### 

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

### 

**Rapport des Test**

[**Introduction 3**](#_8apcc8otz84s)

[**Tests de l’API 5**](#_aswt9a8f76qn)

[Plan de Tests et Stratégie de Validation 5](#_m9681hijqibu)

[Résultats des Tests 6](#_w029co8ncl9t)

[Anomalies et Actions Correctives 7](#_jj3w5y89f17y)

[**Conclusion 7**](#_flz1jlu94hl9)

[Front-End Mobile : Module d'Authentification 8](#_79b3rj5b77up)

[Plan de Tests et Stratégie de Validation 8](#_i3kq3runc08m)

[Résultats des Tests 9](#_twapnhgqb1he)

[Anomalies et Actions Correctives 10](#_kljgpfmj3nmj)

[Conclusion 10](#_gr2gfw8znnvn)

[Front-End Mobile: Module de Suivi des Séances 11](#_4ke5nnqk69u8)

[Objectifs des Tests 11](#_cjtf004103i7)

[Plan de Tests et Stratégie de Validation 11](#_3dqgqfj6ujjv)

[Résultats des Tests 12](#_xl2zilkevu3p)

[Anomalies et Actions Correctives 12](#_xt6vl88eoyer)

[Conclusion 12](#_ulb7zu44gpsp)

[Front-End Mobile: Module de Définition des Objectifs 12](#_dterxiprw98c)

[Objectifs des Tests 13](#_a5rjp1ue9sot)

[Plan de Tests et Stratégie de Validation 13](#_hvbe1r8cv5e)

[Résultats des Tests 14](#_h0aax1xqnp96)

[Anomalies et Actions Correctives 14](#_kv6jsdweemy5)

[Conclusion 14](#_6snx30a5s5dn)

[Écran de Profil Utilisateur 14](#_gwnrqa57mrbr)

[Plan de Test 14](#_yo5wsklfmgmu)

[Stratégie de Test 14](#_5qaobvejlaui)

[Résultats des Tests 15](#_3c6jpp239lc0)

[Anomalies et Actions Correctives 15](#_nc3hknbxfayw)

[**Conclusion 15**](#_9tnggdo6bdif)

# **Introduction**

**Contexte du Projet**: L'application testée est un système complet de suivi des séances d'entraînement, destiné à fournir aux utilisateurs une solution intégrée pour enregistrer et analyser leurs activités physiques. Ce projet comprend plusieurs modules interconnectés, dont :

* **API** : Interface de communication entre le client et le serveur, permettant la gestion des données des utilisateurs, des séances d'entraînement, des objectifs, des notifications et des badges.
* **Base de Données** : Stockage des informations relatives aux utilisateurs, aux séances, aux objectifs, aux notifications, aux badges et aux statistiques de performance.
* **Interface Utilisateur** : Interfaces web et mobile permettant aux utilisateurs d'interagir avec l'application, d'enregistrer leurs séances, de suivre leurs objectifs et de consulter leurs statistiques.

**Architecture du Système**:

* **Frontend Mobile** :
  + Application mobile développée avec Flutter, interagissant avec le backend via des API REST.
* **Frontend Web** :
  + Application web développée avec React.js ou Angular, utilisant également des API REST pour communiquer avec le backend.
* **Backend** :
  + Serveur Node.js avec Express.js, fournissant une API REST pour la communication avec les frontends mobile et web.
* **Base de Données** :
  + **Firestore** : Base de données en temps réel pour la gestion des données dynamiques.
  + **MongoDB** : Base de données flexible pour des données structurées et non structurées.
* **Services Supplémentaires** :
  + **Firebase** : Utilisé pour l'authentification, les notifications et le stockage des fichiers.

**Objectif du Rapport de Tests**: Le rapport a pour but de documenter le processus de test de l'application, en mettant initialement l'accent sur l'API. Il vise à garantir que chaque composant fonctionne conformément aux spécifications et répond aux besoins des utilisateurs. Les points suivants seront couverts :

* **Plan de Tests et Stratégie de Validation** : Méthodes et outils utilisés pour tester l'application, incluant les tests unitaires, d'intégration, fonctionnels et de performance.
* **Résultats des Tests** : Résultats des tests effectués sur l'API, incluant les réussites et les anomalies détectées.
* **Anomalies et Actions Correctives** : Problèmes rencontrés, corrections apportées et prochaines étapes pour résoudre les anomalies.
* **Conclusion** : Évaluation de la qualité globale de l'application basée sur les résultats des tests et les actions correctives entreprises.

