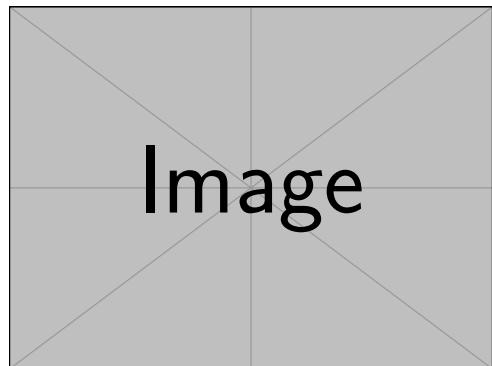


Software Design Document

Application Mobile BasedFit



Version 1.0

17 novembre 2025

Table des matières

Introduction	3
1 Modèle Conceptuel	4
1.1 Sujet de Conception	4
1.2 Parties Prenantes	4
1.3 Préoccupations de Conception	4
2 Vues de Conception	5
2.1 Vue de Contexte	5
2.1.1 Préoccupations	5
2.1.2 Éléments de Conception	5
2.1.3 Langages de Conception	5
2.2 Vue de Composition	5
2.2.1 Préoccupations	5
2.2.2 Éléments de Conception	5
2.2.3 Structure Hiérarchique	5
2.3 Vue Logique	6
2.3.1 Préoccupations	6
2.3.2 Éléments de Conception	6
2.4 Vue des Interfaces	6
2.4.1 Préoccupations	6
2.4.2 Éléments de Conception	6
2.4.3 Spécifications d'Interface	6
2.5 Vue des Données	7
2.5.1 Préoccupations	7
2.5.2 Éléments de Conception	7
2.5.3 Schéma de Données	7
2.6 Vue des États Dynamiques	8
2.6.1 Préoccupations	8
2.6.2 Éléments de Conception	8
2.6.3 Diagramme d'État - Séance d'Entraînement	8
3 Architecture du Système	9
3.1 Architecture Générale	9
3.2 Composants Principaux	9
3.2.1 Application Mobile	9
3.2.2 Services Backend	9

4 Conception Détailée	10
4.1 Module d'Authentification	10
4.1.1 Classes Principales	10
4.2 Module d'Entraînement	10
4.2.1 Classes Principales	10
4.3 Module Nutritionnel	11
4.3.1 Classes Principales	11
5 Exigences Non-Fonctionnelles	12
5.1 Performance	12
5.2 Sécurité	12
5.3 Fiabilité	12
6 Contraintes et Hypothèses	13
6.1 Contraintes Techniques	13
6.2 Hypothèses	13
Conclusion	14
Bibliographie	15
A Annexe A : Diagrammes Techniques	16
A.1 Diagramme de Séquence - Authentification	16
A.2 Diagramme d'État - Programme d'Entraînement	16
A.3 Diagramme de Déploiement	16
B Annexe B : Standards de Code	17
B.1 Conventions de Nommage	17
B.2 Guidelines Sécurité	17

Introduction

Objet

Ce document décrit la conception détaillée de l'application mobile BasedFit selon le standard IEEE 1016-2009. Il sert de référence pour le développement et la maintenance du système.

Portée

Le document couvre l'ensemble des aspects de conception de l'application BasedFit, incluant l'architecture, les composants, les interfaces et la gestion des données.

Conformité

Ce SDD est conforme aux exigences du standard IEEE 1016-2009 et organise l'information selon les vues de conception spécifiées.

Chapitre 1

Modèle Conceptuel

1.1 Sujet de Conception

L'application mobile BasedFit est une plateforme complète de suivi sportif et nutritionnel destinée aux utilisateurs de salles de sport.

1.2 Parties Prenantes

- Utilisateurs finaux (sportifs)
- Gérants de salles de sport
- Développeurs
- Administrateurs système

1.3 Préoccupations de Conception

- Facilité d'utilisation et intuitivité
- Performance et réactivité
- Sécurité des données personnelles
- Personnalisation des contenus
- Interaction sociale
- Gamification et motivation

Chapitre 2

Vues de Conception

2.1 Vue de Contexte

2.1.1 Préoccupations

Services fournis par le système, utilisateurs et interactions avec l'environnement externe.

2.1.2 Éléments de Conception

- **Acteurs** : Utilisateurs, Gérants, Systèmes externes (Google Fit, Apple Health)
- **Services** : Suivi d'entraînement, Nutrition, Communauté, Gamification
- **Flux** : Données de santé, Contenu sportif, Interactions sociales

2.1.3 Langages de Conception

Diagramme de cas d'utilisation UML, diagramme de contexte.

