### 

| **TECHNIQUES DE DEVELOPPEMENT DES SYSTEMES** |
| --- |

### 

**Document Technique**

## 

[**1. Introduction**](#_xceonnm9l8uw) **1**

[**2. Conception de l'Architecture**](#_8yhwc142dkqi) **2**

[2.1. Architecture Globale](#_hu2hx8oladpl) 2

[2.1.1. Diagramme d'Architecture](#_972n074wrdtk) 2

[2.2. Choix Technologiques](#_vjqogzvmnyld) 2

[**3. Développement des Modules**](#_p89ea7ok5rsa) **3**

[3.1. Module d'Authentification](#_45wr2342qvea) 3

[3.1.1. Fonctionnalités](#_hehqcf2bm5kk) 3

[3.1.2. Implémentation](#_ss9waq868c1m) 3

[3.2. Module de Suivi des Séances](#_tz62kp2x0x7k) 3

[3.2.1. Fonctionnalités](#_5haznv21t532) 4

[3.2.2. Implémentation](#_2ox9yjimjx59) 4

[3.3. Module de Gestion des Objectifs](#_r0jo8bd7ux16) 4

[3.3.1. Fonctionnalités](#_fp6wwduabmlx) 4

[3.3.2. Implémentation](#_uq6x012g5cvf) 5

[3.4. Module de Visualisation des Statistiques](#_cx4kts4spcy9) 5

[3.4.1. Fonctionnalités](#_5imcvtqaqsxz) 5

[3.4.2. Implémentation](#_o7kbv22tr4s) 5

[3.5. Module de Notifications](#_wf23dy1u30ii) 6

[3.5.1. Fonctionnalités](#_dyhpi9vs0ysy) 6

[3.5.2. Implémentation](#_vs6j1edwp1uq) 6

[**4. Configuration et Déploiement**](#_5xp0sq8mynp) **6**

[4.1. Environnement de Développement](#_dxybhqyiwpwn) 6

[4.2. Déploiement](#_y5j6nmk7msg1) 7

[4.3. Maintenance et Mise à Jour](#_q3akg7355oy) 7

[**5. Tests et Validation**](#_z5xskbltrk9u) **7**

[5.1. Plan de Tests](#_g9yvasfk6lnn) 7

[5.2. Validation](#_y7ydrs2ocgcw) 7

[**6. Conclusion**](#_vhaee5ol6twa) **8**

## 

## **1. Introduction**

Cette documentation technique est destinée aux développeurs et aux membres de l'équipe technique impliqués dans le développement et la maintenance de l'application Web et mobile de suivi de fitness. Elle détaille les choix technologiques, les configurations spécifiques, les modules clés, et les instructions pour le déploiement de l'application.