**Portée du Rapport**: Le rapport sera mis à jour au fur et à mesure du développement et des tests supplémentaires. Les sections suivantes couvriront les différents aspects du projet :

* **Tests de l'API** : Validation des routes et des fonctionnalités de l'API pour chaque modèle de données (User, Seance, Notification, Badge, Objectif, Statistique).
* **Tests de l'Interface Utilisateur (UI)** : Analyse des performances et de la convivialité de l'interface utilisateur, prévue pour être ajoutée après le développement de l'UI.

### **Tests de l’API**

#### **Plan de Tests et Stratégie de Validation**

**Méthodologie de Test**:

* **Approche Agile** : Les tests de l'API ont été réalisés en parallèle avec le développement de l'application, suivant une approche Agile. Les tests ont été menés de manière itérative pour identifier et corriger les problèmes au fur et à mesure du développement.
* **Types de Tests** :
  + **Tests Unitaires** : Validation des fonctions et méthodes isolées dans les contrôleurs pour vérifier leur bon fonctionnement.
  + **Tests d'Intégration** : Validation des interactions entre les différents modules de l'application, notamment les interactions entre les modèles et les contrôleurs.
  + **Tests Fonctionnels** : Vérification des principales fonctionnalités de l'API, telles que la création, la mise à jour, la récupération et la suppression des entités.
  + **Tests de Performance** : Évaluation des performances de l'API sous diverses charges pour garantir sa robustesse et réactivité.
  + **Tests de Sécurité** : Identification des vulnérabilités potentielles pour assurer la protection des données des utilisateurs.

**Outils Utilisés** :

* **Jest** : Pour les tests unitaires afin de valider les fonctions et méthodes des contrôleurs.
* **Postman** : Pour les tests d'API afin de vérifier le bon fonctionnement des routes et la validité des réponses.
* **Docker** : Pour créer des environnements de test isolés et reproductibles.

**Environnement de Test** :

* **Environnement de Développement** : Utilisé pour les tests unitaires et les premiers tests d'intégration.
* **Environnement de Staging** : Reproduction de l'environnement de production pour des tests plus exhaustifs avant le déploiement.
* **Environnement de Production** : Surveillance et tests de performance en conditions réelles après le déploiement.

#### **Résultats des Tests**

**Tests Unitaires** : Les tests unitaires ont validé les fonctions des contrôleurs et modèles de données. Les tests ont confirmé que les méthodes isolées fonctionnent correctement.

**Tests d'Intégration** : Les tests d'intégration ont vérifié les interactions entre les différents modules. Les résultats principaux sont :

* **Seance** : Les routes de gestion des séances (création, mise à jour, récupération, suppression) sont fonctionnelles.
* **User** : Les opérations de gestion des utilisateurs se comportent comme prévu.
* **Badge** : Les fonctionnalités de gestion des badges fonctionnent comme attendu.
* **Notification** : Les opérations sur les notifications (création, récupération, mise à jour, suppression) sont correctes.
* **Objectif** : La définition et le suivi des objectifs sont conformes aux attentes.

**Tests Fonctionnels** : Les tests fonctionnels ont couvert les scénarios principaux :

* **Création de Ressources** : Les utilisateurs, séances, objectifs, badges et notifications sont créés correctement.
* **Mise à Jour de Ressources** : Les mises à jour des ressources se déroulent sans erreurs.
* **Récupération de Ressources** : Les données sont récupérées correctement selon les critères spécifiés.
* **Suppression de Ressources** : Les suppressions se réalisent comme prévu, avec les vérifications nécessaires pour éviter les suppressions accidentelles.

**Tests de Performance** : L'API a démontré sa capacité à gérer une charge utilisateur modérée sans dégradations significatives des performances.

**Tests de Sécurité** : Des vulnérabilités mineures ont été identifiées et corrigées pour garantir la sécurité des données des utilisateurs.

#### **Anomalies et Actions Correctives**

**Liste des Anomalies** :

* **Bug 001** : Problème de validation des coordonnées GPS dans le modèle Seance.
  + *Action Corrective* : Mise à jour de la logique de validation pour accepter les coordonnées au format valide.
* **Bug 002** : Erreur 500 lors de la mise à jour d'un utilisateur avec des champs non modifiables.
  + *Action Corrective* : Correction de la logique de mise à jour dans le contrôleur User.
* **Bug 003** : Dépassement des limites de taille de requête pour les données de Notification.
  + *Action Corrective* : Ajustement des limites de taille et amélioration de la gestion des erreurs pour les requêtes volumineuses.

