

# Rapport de Projet Mobile

## FitTrack

Application de Suivi Fitness et Gamification

**Technologie :** Flutter & Dart

*Auteur :*

Aiman Merouania  
Youssef Abdelmoumene

*Professeur :*

Anass DEROUSSI

17 décembre 2025

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contexte et Motivation . . . . .	3
1.2	Problématique . . . . .	3
1.3	Solution Proposée . . . . .	3
<b>2</b>	<b>Conception et Architecture Technique</b>	<b>4</b>
2.1	Choix Technologiques . . . . .	4
2.1.1	Flutter et Dart . . . . .	4
2.1.2	Gestion d'État : Pattern BLoC . . . . .	4
2.2	Clean Architecture . . . . .	4
2.2.1	1. Domain Layer (Cœur) . . . . .	4
2.2.2	2. Data Layer (Données) . . . . .	4
2.2.3	3. Presentation Layer (UI) . . . . .	5
2.3	Injection de Dépendances . . . . .	5
<b>3</b>	<b>Détail des Fonctionnalités Implémentées</b>	<b>6</b>
3.1	Module 1 : Gestion des Programmes (Workouts) . . . . .	6
3.2	Module 2 : La Séance Active (Le Cœur du Projet) . . . . .	6
3.3	Module 3 : Gamification . . . . .	6
3.4	Module 4 : Calendrier et Intégration . . . . .	7
<b>4</b>	<b>Présentation des Écrans de l'Application</b>	<b>8</b>
4.1	Écran d'Authentification . . . . .	8
4.1.1	Écran de Connexion (Login) . . . . .	8
4.1.2	Écran d'Inscription . . . . .	9
4.2	Écran d'Accueil (Home) . . . . .	10
4.3	Gestion des Programmes . . . . .	11
4.3.1	Bibliothèque de Programmes . . . . .	11
4.3.2	Création de Nouveau Programme . . . . .	12
4.4	Gestion des Exercices . . . . .	13
4.4.1	Bibliothèque d'Exercices . . . . .	13
4.4.2	Création d'Exercice Personnalisé . . . . .	14
4.5	Calendrier et Planification . . . . .	15
4.6	Profil Utilisateur . . . . .	16
4.7	Statistiques et Analyses . . . . .	17
4.8	Synthèse de l'Expérience Utilisateur . . . . .	17

---

<b>5</b>	<b>Interface Utilisateur (UI/UX)</b>	<b>18</b>
5.1	Design System "Neon Dark" . . . . .	18
5.2	Composants Personnalisés . . . . .	18
5.2.1	GlassCard . . . . .	18
5.2.2	XP Bar . . . . .	18
5.3	Animations . . . . .	18
<b>6</b>	<b>Bilan Personnel et Perspectives</b>	<b>19</b>
6.1	Difficultés et Apprentissages . . . . .	19
6.2	Améliorations Futures . . . . .	19
6.3	Conclusion . . . . .	19

# Chapitre 1

## Introduction

### 1.1 Contexte et Motivation

Le marché des applications de fitness est saturé de solutions de suivi, mais beaucoup souffrent de deux défauts majeurs : une complexité excessive pour le pratiquant moyen, ou un manque d'engagement à long terme. La musculation et le fitness sont des disciplines qui demandent de la régularité. Or, la perte de motivation est la cause principale d'abandon après les premiers mois.

C'est dans ce contexte que s'inscrit le projet **FitTrack**. L'idée n'est pas seulement de fournir un carnet d'entraînement numérique, mais de transformer l'expérience sportive en un jeu (Gamification) pour maximiser la rétention et le plaisir de l'utilisateur.

### 1.2 Problématique

Comment concevoir une application mobile performante, maintenable et engageante qui permet à l'utilisateur de :

1. Créer et personnaliser ses propres programmes d'entraînement complexe.
2. Suivre ses séances en temps réel sans friction technique.
3. Visualiser sa progression de manière gratifiante.

### 1.3 Solution Proposée

FitTrack est une application développée avec le framework **Flutter**, choisie pour sa capacité à produire des interfaces natives fluides sur Android et iOS à partir d'une base de code unique. L'application intègre :

- Un système de **Gamification RPG** (Barre d'expérience, Niveaux tel que "Titan", "Légende").
- Une architecture **Clean Architecture** pour assurer la solidité du code.
- Des fonctionnalités avancées comme l'export JSON, le mode Sombre/Clair dynamique et un calendrier interactif.

# Chapitre 2

## Conception et Architecture Technique

### 2.1 Choix Technologiques

#### 2.1.1 Flutter et Dart

Le choix de Flutter s'est imposé pour plusieurs raisons :

- **Performance** : Le moteur de rendu Skia permet du 60/120 FPS constant.
- **Hot Reload** : Accélère drastiquement le cycle de développement.
- **Widget Tree** : Une approche déclarative de l'UI parfaitement adaptée aux interfaces dynamiques comme celles de FitTrack.

#### 2.1.2 Gestion d'État : Pattern BLoC

Pour gérer la complexité des états (authentification, chronomètres, formulaires), nous avons opté pour le pattern **BLoC (Business Logic Component)** via la librairie `flutter_bloc`. Ce pattern permet de séparer strictement la logique métier de l'interface graphique.

- **Events** : L'UI envoie des événements (ex : `WorkoutStarted`, `SetValidated`).
- **States** : Le BLoC traite l'événement et émet un nouvel état (ex : `ActiveWorkoutState` avec le nouveau temps écoulé).

