**A. Identification**

**A.1. Identification du document**

| **Document:** |  |
| --- | --- |
| **Project:** | FITNESS-APP |
| **Version du projet:** | **1.0** |
| **Version du document:** | **1** |
| **Sécurité du document:** | CONFIDENTIEL |
| **Date de création:** | 31/06/2024 |
| **Par:** | LAINÉ STÉPHANE |

**A.2. Historique des changements**

| **Qui?** | **Quand?** | **Quoi?** |
| --- | --- | --- |
| **LAINE STEPHANE** |  |  |
|  |  |  |

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### 

#### **1. Introduction**

Ce document décrit le plan de test pour l'application de suivi de fitness. L'objectif est de valider que toutes les fonctionnalités de l'application fonctionnent correctement, qu'elles répondent aux spécifications, et qu'elles offrent une expérience utilisateur optimale.

#### **2. Objectifs des Tests**

* Vérifier que l'application mobile et web répond aux spécifications fonctionnelles et non fonctionnelles.
* S'assurer que l'authentification et la gestion des utilisateurs sont sécurisées et fonctionnelles.
* Tester la précision et la fiabilité du suivi des activités physiques.
* Valider la bonne performance des interfaces utilisateur sous différentes conditions de charge.
* Confirmer que les notifications, les rappels et les objectifs sont correctement implémentés.
* Vérifier l'intégration avec les services tiers tels que Firebase pour l'authentification et les notifications.

#### **3. Portée des Tests**

Les tests couvriront les éléments suivants :

* **Fonctionnalités principales** : Authentification, gestion des utilisateurs, suivi des activités, définition et suivi des objectifs, visualisation des statistiques, et notifications.
* **Compatibilité** : Tests sur différents navigateurs pour l'application web (Chrome, Firefox, Safari) et sur plusieurs plateformes mobiles (Android, iOS).
* **Performance** : Tests de charge pour s'assurer que l'application supporte un grand nombre d'utilisateurs simultanément.
* **Sécurité** : Tests pour vérifier la protection des données utilisateur et la résistance aux attaques courantes (injections SQL, XSS, etc.).
* **UI/UX** : Tests de l'expérience utilisateur, y compris la réactivité, la navigation et l'attrait visuel.

#### **4. Environnement de Test**

Les tests seront effectués dans les environnements suivants :

* **Environnement de développement** : Machines locales avec des configurations standards.
* **Environnement de staging** : Serveurs proches de la production pour simuler des conditions réelles.
* **Environnement de production** : Tests finaux après le déploiement, avant l'ouverture au public.

#### **5. Types de Tests**

* **Tests unitaires** : Chaque composant du code sera testé isolément pour vérifier son bon fonctionnement.
* **Tests d'intégration** : Vérification de la communication entre les différents composants et services (frontend-backend, API REST).
* **Tests fonctionnels** : Validation que chaque fonctionnalité de l'application fonctionne conformément aux spécifications.
* **Tests de régression** : S'assurer que les nouvelles modifications n'introduisent pas de bugs dans les fonctionnalités existantes.
* **Tests de performance** : Mesurer la rapidité, la stabilité et la scalabilité de l'application sous diverses charges.
* **Tests de sécurité** : Valider que l'application est sécurisée contre les attaques et que les données des utilisateurs sont protégées.

#### **6. Critères d'Acceptation**

* **Fonctionnalité** : Toutes les fonctionnalités principales doivent fonctionner sans erreur.
* **Performance** : L'application doit rester réactive sous des charges utilisateur élevées.
* **Sécurité** : Aucune vulnérabilité majeure ne doit être présente.
* **Compatibilité** : L'application doit fonctionner correctement sur tous les navigateurs et appareils cibles.
* **UI/UX** : L'application doit offrir une expérience utilisateur intuitive et agréable.

#### **7. Planification des Tests**

* **Phase de tests unitaires** : Pendant le développement.
* **Phase de tests d'intégration** : Après l'implémentation des principales fonctionnalités.
* **Phase de tests fonctionnels** : Pendant et après la phase de développement.
* **Phase de tests de régression** : Avant chaque nouvelle version ou mise à jour.
* **Phase de tests de performance et sécurité** : Avant le déploiement en production.
* **Tests de validation finale** : Juste avant la mise en production.