**Actions Correctives** : Chaque anomalie a été corrigée dans le code, suivie de nouveaux tests pour valider les corrections apportées.

### **Conclusion**

**Évaluation Globale** : L'API a passé la majorité des tests, avec quelques anomalies mineures corrigées. Elle est jugée stable et conforme aux exigences fonctionnelles spécifiées.

### **Front-End Mobile : Module d'Authentification**

Le module d'authentification permet aux utilisateurs de s'inscrire et de se connecter à l'application via email/mot de passe, Google, ou Facebook. Une fois inscrites, les informations des nouveaux utilisateurs sont envoyées à l'API pour être enregistrées dans la base de données. Ce rapport de test détaille les tests effectués pour valider le bon fonctionnement de ce module.

**Objectifs des Tests**

* **Fonctionnalité :** Vérifier que chaque méthode d'authentification (email/mot de passe, Google, Facebook) fonctionne correctement.
* **Sécurité :** Assurer la sécurité des processus d'inscription et de connexion, notamment la protection des mots de passe et l'intégrité des données.
* **Intégration :** Valider l'interaction entre le module d'authentification et l'API pour l'enregistrement des nouveaux utilisateurs.
* **Performance :** Mesurer les temps de réponse lors de l'inscription et de la connexion des utilisateurs.

#### **Plan de Tests et Stratégie de Validation**

**Méthodologie de Test**

Les tests ont été effectués en suivant une approche Agile, itérative, avec des validations continues tout au long du développement.

**Types de Tests**

* **Tests Unitaires :**
  + Validation des méthodes de création d'utilisateur, de connexion, et de validation des entrées.
  + Vérification de l'intégrité du processus de chiffrement des mots de passe.
* **Tests d'Intégration :**
  + Test de l'envoi des données des nouveaux inscrits à l'API.
  + Vérification de l'intégration des authentifications via Google et Facebook.
* **Tests Fonctionnels :**
  + Test des scénarios complets d'inscription et de connexion.
  + Vérification des messages d'erreur en cas de données incorrectes ou manquantes.
* **Tests de Sécurité :**
  + Test de la sécurité des données lors des échanges entre le frontend et le backend.
  + Vérification de la protection contre les attaques courantes (injections SQL, XSS, etc.).
* **Tests de Performance :**
  + Mesure des temps de réponse des API d'authentification sous diverses conditions.

**Outils Utilisés**

* **Jest :** Pour les tests unitaires des fonctions et méthodes de l'authentification.
* **Postman :** Pour tester les différentes routes d'authentification et valider les réponses.
* **OWASP ZAP :** Pour l'analyse des vulnérabilités de sécurité.
* **JMeter :** Pour les tests de performance et la simulation de charge utilisateur.

**Environnement de Test**

* **Environnement de Développement :** Utilisé pour les tests initiaux.
* **Environnement de Staging :** Utilisé pour des tests plus exhaustifs avant déploiement en production.

#### **Résultats des Tests**

**Tests Unitaires :**

* **Inscription par Email/Mot de Passe :** Test réussi, les utilisateurs sont correctement créés avec des mots de passe chiffrés.
* **Connexion par Email/Mot de Passe :** Test réussi, l'utilisateur peut se connecter avec les identifiants corrects, et les erreurs sont bien gérées.
* **Connexion via Google et Facebook :** Test réussi, les utilisateurs peuvent se connecter en utilisant leurs comptes Google ou Facebook.

**Tests d'Intégration :**

* **Enregistrement des Nouveaux Utilisateurs :** Test réussi, les informations des nouveaux inscrits sont correctement envoyées et stockées dans la base de données via l'API.
* **Interaction avec l'API :** Test réussi, l'authentification via Google et Facebook fonctionne sans conflits avec les autres parties du système.

**Tests Fonctionnels :**

* **Scénario d'Inscription :** Test réussi, toutes les routes d'inscription fonctionnent comme prévu.
* **Scénario de connexion :** Test réussi, la connexion via toutes les méthodes est opérationnelle.
* **Gestion des Erreurs :** Test réussi, les messages d'erreur s'affichent correctement en cas de données incorrectes.

**Tests de Sécurité :**

* **Chiffrement des Mots de Passe :** Test réussi, les mots de passe sont stockés de manière sécurisée.
* **Protection contre les attaques :** Test réussi, aucune vulnérabilité majeure n'a été détectée.

**Tests de Performance :**

* **Temps de Réponse :** Les temps de réponse des routes d'authentification sont jugés acceptables sous une charge modérée.