### 2.2 Clean Architecture

Le projet suit rigoureusement les principes de la Clean Architecture en trois couches :

#### 2.2.1 1. Domain Layer (Cœur)

C'est la couche la plus interne. Elle ne dépend d'aucune librairie externe ni du framework Flutter.

- **Entities** : Objets métiers purs (ex : `WorkoutEntity`, `ExerciseSetEntity`).
- **Repositories Interfaces** : Contrats définissant les opérations possibles (ex : `saveWorkout()`, `getScheduledWorkouts()`).

#### 2.2.2 2. Data Layer (Données)

Responsable de la récupération et de la persistance.

- **DTO (Data Transfer Objects)** : Modèles incluant les méthodes de sérialisation JSON ('fromMap', 'toMap').
- **Repositories Implementations** : Code concret interagissant avec les sources de données (Firebase ou Mock en mémoire local).

### 2.2.3 3. Presentation Layer (UI)

La couche visible par l'utilisateur.

- **Pages** : Écrans principaux ('HomePage', 'ActiveWorkoutPage').
- **Widgets** : Composants réutilisables ('XPBar', 'GlassCard').
- **Cubits/BLoCs** : Lien entre l'UI et le Domaine.

## 2.3 Injection de Dépendances

Nous utilisons `RepositoryProvider` et `BlocProvider` pour injecter les dépendances dans l'arbre des widgets. Cela rend le code testable et modulaire. Par exemple, changer de la base de données Mock à Firebase ne demande de modification qu'à un seul endroit ('main.dart').

# Chapitre 3

## Détail des Fonctionnalités Implémentées

### 3.1 Module 1 : Gestion des Programmes (Workouts)

Ce module permet la création complète de séances. **Structure de données** : Un `Workout` contient une liste de `WorkoutExercise`, qui contient elle-même une liste de `ExerciseSet`. Cette structure hiérarchique complexe a nécessité une gestion fine des mises à jour d'état dans le `WorkoutEditorCubit`.

```
1 class WorkoutEntity {
2     final String id;
3     final String name;
4     final List<WorkoutExerciseEntity> exercises;
5     final bool isTemplate;
6     // ...
7 }
```

Listing 3.1 – Exemple d'Entité Workout

### 3.2 Module 2 : La Séance Active (Le Cœur du Projet)

C'est la fonctionnalité la plus critique. L'utilisateur lance une séance et l'application doit :

- Maintenir un chronomètre global.
- Gérer un minuteur de repos indépendant qui se superpose à l'interface.
- Permettre la modification des performances (Poids/Reps) à la volée.

**Challenge Technique** : Le chronomètre ne doit pas être bloqué par les interactions UI. Nous utilisons un `Timer.periodic` dans le `ActiveWorkoutCubit` qui émet un nouvel état chaque seconde.

### 3.3 Module 3 : Gamification

Pour rendre l'application addictive de manière positive, nous avons implémenté un système de récompense.

- **Logique** : Chaque séance terminée rapporte 150 XP.

- **Algorithmme** : Le seuil d'XP nécessaire pour le niveau suivant augmente de 20% à chaque niveau ( $MaxXP = MaxXP * 1.2$ ).
- **Titres** : Des titres honorifiques sont débloqués aux niveaux 5, 10, 20 et 50.

```
1 void addXP(int amount) {  
2     int newXP = state.stats.currentXP + amount;  
3     // Boucle pour gerer plusieurs montees de niveau d'un coup  
4     while (newXP >= maxXP) {  
5         newXP -= maxXP;  
6         level++;  
7         maxXP = (maxXP * 1.2).round();  
8         leveledUp = true;  
9     }  
10    // ... Emission du nouvel etat  
11 }
```

Listing 3.2 – Logique de Level Up dans GamificationCubit

### 3.4 Module 4 : Calendrier et Intégration

Le calendrier n'est pas juste visuel. Il est le point de convergence de l'application.

- **Flux complet** : Planifier → Voir dans le calendrier → Cliquer sur "Démarrer" → S'entraîner → Finir.
- **Feedback** : À la fin de la séance, le `ActiveWorkoutCubit` communique avec le `CalendarRepository` pour cocher la séance comme "Faite".

# Chapitre 4

## Présentation des Écrans de l'Application

Cette section présente l'ensemble des interfaces utilisateur de FitTrack avec leurs fonctionnalités respectives.

