

Cahier des Charges – Fitness Log (Suivi d'Entraînement Sportif)

BAH TOURE	Oumar toure
EL MAGHRAOUI	Ismail
ENGAR	Aymen
HORTICE SEDJRO AUBED	Adognibo

1. Présentation générale

L'application **Fitness Log** est un journal sportif permettant à l'utilisateur de suivre ses séances d'entraînement, d'analyser ses progrès et d'atteindre ses objectifs hebdomadaires. Elle offre une interface simple pour enregistrer des activités, calculer automatiquement les calories estimées, visualiser des statistiques et organiser le suivi sur le long terme.

2. Objectifs du projet

- Mettre en pratique la **persistance des données** avec SQLite.
- Concevoir une interface ergonomique adaptée au suivi sportif.
- Mettre en œuvre des **calculs automatiques** (calories, totaux, durée).
- Construire un tableau de bord (**dashboard**) dynamique avec graphiques.
- Apprendre à structurer une application Flutter de manière modulaire (models, services, providers).

3. Problématique

Aujourd'hui, de nombreuses personnes souhaitent suivre leurs activités sportives mais manquent d'un outil simple, visuel, et entièrement hors-ligne.

Fitness Log répond à ce besoin en offrant un système complet pour :

- enregistrer ses entraînements,
- visualiser les statistiques,
- suivre les objectifs hebdomadaires.

4. Fonctionnalités attendues

4.1 Gestion des séances d'entraînement (obligatoire)

Une séance (Workout) contient :

- type de sport (Cardio, Musculation, Marche, Course, Yoga...)
- durée (en minutes)
- estimation calories brûlées
- date
- notes optionnelles

Actions :

- Ajouter une séance
- Modifier
- Supprimer
- Dupliquer une séance

4.2 Liste des séances

L'utilisateur voit l'historique complet avec :

- icône selon type d'activité
- durée
- calories
- date formatée

Filtres :

- par type (Cardio, Musculation...)
- par période (Jour / Semaine / Mois)

4.3 Statistiques (Dashboard)

Interface visuelle comprenant :

- total des calories brûlées cette semaine
- durée totale des entraînements
- nombre de séances
- graphiques (barres ou lignes) :
 - calories par jour
 - évolution sur plusieurs semaines
- graphique circulaire : répartition des types d'activités

4.4 Objectif hebdomadaire

L'utilisateur peut définir un objectif, ex :

- 150 minutes/semaine
- ou 2000 calories/semaine

L'application doit afficher :

- une barre de progression
- un pourcentage d'avancement
- une couleur selon le statut :
 - vert (<70%)
 - orange (70-100%)
 - rouge (>100%)

4.5 Stockage des données (SQLite)

Base fitness.db

Table recommandée :

```
workouts(  
  id INTEGER PRIMARY KEY,  
  activityType TEXT,  
  duration INTEGER,  
  calories INTEGER,  
  notes TEXT,  
  date TEXT
```

)

Requêtes obligatoires :

- INSERT
- UPDATE
- DELETE
- SELECT par date / type
- SUM(calories) par semaine
- COUNT séances par type

4.6 Interface & ergonomie

- UI moderne, simple et claire
- Icônes thématiques pour chaque sport
- Graphiques lisibles
- Mode sombre/claire (bonus)

5. Fonctionnalités Optionnelles

5.1 Suivi GPS (pour marche/course)

- distance parcourue
- vitesse moyenne
- carte du parcours (Google Maps)

5.2 Export CSV

Permettre au sportif d'exporter :

- toutes ses séances
- ses statistiques

5.3 Notifications

Exemples :

- "Vous n'avez pas encore atteint votre objectif cette semaine"
- "Bravo, objectif atteint !"

5.4 Recommandations automatiques

Selon :

- activité la plus faible
- calories restantes pour objectif

5.5 Analyse avancée

- séquence des meilleures semaines
- record personnel (calories max en un jour)

6. Contraintes techniques

- Le projet doit être réalisé en **Flutter**.
- Les données doivent être persistées via **SQLite**.
- Graphiques réalisés avec **fl_chart** ou équivalent.
- Architecture recommandée :
 - models/
 - services/ (SQLite, statistiques)
 - providers/ (state management)
 - views/ (UI)
 - widgets/