#### **Anomalies et Actions Correctives**

* **Anomalie 001 :** Problème lors de l'authentification via Facebook sur certains appareils.
  + **Action Corrective :** Mise à jour du SDK Facebook pour corriger la compatibilité.
* **Anomalie 002 :** Délais excessifs lors de la connexion avec Google sous certaines conditions réseau.
  + **Action Corrective :** Optimisation des appels API à Google.

#### **Conclusion**

Le module d'authentification a passé la majorité des tests avec succès. Quelques anomalies mineures ont été détectées et corrigées. Le module est jugé stable et conforme aux exigences fonctionnelles spécifiées. La sécurité des données utilisateurs est assurée, et l'intégration avec les autres parties du système est réussie.

### **Front-End Mobile: Module de Suivi des Séances**

Le module de suivi des séances permet non seulement d'enregistrer des séances manuelles et automatiques mais aussi de consulter l'historique des séances passées. Cette section couvre les tests effectués pour valider ces fonctionnalités.

#### **Objectifs des Tests**

* **Fonctionnalité** :
  + S'assurer que les séances sont correctement enregistrées dans Firestore.
  + Vérifier que l'utilisateur peut consulter l'historique de ses séances avec précision.
* **Performance** :
  + Évaluer la rapidité de l'enregistrement des données et de l'affichage de l'historique, même avec un grand nombre de séances.
* **Expérience Utilisateur** :
  + Vérifier que l'interface d'affichage de l'historique est intuitive et que les données sont présentées de manière claire.

#### **Plan de Tests et Stratégie de Validation**

**Méthodologie de Test**

* Tests axés sur la validation des fonctionnalités principales (enregistrement et consultation de l'historique), la performance en termes de temps de réponse, et l'expérience utilisateur.

**Types de Tests**

* **Tests Unitaires** :
  + Vérification des fonctions de création et de récupération des données de séance dans Firestore.
* **Tests d'Intégration** :
  + Test de l'intégration entre les différents composants du module de suivi (enregistrement de séance, affichage de l'historique).
  + Vérification de la cohérence entre les données stockées et celles affichées.
* **Tests Fonctionnels** :
  + Validation de scénarios complets où un utilisateur enregistre une séance puis consulte son historique.
  + Test de la fonctionnalité de filtrage et de tri des séances dans l'historique.
  + Vérification de l'affichage des détails de chaque séance (type, durée, calories brûlées, etc.).

#### **Résultats des Tests**

* **Enregistrement de Séances** : Les tests montrent que les séances sont enregistrées correctement dans Firestore, avec toutes les données associées.
* **Affichage de l'Historique** : Les tests confirment que l'utilisateur peut consulter son historique sans anomalies. Les séances sont affichées avec les détails corrects, et l'interface est réactive même avec un grand nombre de séances.
* **Performance** : L'affichage de l'historique est rapide, même avec plusieurs dizaines de séances enregistrées. Les opérations de filtrage et de tri sont effectuées sans latence perceptible.

#### **Anomalies et Actions Correctives**

* **Anomalie Mineure** : Une légère latence a été observée lors du filtrage des séances sur certains appareils plus anciens.
  + **Action Corrective** : Optimisation de la requête de récupération des données dans Firestore pour réduire le temps de réponse.

#### **Conclusion**

Le module de suivi des séances, y compris la fonctionnalité d'affichage de l'historique, a passé les tests avec succès. La fonctionnalité est stable, et l'expérience utilisateur est satisfaisante.

### **Front-End Mobile: Module de Définition des Objectifs**

Le module de définition des objectifs permet aux utilisateurs de définir, suivre et modifier leurs objectifs de fitness au sein de l'application. Cette section couvre les tests effectués pour valider ces fonctionnalités.

#### **Objectifs des Tests**

* **Fonctionnalité** :
  + S'assurer que les utilisateurs peuvent définir de nouveaux objectifs avec les paramètres appropriés (type d'objectif, valeur cible, date de début et de fin, etc.).
  + Vérifier que les objectifs sont enregistrés correctement dans Firestore et synchronisés avec l'API.
  + S'assurer que les utilisateurs peuvent consulter, modifier, et supprimer des objectifs.
* **Performance** :
  + Évaluer la rapidité de l'enregistrement et de la récupération des objectifs, en particulier lors de l'utilisation de l'API et de Firestore simultanément.
* **Expérience Utilisateur** :
  + Vérifier que l'interface de définition des objectifs est intuitive, avec des champs et des icônes clairement identifiables.
  + Assurer que les utilisateurs reçoivent des notifications de progression et peuvent visualiser leur avancement de manière claire.