### 4.1 Écran d'Authentification

#### 4.1.1 Écran de Connexion (Login)

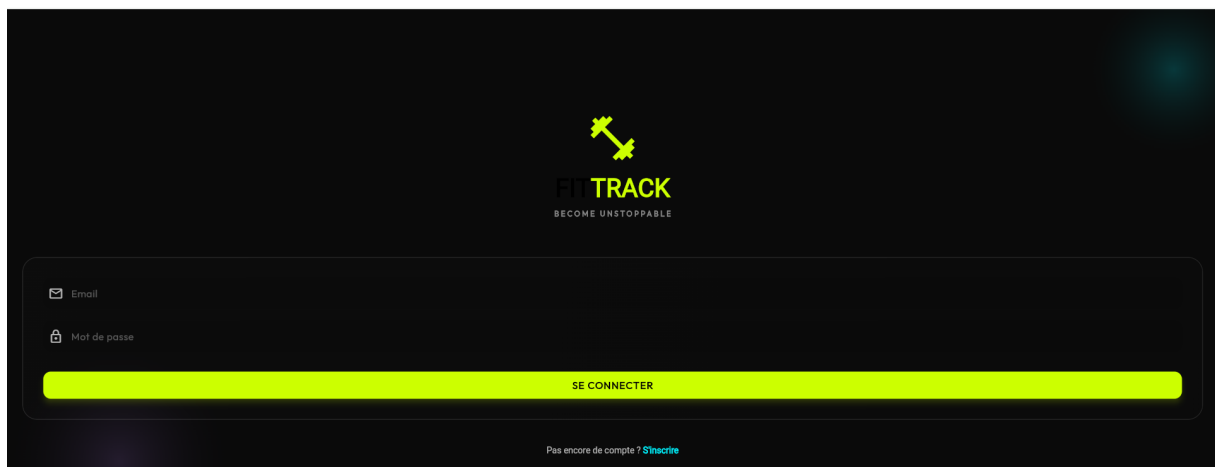


FIGURE 4.1 – Écran de connexion FitTrack

**Description :** Premier point de contact avec l'application, l'écran de connexion arbore le logo FitTrack avec le slogan motivant "BECOME UNSTOPPABLE".

**Éléments UI :**

- Logo avec icône d'haltère en jaune néon
- Champs de saisie pour l'email et le mot de passe
- Bouton "SE CONNECTER" en jaune vif
- Lien vers l'inscription en cyan

**Design :** Fond noir profond avec accents jaune néon créant un contraste fort et énergétique. L'utilisation du jaune symbolise la puissance et l'énergie, parfait pour une application fitness.

**Fonctionnalités :**

- Authentification utilisateur via email/mot de passe
- Navigation vers l'écran d'inscription
- Validation des champs avec feedback visuel

### 4.1.2 Écran d'Inscription

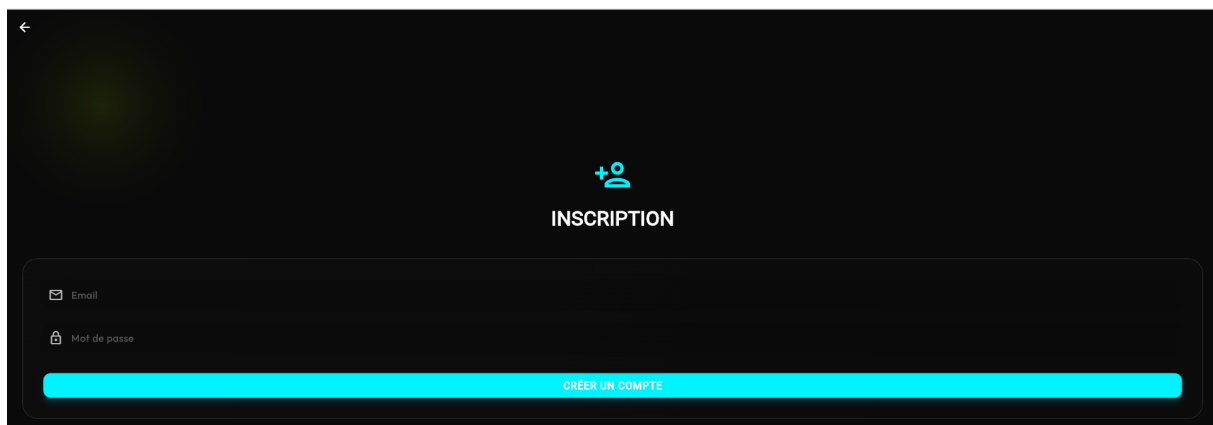


FIGURE 4.2 – Écran d'inscription

**Description :** Interface minimaliste permettant la création rapide d'un compte.

**Éléments UI :**

- Titre "INSCRIPTION" avec icône utilisateur
- Champs email et mot de passe
- Bouton "CRÉER UN COMPTE" en jaune brillant

**Fonctionnalités :**

- Création de compte avec validation des données
- Vérification de la force du mot de passe
- Redirection automatique après inscription réussie

