réservation. Ce système intelligent permet une gestion fluide et optimale des séances, garantissant aux coaches un planning structuré et sans conflit.

3


Une planification simplifiée

L'interface intuitive offre aux coaches une navigation claire et rapide. Ils peuvent ajuster leurs disponibilités en fonction de leurs besoins et assurer un suivi efficace de leur emploi du temps, tout en offrant aux clients une visibilité en temps réel sur leurs options de réservation.

27. Client

Rechercher un utilisateur...

7 utilisateurs trouvés




Robin OUDARD

aa@gmail.com

19 ans ♂ homme

55 kg • 178 cm

Dernière connexion: 25 avril 2025 à 11:52




Pierre

dimitrovpiteo94@gmail.com

28 ans ♂ homme

70 kg • 176 cm


Dernière connexion: 24 avril 2025 à 12:41



Kems

sandidkemy12003@gmail.com


Dernière connexion: 24 avril 2025 à 12:45



Anto

dinhantoine05@gmail.com

Dernière connexion: 24 avril 2025 à 12:44



Aymeric

test@gmail.com

555 ans ♂ homme

555 kg • 555 cm

Profil

Client

Accueil

Planning

Rendez-vous

1

Gestion des clients

La rubrique client permet à l'utilisateur coach de consulter la liste complète de ses clients ainsi que leurs informations personnelles. Il pourra visualiser les données telles que l'âge, le poids, la taille et les coordonnées de contact. Cette interface facilite le suivi personnalisé et permet d'accéder rapidement aux profils détaillés en un seul clic.

2

Statistiques et progression

Cette fonctionnalité offre une analyse visuelle de la progression de chaque client à travers des graphiques personnalisés. Les indicateurs clés comme l'évolution du poids, les performances, l'assiduité et la régularité des séances sont automatiquement calculés et affichés. Ces données permettent au coach d'ajuster les programmes d'entraînement en fonction des objectifs fixés et des résultats obtenus.

30



BTS SIO 2023-2025

10

ANNEXES

ALAIN RABIE

28. Projet personnel

Ce projet représente pour moi une avancée personnelle significative, car il a été le fruit de nombreuses recettes élaborées avec soin.

Grâce à mon expertise en nutrition et en sport, j'ai pu établir les meilleurs calculs caloriques et les exercices les plus efficaces pour permettre à l'utilisateur d'atteindre ses objectifs de manière encadrée et rapide.

Cette application a été un objectif de longue date. Un site web a été créé à cette adresse : <http://rabie.fr/CallApp-main/index.html>.

À terme, cette application constituera une belle opportunité pour moi de lancer mon entreprise et de partager mes connaissances avec un public plus large.

De plus, je prévois d'intégrer un système d'abonnement afin de générer des bénéfices et de soutenir le développement continu de l'application.

Cela me permettra également d'envisager une expansion future de mon entreprise, en offrant encore plus de fonctionnalités et de services à mes utilisateurs.

29. Annexes

LIEN DU GITHUB :

[HTTPS://GITHUB.COM/ALAINRABIE/AP4.GIT](https://github.com/ALAINRABIE/AP4.GIT)

IDENTIFIANT DE TEST UTILISATEUR :

MAIL : TEST@GMAIL.COM

MOTS DE PASSE : TEST123

PORTFOLIO

RABIE.FR

LIEN DU SITE DÉDIÉ À L'APPLICATION:

[HTTP://RABIE.FR/CALLAPP-MAIN/INDEX.HTML](http://rabie.fr/callapp-main/index.html)