#### **Plan de Tests et Stratégie de Validation**

**Méthodologie de Test**

* Tests basés sur des scénarios d'utilisation courants, y compris la création, la modification, la suppression, et la consultation d'objectifs.

**Types de Tests**

* **Tests Unitaires** :
  + Validation des fonctions de création, modification, et suppression d'objectifs.
  + Test des calculs de progression et de validation des dates d'objectifs.
* **Tests d'Intégration** :
  + Validation de l'intégration entre l'interface utilisateur, Firestore, et l'API.
  + Test de la synchronisation des données entre Firestore et l'API pour les objectifs créés et modifiés.
* **Tests Fonctionnels** :
  + Vérification que les objectifs sont correctement enregistrés et affichés avec tous les détails.
  + Test de l'affichage de la progression des objectifs et des notifications de rappel.
  + Vérification du processus de modification et suppression d'un objectif existant.

#### **Résultats des Tests**

* **Création d'Objectifs** : Les tests montrent que les objectifs sont définis correctement et enregistrés dans Firestore avec tous les détails requis. L'API retourne l'identifiant (\_id) qui est correctement synchronisé avec Firestore.
* **Modification et Suppression d'Objectifs** : Les tests confirment que les utilisateurs peuvent modifier et supprimer des objectifs sans anomalies. Les modifications sont mises à jour instantanément dans Firestore et l'API.
* **Affichage et Suivi des Objectifs** : Les objectifs sont affichés avec précision, et la progression est calculée correctement en fonction des séances enregistrées. Les notifications de rappel sont envoyées aux utilisateurs comme prévu.
* **Performance** : Le module fonctionne de manière fluide, avec des temps de réponse rapides pour l'enregistrement et la récupération des données, même lorsque de nombreux objectifs sont définis.

#### **Anomalies et Actions Correctives**

* **Aucune anomalie majeure n'a été détectée** : Le module a fonctionné comme prévu lors de tous les tests.

#### **Conclusion**

Le module de définition des objectifs a passé les tests avec succès. Il est stable, performant, et offre une expérience utilisateur satisfaisante, permettant aux utilisateurs de gérer leurs objectifs de manière efficace.

## **Écran de Profil Utilisateur**

#### **Plan de Test**

L'écran de profil utilisateur permet d'afficher et de modifier les informations de l'utilisateur, avec une option de déconnexion.

#### **Stratégie de Test**

* **Tests fonctionnels** : Vérification de l'affichage et de la modification des informations utilisateur.
* **Tests de sécurité** : Vérification de la sécurité lors de la mise à jour des informations.
* **Tests d'interface utilisateur** : Vérification de la présentation des informations et de la facilité d'accès au bouton de déconnexion.

#### **Résultats des Tests**

* **Affichage des informations utilisateur** : Testé et fonctionnel. Les informations sont affichées correctement.
* **Modification des informations** : Testé et fonctionnel. Les informations sont mises à jour en temps réel.
* **Déconnexion** : Testé et fonctionnel. Le bouton de déconnexion fonctionne correctement.

#### **Anomalies et Actions Correctives**

* **Validation des données** : Des ajustements ont été faits pour renforcer la validation des données lors de la mise à jour des informations utilisateur.

# **Conclusion**

L'ensemble des tests réalisés sur l'API et le module d'authentification du front-end mobile a permis de valider la majorité des fonctionnalités et d'assurer la stabilité du système. Les quelques anomalies identifiées ont été corrigées rapidement, ce qui a permis de renforcer la fiabilité et la sécurité de l'application.

Sur le plan de l'API, les tests unitaires, d'intégration, fonctionnels, et de sécurité ont confirmé que les routes sont bien implémentées et répondent correctement aux requêtes, avec une performance acceptable sous une charge modérée. L'intégration avec la base de données Firestore et MongoDB a été réussie, garantissant une gestion efficace des données.

Concernant le front-end mobile, le module d'authentification a démontré sa capacité à gérer divers types de connexions (email/mot de passe, Google, Facebook) tout en assurant la sécurité des données utilisateur. L'interaction avec l'API pour l'enregistrement des utilisateurs s'est avérée fluide et sans anomalies majeures.

En conclusion, l'application, dans sa version testée, est considérée stable et prête pour un déploiement en production. Toutefois, une surveillance continue et des tests supplémentaires seront nécessaires à mesure que l'application évolue et que de nouvelles fonctionnalités sont ajoutées.