## 4.2 Écran d'Accueil (Home)

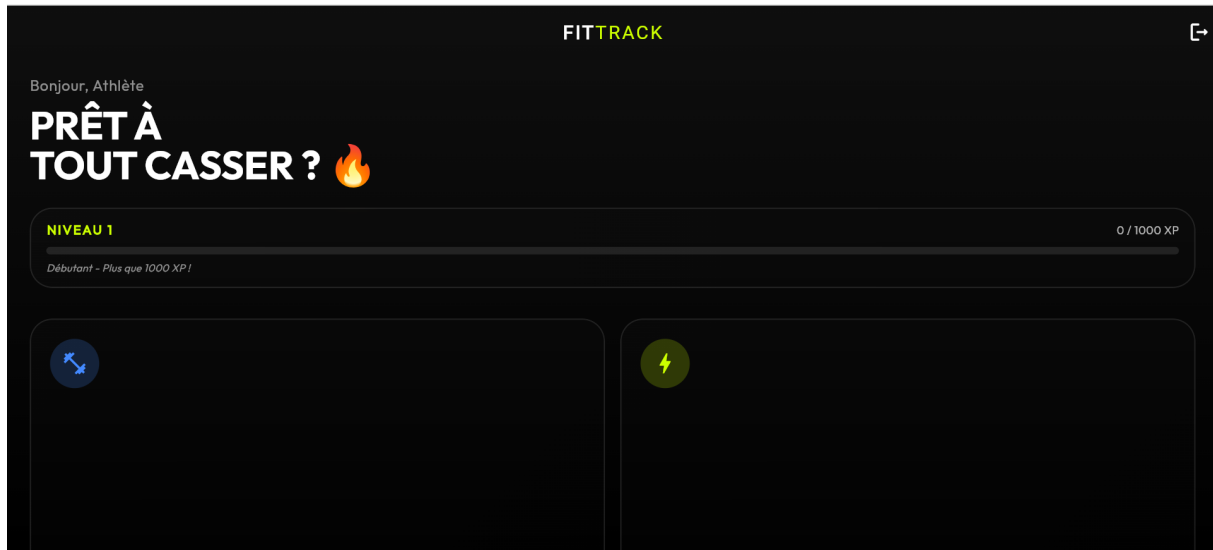


FIGURE 4.3 – Écran d'accueil avec système de gamification

**Description :** Le cœur de l'expérience FitTrack, cet écran accueille l'utilisateur avec un message personnalisé et affiche son statut RPG.

**Éléments UI :**

- Message d'accueil : "Bonjour, Athlète"
- Slogan motivant : "PRÊT À TOUT CASSER ? "
- Barre d'expérience : "NIVEAU 1" avec progression 0/1000 XP
- Deux cartes d'action principales avec icônes distinctives (haltère bleu, éclair jaune)

**Design :** Fond noir avec dégradé subtil. La barre XP utilise un bleu cyan pour créer une sensation de progression. Les cartes d'action utilisent des effets de Glassmorphism.

**Fonctionnalités :**

- Affichage dynamique du niveau et de l'XP de l'utilisateur
- Navigation rapide vers les séances programmées
- Accès aux statistiques de performance
- Mise à jour en temps réel de la progression

## 4.3 Gestion des Programmes

### 4.3.1 Bibliothèque de Programmes

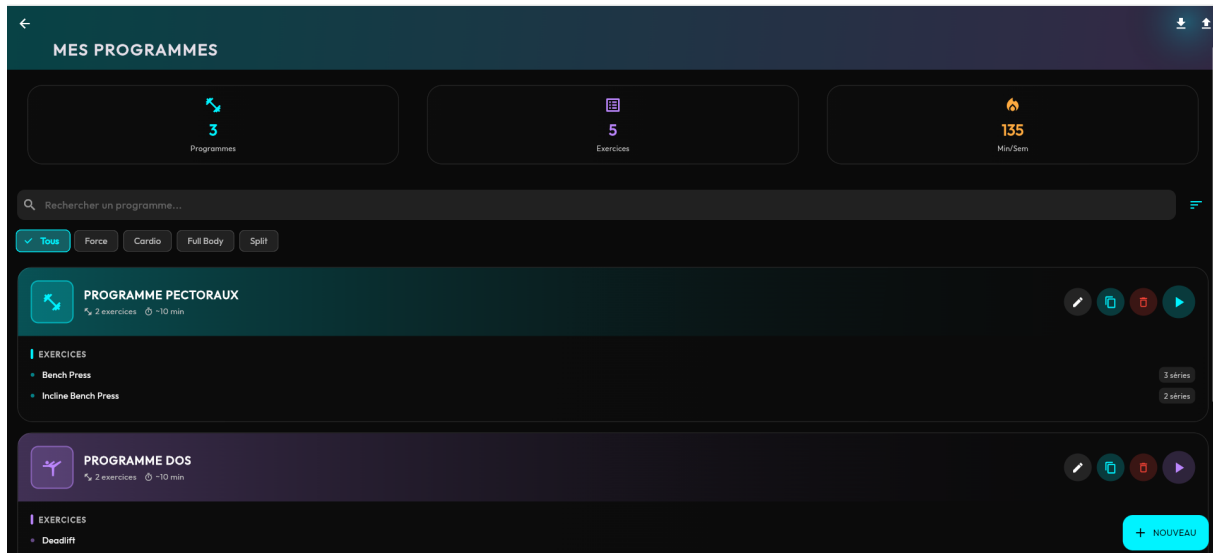


FIGURE 4.4 – Liste des programmes d'entraînement

**Description :** Vue d'ensemble de tous les programmes créés par l'utilisateur avec statistiques globales.

**Éléments UI :**

- En-tête avec 3 statistiques clés : 3 Programmes, 5 Exercices, 135 minutes
- Barre de recherche pour filtrer les programmes
- Filtres par catégorie : Tous, Force, Cardio, Full Body, Split
- Cartes de programmes affichant nom, exercices inclus et durée
- Bouton "+ NOUVEAU" en bas à droite

**Design :** Dégradé bleu-violet sur fond noir. Chaque carte de programme utilise un effet de verre avec bordure subtile. Les icônes de filtres sont colorées selon leur type.