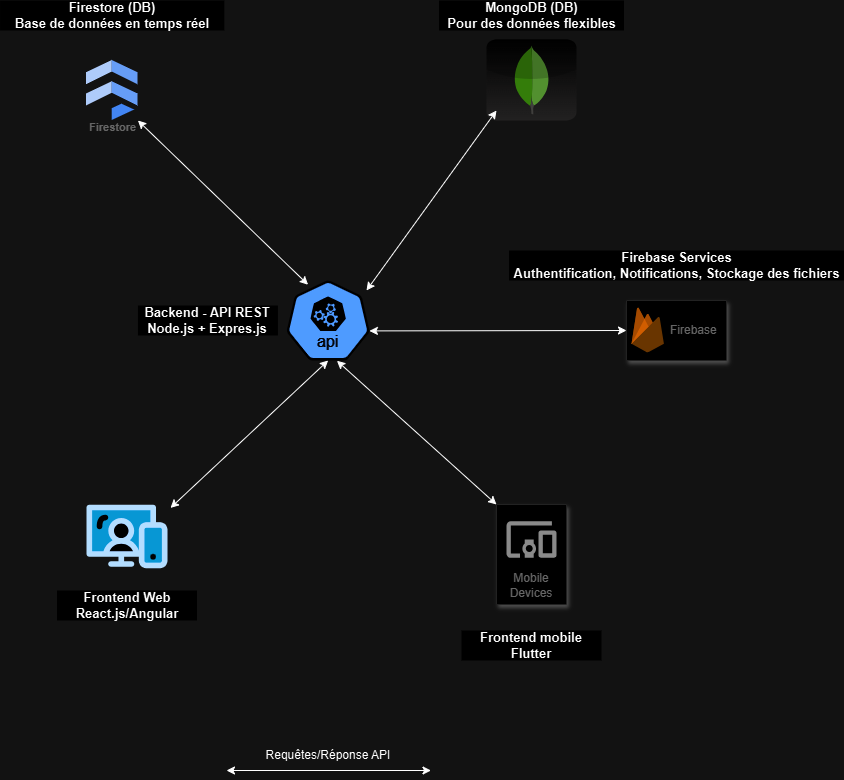
## **2. Conception de l'Architecture**

### **2.1. Architecture Globale**

L'architecture de l'application est divisée en plusieurs composants principaux :

* **Frontend Mobile (Flutter)**
* **Frontend Web (React.js/Angular)**
* **Backend (Node.js avec Express.js)**
* **Bases de Données (Firestore et MongoDB)**
* **Services Firebase**

#### **2.1.1. Diagramme d'Architecture**



### **2.2. Choix Technologiques**

* **Frontend Mobile :** Développement avec Flutter pour une compatibilité multiplateforme (iOS et Android).
* **Frontend Web :** Utilisation de React.js pour une application dynamique et rapide, ou Angular pour une structure plus rigide et organisée.
* **Backend :** Node.js avec Express.js pour la gestion de la logique métier et des API REST.
* **Bases de Données :** Firestore pour des données en temps réel et MongoDB pour les structures de données flexibles.
* **Services Supplémentaires :** Firebase pour l'authentification, les notifications, et le stockage.

## **3. Développement des Modules**

### **3.1. Module d'Authentification**

#### **3.1.1. Fonctionnalités**

* Inscription/Connexion via email et mot de passe.
* Authentification via Google et Facebook.

#### **3.1.2. Implémentation**

javascript

// Exemple de code d'authentification avec Firebase Authentication

import firebase from 'firebase/app';

import 'firebase/auth';

const signUpWithEmail = async (email, password) => {

try {

await firebase.auth().createUserWithEmailAndPassword(email, password);

// Autres configurations si nécessaire

} catch (error) {

console.error("Erreur d'inscription :", error);

}

};

* **Firebase Configuration:** Incluez la configuration Firebase dans votre projet.

### **3.2. Module de Suivi des Séances**

#### **3.2.1. Fonctionnalités**

* Enregistrement manuel des types d'exercice, de la durée et des calories brûlées.
* Suivi automatique via GPS et accéléromètre.

#### **3.2.2. Implémentation**

dart

// Exemple de code pour capturer les données GPS en Flutter

import 'package:geolocator/geolocator.dart';

Future<Position> \_getCurrentLocation() async {

bool serviceEnabled;

LocationPermission permission;

serviceEnabled = await Geolocator.isLocationServiceEnabled();

if (!serviceEnabled) {

return Future.error('Le service de localisation est désactivé.');

}

permission = await Geolocator.checkPermission();

if (permission == LocationPermission.denied) {

permission = await Geolocator.requestPermission();

if (permission == LocationPermission.denied) {

return Future.error('Les autorisations de localisation sont refusées.');

}

}

return await Geolocator.getCurrentPosition(desiredAccuracy: LocationAccuracy.high);

}

* **Intégration des capteurs :** Utilisation de l'API Flutter pour intégrer les capteurs tels que le GPS et l'accéléromètre.