2.2 Vue de Composition

2.2.1 Préoccupations

Structure modulaire et organisation hiérarchique des composants.

2.2.2 Éléments de Conception

- **Entités** : Modules d'authentification, entraînement, nutrition, communauté
- **Relations** : Composition, dépendance, utilisation
- **Attributs** : Identification, type, fonction

2.2.3 Structure Hiérarchique

```
1 BasedFit Application
2     Module Authentification
3     Module Entrainement
4         Gestion Programmes
5         Suivi Sances
6         Analyse Performance
7     Module Nutrition
8         Journal Alimentaire
9         Recommandations
10        Suivi Macros
```

```
11     Module Communaut
12         Forum
13         Messagerie
14         Classements
```

2.3 Vue Logique

2.3.1 Préoccupations

Structure statique des classes, interfaces et leurs relations.

2.3.2 Éléments de Conception

```
1 // Classes principales du domaine
2 class User {
3     - UUID userId
4     - String email
5     - PersonalData personalInfo
6     - List<Achievement> achievements
7     + createProfile()
8     + updatePreferences()
9 }
10
11 class WorkoutProgram {
12     - UUID programId
13     - ProgramType type
14     - List<Exercise> exercises
15     - FitnessGoals goals
16     + generateProgram()
17     + trackProgress()
18 }
19
20 class NutritionPlan {
21     - UUID planId
22     - DailyGoals goals
23     - List<Meal> meals
24     + calculateNutrition()
25     + generateRecommendations()
26 }
```

2.4 Vue des Interfaces

2.4.1 Préoccupations

Définition des interfaces utilisateur et des APIs.

2.4.2 Éléments de Conception

- **Interfaces Utilisateur** : Écrans mobiles, navigation, formulaires
- **APIs** : Endpoints REST, formats de données, authentification
- **Contraintes** : Responsive design, accessibilité A2 français

2.4.3 Spécifications d'Interface

```

1 // Interface d'authentification
2 POST /api/v1/auth/login
3 Content-Type: application/json
4 {
5     "email": "user@example.com",
6     "password": "hashed_password"
7 }
8
9 // Interface de suivi d'entraînement
10 POST /api/v1/workouts/sessions
11 {
12     "userId": "uuid",
13     "exercises": [
14         {
15             "exerciseId": "uuid",
16             "sets": [
17                 {"reps": 10, "weight": 50}
18             ]
19         }
20     ]
21 }
```

2.5 Vue des Données

2.5.1 Préoccupations

Structure des données persistantes et stratégies d'accès.

2.5.2 Éléments de Conception

- **Entités** : Utilisateurs, Séances, Aliments, Succès
- **Relations** : Associations, hiérarchies, contraintes
- **Attributs** : Types de données, contraintes d'intégrité

2.5.3 Schéma de Données

```

1 -- Table des utilisateurs
2 CREATE TABLE users (
3     user_id UUID PRIMARY KEY,
4     email VARCHAR(255) UNIQUE NOT NULL,
5     password_hash VARCHAR(255) NOT NULL,
6     first_name VARCHAR(100),
7     last_name VARCHAR(100),
8     birth_date DATE,
9     created_at TIMESTAMP DEFAULT NOW()
10 );
11
12 -- Table des séances d'entraînement
13 CREATE TABLE workout_sessions (
14     session_id UUID PRIMARY KEY,
15     user_id UUID REFERENCES users(user_id),
16     session_date DATE NOT NULL,
17     duration_minutes INTEGER,
18     total_calories INTEGER
19 );
```

2.6 Vue des États Dynamiques

2.6.1 Préoccupations

Comportement dynamique du système et transitions d'état.

2.6.2 Éléments de Conception

- **États** : Initial, En cours, Terminé, Erreur
- **Transitions** : Événements, conditions, actions
- **Contraintes** : Gardes, invariants

2.6.3 Diagramme d'État - Séance d'Entraînement

```
1 [Non d marr e]
2   (d marrer) [En cours]
3   (mettre en pause) [En pause]
4   (reprendre) [En cours]
5   (terminer) [Termin e]
6   (analyser) [Analys e]
```

Chapitre 3

Architecture du Système

3.1 Architecture Générale

Architecture en couches avec séparation des préoccupations :

- **Présentation** : Interface mobile (React Native)
- **Métier** : Logique applicative et règles de gestion
- **Données** : Persistance et gestion des données

3.2 Composants Principaux

3.2.1 Application Mobile

- Framework : React Native avec TypeScript
- Navigation : React Navigation
- État : Redux Toolkit
- Stockage local : AsyncStorage