**Fonctionnalités :**

- Recherche textuelle en temps réel
- Filtrage par catégorie d'entraînement
- Actions contextuelles : Éditer, Dupliquer, Supprimer
- Affichage du nombre d'exercices et durée estimée
- Création rapide de nouveau programme

### 4.3.2 Création de Nouveau Programme

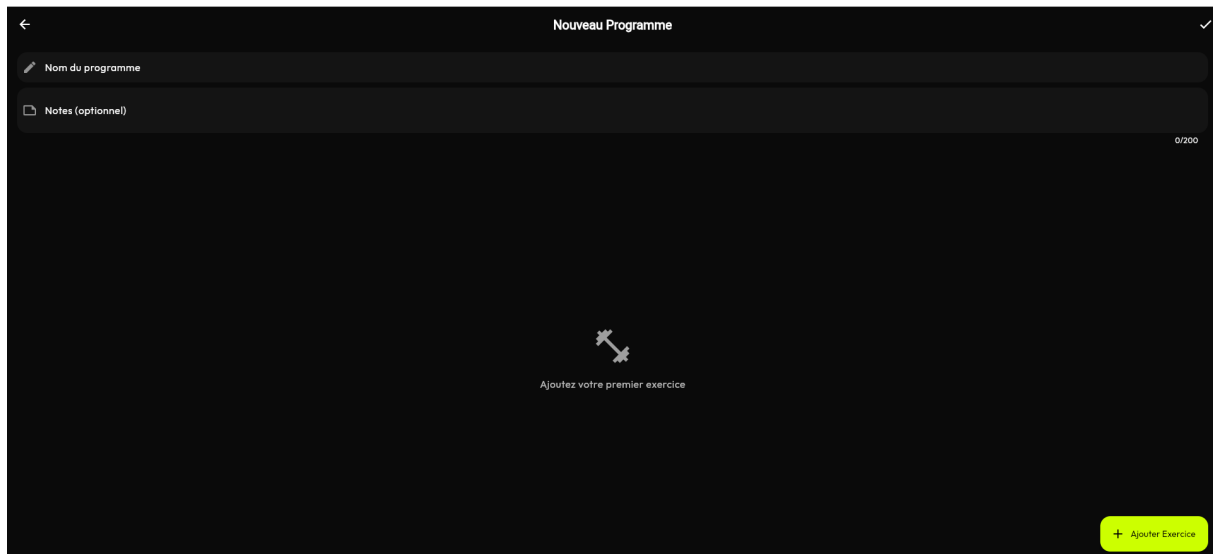


FIGURE 4.5 – Écran de création de programme

**Description :** Interface épurée pour construire un nouveau programme d'entraînement de zéro.

**Éléments UI :**

- Champ "Nom du programme" en haut
- Section "Notes (optionnel)" pour ajouter des détails
- Zone centrale avec message : "Ajoutez votre premier exercice"
- Bouton "+ AJOUTER EXERCICE" en cyan

**Design :** Fond noir minimaliste mettant en avant le bouton d'action principal. L'utilisation d'icônes d'haltères crée une cohérence visuelle.

**Fonctionnalités :**

- Saisie du nom du programme avec validation
- Ajout de notes descriptives optionnelles
- Navigation vers la bibliothèque d'exercices
- Sauvegarde automatique en tant que brouillon

## 4.4 Gestion des Exercices

### 4.4.1 Bibliothèque d'Exercices

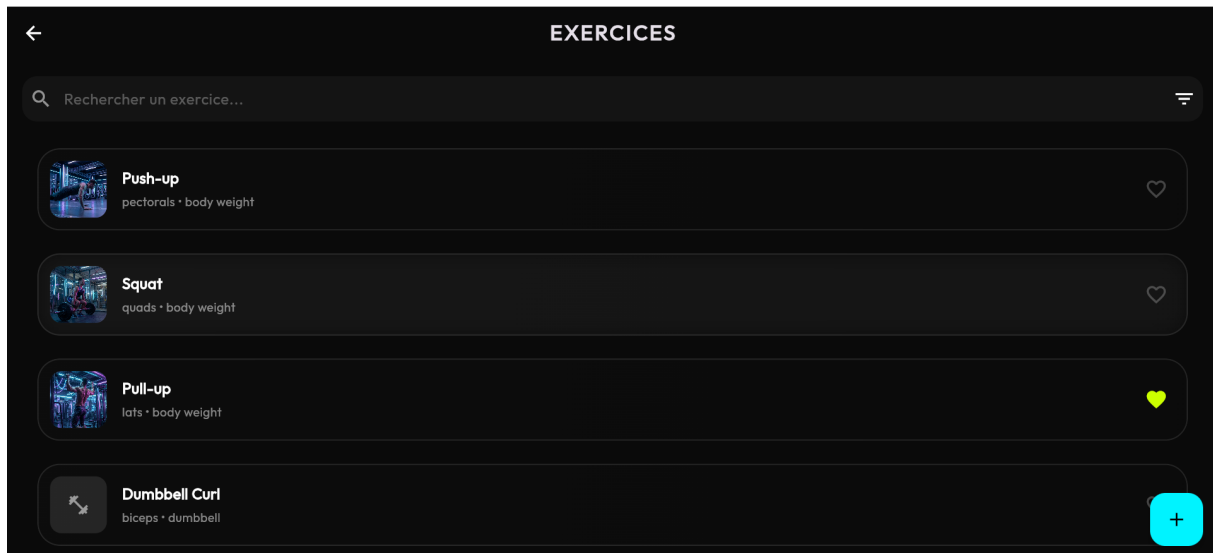


FIGURE 4.6 – Liste des exercices disponibles

**Description :** Catalogue complet des exercices avec recherche et favoris.

**Éléments UI :**

- Barre de recherche : "Rechercher un exercice..."
- Liste d'exercices avec vignettes illustratives
- Informations : Nom, groupe musculaire ciblé, type d'équipement
- Icône cœur pour marquer en favori
- Bouton "+" flottant pour créer un exercice personnalisé

**Exercices affichés (exemples) :**

- Push-up : pectorals • body weight
- Squat : quads • body weight
- Pull-up : lats • body weight (marqué favori)
- Dumbbell Curl : biceps • dumbbell

**Design :** Fond noir avec cartes pour chaque exercice. Les vignettes utilisent des illustrations en bleu néon montrant la position de l'exercice. Icônes en jaune pour les favoris.