### **3.3. Module de Gestion des Objectifs**

#### **3.3.1. Fonctionnalités**

* Définition des objectifs personnalisés.
* Notification des progrès.

#### **3.3.2. Implémentation**

javascript

// Exemple de code pour la gestion des objectifs

const addGoal = async (userId, goal) => {

try {

await firebase.firestore().collection('goals').add({

userId: userId,

goal: goal,

createdAt: firebase.firestore.FieldValue.serverTimestamp(),

});

// Envoyer une notification pour le rappel de l'objectif

} catch (error) {

console.error("Erreur lors de l'ajout d'un objectif :", error);

}

};

### **3.4. Module de Visualisation des Statistiques**

#### **3.4.1. Fonctionnalités**

* Création de tableaux de bord dynamiques.
* Comparaison des performances actuelles avec les périodes précédentes.

#### **3.4.2. Implémentation**

javascript

// Exemple de code pour générer des graphiques avec Chart.js

import Chart from 'chart.js';

const ctx = document.getElementById('myChart').getContext('2d');

const myChart = new Chart(ctx, {

type: 'line',

data: {

labels: ['Janvier', 'Février', 'Mars', 'Avril', 'Mai', 'Juin'],

datasets: [{

label: 'Calories brûlées',

data: [65, 59, 80, 81, 56, 55],

fill: false,

borderColor: 'rgba(75, 192, 192, 1)',

tension: 0.1

}]

},

});

### **3.5. Module de Notifications**

#### **3.5.1. Fonctionnalités**

* Notifications push pour rappeler les objectifs quotidiens et les séances planifiées.

#### **3.5.2. Implémentation**

javascript

// Exemple de code pour envoyer des notifications avec Firebase Cloud Messaging

import firebase from 'firebase/app';

import 'firebase/messaging';

const sendNotification = async (message) => {

try {

const messaging = firebase.messaging();

await messaging.send(message);

} catch (error) {

console.error("Erreur lors de l'envoi de la notification :", error);

}

};

## **4. Configuration et Déploiement**

### **4.1. Environnement de Développement**

* **Installation des Dépendances :** Utilisation de npm pour installer les dépendances nécessaires.

bash

npm install

* **Configurations Initiales :** Configuration des variables d'environnement pour l'accès à Firebase, MongoDB, etc.

### **4.2. Déploiement**

* **CI/CD Pipeline:** Configuration d'un pipeline de déploiement continu avec des outils comme Jenkins ou GitHub Actions.
* **Hébergement :** Utilisation de services cloud comme Firebase Hosting ou AWS S3 pour héberger le frontend, et déploiement du backend sur une instance Node.js via des services comme Heroku ou AWS Elastic Beanstalk.

### **4.3. Maintenance et Mise à Jour**

* **Surveillance :** Mise en place de services de monitoring comme New Relic ou Google Analytics pour surveiller la performance de l'application.
* **Gestion des Versions :** Utilisation de Git pour le contrôle de version, avec des branches spécifiques pour les développements, les tests, et la production.

## **5. Tests et Validation**

### **5.1. Plan de Tests**

* **Unit Testing :** Utilisation de frameworks comme Jest (pour Reactjs) ou Mocha (pour Node.js) pour les tests unitaires.
* **Integration Testing :** Tests des interactions entre les modules, avec des outils comme Postman pour les API.
* **UI/UX Testing :** Tests manuels et automatisés pour valider la navigation et l'interface utilisateur.

### **5.2. Validation**

* **Critères d'Acceptation :** Liste des fonctionnalités devant être validées par les tests pour garantir que l'application répond aux exigences fonctionnelles et non fonctionnelles.

## **6. Conclusion**

Ce document technique fournit les détails nécessaires pour le développement, la configuration et le déploiement de l'application de suivi de fitness. Il est essentiel de suivre les bonnes pratiques décrites pour assurer la qualité et la maintenabilité du code.