3.2.2 Services Backend

- API Gateway : Node.js/Express
- Authentification : Service JWT/OAuth2
- Entraînement : Service Python/FastAPI
- Nutrition : Service Java/Spring Boot
- Base de données : PostgreSQL

Chapitre 4

Conception Détailée

4.1 Module d'Authentification

4.1.1 Classes Principales

```
1 class AuthManager {
2     - UserRepository userRepo
3     - TokenService tokenService
4     - SecurityValidator validator
5
6     + login(credentials)
7     + register(userData)
8     + logout(userId)
9     + resetPassword(email)
10 }
11
12 class UserSession {
13     - UUID sessionId
14     - UUID userId
15     - DateTime expiry
16     - String deviceInfo
17     + isValid()
18     + refresh()
19     + terminate()
20 }
```

4.2 Module d'Entraînement

4.2.1 Classes Principales

```
1 class WorkoutGenerator {
2     - ExerciseDatabase exerciseDB
3     - AlgorithmFactory algorithms
4
5     + generateProgram(userProfile, goals)
6     + adaptProgram(progressData)
7     + suggestExercises(preferences)
8 }
9
10 class ExerciseTracker {
11     - SensorManager sensors
12     - FormAnalyzer formAnalyzer
13     - PerformanceCalculator calculator
14 }
```

```
15     + startTracking(exercise)
16     + recordRepetition(data)
17     + analyzeForm(videoFrame)
18     + calculateMetrics()
19 }
```

4.3 Module Nutritionnel

4.3.1 Classes Principales

```
1 class NutritionAnalyzer {
2     - FoodDatabase foodDB
3     - GoalCalculator goalCalc
4     - RecommendationEngine recommender
5
6     + logFoodConsumption(entry)
7     + calculateDailyTotals(date)
8     + generateRecommendations(profile)
9     + detectPatterns(history)
10 }
11
12 class MealPlanner {
13     - RecipeDatabase recipes
14     - PreferenceMatcher matcher
15     - NutritionValidator validator
16
17     + generateMealPlan(constraints)
18     + adjustPortions(plan, factors)
19     + substituteIngredients(meal, allergies)
20 }
```

Chapitre 5

Exigences Non-Fonctionnelles

5.1 Performance

- Temps de réponse interface : < 100ms
- Génération de programme : < 3 secondes
- Recherche d'exercices : < 1 seconde
- Capacité : 10,000 utilisateurs simultanés

5.2 Sécurité

- Authentification : 2FA optionnelle
- Chiffrement : AES-256 pour données sensibles
- Communications : TLS 1.3 obligatoire
- Conformité : RGPD pour données santé

5.3 Fiabilité

- Disponibilité : 99.5% uptime
- Récupération : Backup automatique quotidien
- Tolérance aux pannes : Dégradation gracieuse

Chapitre 6

Contraintes et Hypothèses

6.1 Contraintes Techniques

- Support Android 7.0+ et iOS 11.0+
- Taille application < 100MB
- Usage données mobile optimisé
- Consommation batterie limitée

6.2 Hypothèses

- Connexion internet intermittente acceptable
- Utilisateurs familiers avec applications mobiles
- APIs externes disponibles (Google Fit, Apple Health)
- Données de santé accessibles via standards

Conclusion

Ce document de conception fournit une base solide pour le développement de l'application BasedFit. Il respecte le standard IEEE 1016-2009 et couvre l'ensemble des aspects techniques nécessaires à la réalisation du projet.

Les prochaines étapes incluent la revue de conception, l'ajustement des spécifications basé sur les retours, et le début de l'implémentation selon le plan défini.

Bibliographie

- [1] IEEE Computer Society, *IEEE Standard for Information Technology—Systems Design—Software Design Descriptions*, IEEE Std 1016-2009, 2009.
- [2] Équipe BasedFit, *Spécification des Exigences Logiciel - BasedFit*, Version 1.3, 2025.
- [3] React Native Documentation, <https://reactnative.dev/docs/getting-started>
- [4] PostgreSQL Documentation, <https://www.postgresql.org/docs/>

Annexe A

Annexe A : Diagrammes Techniques

A.1 Diagramme de Séquence - Authentification

Description du flux d'authentification utilisateur.

A.2 Diagramme d'État - Programme d'Entraînement

États et transitions d'un programme sportif.

A.3 Diagramme de Déploiement

Architecture de déploiement de l'application.

Annexe B

Annexe B : Standards de Code

B.1 Conventions de Nommage

- Variables : camelCase
- Classes : PascalCase
- Constantes : UPPER_CASE
- Fichiers : kebab-case

B.2 Guidelines Sécurité

- Validation des entrées utilisateur
- Chiffrement des données sensibles
- Gestion sécurisée des tokens
- Audit des actions critiques