**Fonctionnalités :**

- Recherche instantanée par nom d'exercice
- Système de favoris pour accès rapide
- Visualisation des détails de l'exercice (au tap)
- Création d'exercices personnalisés
- Catégorisation automatique par muscle et équipement

### 4.4.2 Création d'Exercice Personnalisé

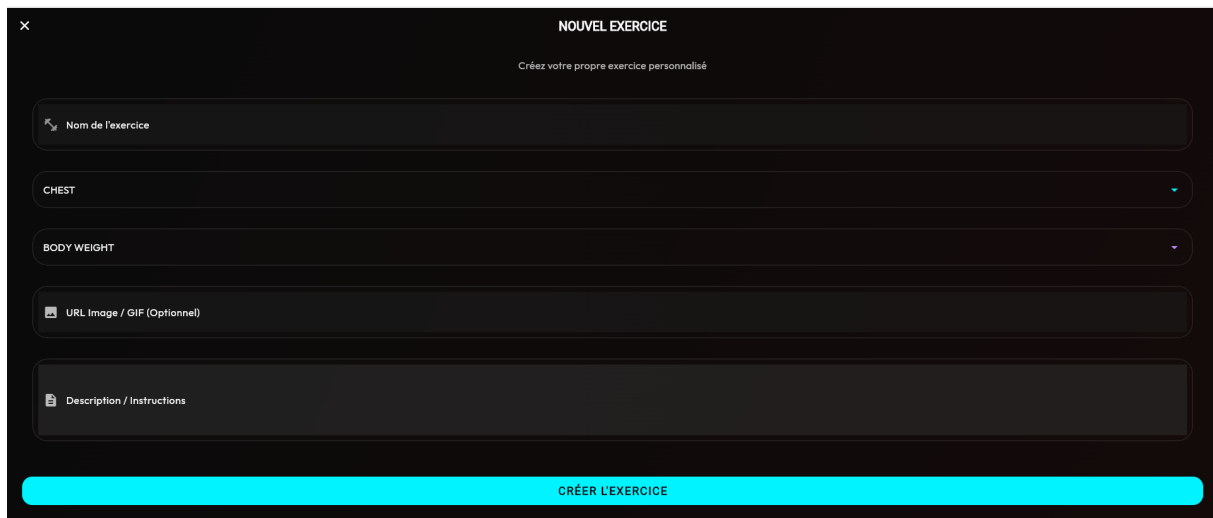


FIGURE 4.7 – Formulaire de création d'exercice

**Description :** Interface complète pour définir un exercice sur mesure.

**Éléments UI :**

- Champ "Nom de l'exercice"
- Sélecteur "CHEST" (groupe musculaire)
- Sélecteur "BODY WEIGHT" (type d'équipement)
- Champ optionnel "URL Image / GIF (Optionnel)"
- Zone de texte "Description / Instructions"
- Bouton "CRÉER L'EXERCICE" en cyan

**Design :** Fond noir avec champs de saisie encadrés. Les sélecteurs utilisent des menus déroulants stylisés. Le bouton d'action principal occupe toute la largeur pour une accessibilité maximale.

**Fonctionnalités :**

- Saisie du nom avec vérification des doublons
- Sélection du groupe musculaire parmi une liste prédéfinie
- Choix du type d'équipement nécessaire
- Ajout d'une image ou GIF via URL
- Rédaction d'instructions détaillées
- Validation et ajout à la bibliothèque personnelle

## 4.5 Calendrier et Planification

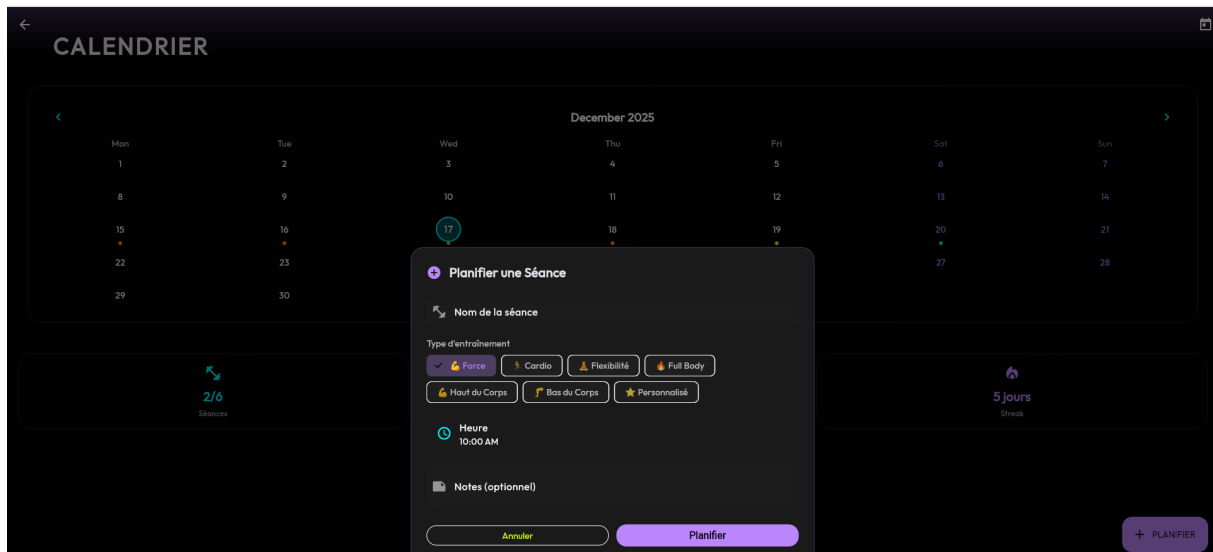


FIGURE 4.8 – Calendrier avec planification de séance

**Description :** Vue calendrier mensuelle permettant de planifier et visualiser les séances.

**Éléments UI :**

- Calendrier du mois (Décembre 2025) avec dates cliquables
- Indicateurs visuels : 2/6 Exercices, 5 jours streak
- Modale de planification superposée avec :
  - Champ "Nom de la séance"
  - Sélecteur de type d'entraînement (Force, Cardio, Flexibilité, Full Body, Haut du Corps, Bas du Corps, Personnalisé)
  - Sélecteur d'heure : 10 :00 AM
  - Section "Notes (optionnel)"
  - Boutons "Annuler" et "Planifier"

**Design :** Le calendrier utilise un fond noir avec la date sélectionnée (17) entourée d'un cercle cyan. La modale utilise un effet de verre semi-transparent. Les types d'entraînement sont représentés par des badges colorés avec icônes.

**Fonctionnalités :**

- Navigation mensuelle (flèches gauche/droite)
- Sélection de date pour planification
- Choix du type d'entraînement avec icônes distinctives
- Définition de l'heure de la séance
- Ajout de notes contextuelles
- Visualisation des séances planifiées sur le calendrier
- Indicateurs de progression (streak, séances complétées)

## 4.6 Profil Utilisateur

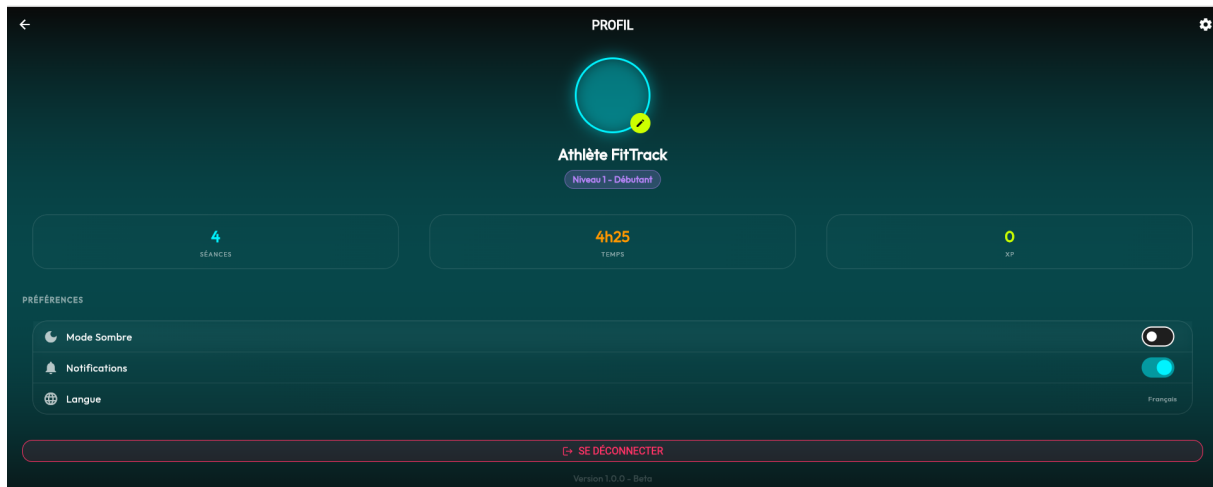


FIGURE 4.9 – Écran de profil et paramètres

**Description :** Page de profil affichant les statistiques personnelles et les préférences.

**Éléments UI :**

- Avatar utilisateur avec bouton d'édition
- Nom : "Athlète FitTrack"
- Badge de niveau : "Niveau 1 - Débutant" (violet)
- Statistiques :
  - 4 SÉANCES
  - 4h25 de temps total
  - 0 XP
- Section PRÉFÉRENCES :
  - Mode Sombre (toggle activé)
  - Notifications (toggle activé)
  - Langue (Français)
- Bouton "SE DÉCONNECTER" en rouge

**Design :** Fond dégradé bleu-vert (teal). L'avatar est encadré d'un cercle cyan. Les statistiques sont présentées dans des cartes semi-transparentes. Les toggles utilisent des couleurs vives pour indiquer l'état actif.

**Fonctionnalités :**

- Affichage des statistiques personnelles
- Modification de l'avatar et du nom
- Basculement du mode sombre/clair
- Gestion des notifications push
- Changement de langue de l'interface
- Déconnexion sécurisée

## 4.7 Statistiques et Analyses

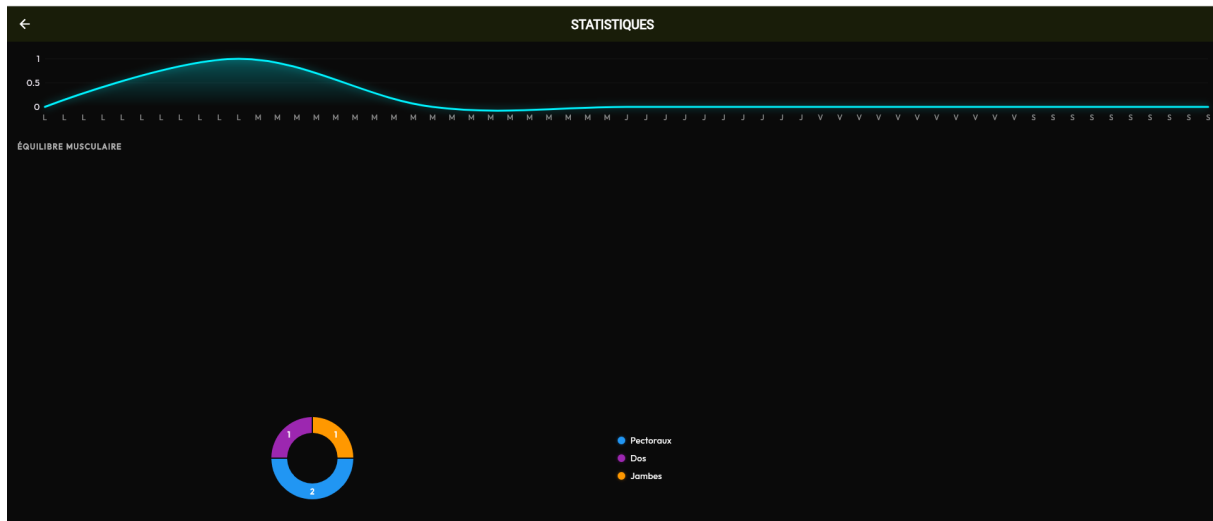


FIGURE 4.10 – Tableau de bord statistiques

**Description :** Visualisation des données d'entraînement et d'équilibre musculaire.

**Éléments UI :**

- Graphique linéaire montrant l'évolution dans le temps (janvier à septembre)
- Section "ÉQUILIBRE MUSCULAIRE" avec diagramme circulaire :
  - Pectoraux (violet)
  - Dos (cyan)
  - Jambes (orange)

**Design :** Fond noir avec graphiques utilisant des couleurs néon pour la lisibilité. Le graphique linéaire utilise un dégradé cyan. Le diagramme circulaire emploie trois couleurs distinctes pour différencier les groupes musculaires.

**Fonctionnalités :**

- Suivi de la progression temporelle
- Visualisation de l'équilibre musculaire pour éviter les déséquilibres
- Analyse des tendances d'entraînement
- Identification des groupes musculaires négligés
- Export des données (fonctionnalité cachée dans le menu)

## 4.8 Synthèse de l'Expérience Utilisateur

L'ensemble des écrans de FitTrack suit une philosophie de design cohérente :

- **Cohérence visuelle :** Palette de couleurs néon (cyan, jaune, violet) sur fond noir
- **Glassmorphism :** Utilisation systématique d'effets de verre pour les cartes
- **Iconographie claire :** Chaque fonction est accompagnée d'une icône parlante
- **Feedback visuel :** Les interactions sont toujours confirmées visuellement
- **Navigation intuitive :** Retours arrière, boutons flottants, navigation en bas

Cette approche garantit une courbe d'apprentissage minimale tout en offrant une expérience premium et engageante.

# Chapitre 5

## Interface Utilisateur (UI/UX)

### 5.1 Design System "Neon Dark"

L'identité visuelle de FitTrack repose sur un thème sombre profond ("Dark Mode") accentué par des couleurs néons (Vert, Bleu, Violet) pour donner une énergie sportive et moderne.

### 5.2 Composants Personnalisés

#### 5.2.1 GlassCard

Pour donner de la profondeur, nous avons créé un widget `GlassCard` utilisant l'effet de flou (Glassmorphism) qui est utilisé partout dans l'application pour encadrer le contenu.

#### 5.2.2 XP Bar

La barre d'expérience est un composant complexe qui utilise `Stack` et `LayoutBuilder` pour animer la jauge de remplissage de manière fluide (`AnimatedContainer`).

### 5.3 Animations

L'utilisation de la librairie `flutter_animate` permet d'ajouter des micro-interactions (apparitions en fondu, glissements) qui rendent l'application vivante et professionnelle.

# Chapitre 6

## Bilan Personnel et Perspectives

### 6.1 Difficultés et Apprentissages

Ce projet a été l'occasion de confronter la théorie à la pratique.

- **Complexité de l'État** : Gérer l'état d'une séance active avec des dizaines de champs modifiables (séries, poids) a été un défi pour garder le code propre. Le pattern BLoC a été salvateur ici.
- **Architecture** : Séparer strictement le Domaine de la Data semblait verbeux au début, mais a prouvé son utilité lorsqu'il a fallu ajouter des méthodes au Repository sans casser l'UI.

### 6.2 Améliorations Futures

Si le temps le permettait, voici les prochaines étapes :

1. **Synchronisation Cloud Réelle** : Passer du Mock/Local à une vraie base de données Firestore pour retrouver ses données sur plusieurs appareils.
2. **Mode social** : Ajouter des amis et comparer les niveaux.
3. **Wear OS / WatchOS** : Une application compagnon pour montre connectée pour gérer la séance sans sortir le téléphone.

### 6.3 Conclusion

FitTrack est aujourd'hui une application fonctionnelle, stable et agréable à utiliser. Elle remplit tous les objectifs fixés initialement et pose des bases solides pour une évolution future. Ce projet a permis de valider des compétences clés en développement mobile moderne et en architecture logicielle.