#### **8. Rôles et Responsabilités**

* **Testeurs** : Responsable de la création, de l'exécution des tests, et de la documentation des résultats.
* **Développeurs** : Correction des bugs identifiés et ajustement du code en fonction des résultats des tests.
* **Chef de projet** : Supervision du processus de test et validation finale des résultats.

#### **9. Outils de Test**

* **Outils de gestion des tests** : Jira, TestRail.
* **Outils de tests unitaires** : Jest, Mocha (pour JavaScript), JUnit (pour Java).
* **Outils de tests de performance** : JMeter, LoadRunner.
* **Outils de tests de sécurité** : OWASP ZAP, Burp Suite.
* **Simulateurs et émulateurs** : Android Studio, Xcode.

#### **10. Rapports et Documentation**

Chaque phase de test sera documentée, et les résultats seront enregistrés dans des rapports détaillés. Ces rapports incluront les cas de tests, les résultats obtenus, les anomalies détectées, et les actions correctives prises.

Rapport de test

### **Rapport de Test pour l'Application de Suivi de Fitness**

#### **1. Introduction**

Ce rapport de test documente les résultats des tests effectués sur l'application de suivi de fitness. Les tests ont été menés pour s'assurer que l'application fonctionne conformément aux spécifications et offre une expérience utilisateur fluide et sécurisée.

#### **2. Objectif du Test**

L'objectif principal des tests était de valider :

* La conformité des fonctionnalités de l'application par rapport aux spécifications.
* La performance de l'application sous différentes charges.
* La sécurité des données utilisateur.
* La compatibilité de l'application avec divers appareils et navigateurs.
* La qualité de l'expérience utilisateur (UI/UX).

#### **3. Environnement de Test**

Les tests ont été effectués dans les environnements suivants :

* **Environnement de Développement** : Tests préliminaires sur des machines locales.
* **Environnement de Staging** : Tests avancés sur des serveurs similaires à l'environnement de production.
* **Navigateurs Testés** : Chrome, Firefox, Safari.
* **Appareils Mobiles Testés** : Android (versions 9 à 12), iOS (versions 14 à 16).

#### **4. Résumé des Tests Effectués**

| **Type de Test** | **Résultats** |
| --- | --- |
| **Tests Unitaires** | 95% des tests unitaires ont réussi. Les 5% restants concernent des bugs mineurs corrigés par la suite. |
| **Tests d'Intégration** | Les API REST entre le frontend et le backend ont fonctionné correctement, sans problèmes majeurs. |
| **Tests Fonctionnels** | Toutes les fonctionnalités principales ont été validées avec succès. Aucun problème critique identifié. |
| **Tests de Régression** | Aucun bug de régression détecté après l'ajout de nouvelles fonctionnalités. |
| **Tests de Performance** | L'application a maintenu une réponse rapide sous des charges allant jusqu'à 500 utilisateurs simultanés. |
| **Tests de Sécurité** | Aucune vulnérabilité majeure détectée. L'application est conforme aux standards de sécurité. |
| **Tests UI/UX** | L'interface utilisateur est jugée intuitive et réactive. Quelques améliorations mineures sont recommandées. |

#### **5. Détails des Résultats de Test**

##### **5.1 Tests Unitaires**

* **Objectif** : Vérifier le bon fonctionnement de chaque composant du code.
* **Résultats** :
  + Composants testés : 100% des composants critiques.
  + Taux de réussite : 95%.
  + Problèmes rencontrés : Bugs mineurs dans les calculs de statistiques, corrigés rapidement.

##### **5.2 Tests d'Intégration**

* **Objectif** : Valider la communication entre les composants backend et frontend via les API REST.
* **Résultats** :
  + API de suivi des activités physiques : Réponses correctes sous différentes conditions.
  + Authentification utilisateur : Fonctionne correctement avec des services tiers (Google, Facebook).
  + Problèmes rencontrés : Aucun.

##### **5.3 Tests Fonctionnels**

* **Objectif** : Valider les fonctionnalités principales de l'application.
* **Résultats** :
  + Authentification : Fonctionne sans problème, y compris la récupération de mot de passe.
  + Suivi des activités : Enregistrement manuel et automatique des séances d'entraînement sans erreurs.
  + Notifications : Notifications push envoyées comme prévu.
  + Problèmes rencontrés : Aucun problème critique, quelques ajustements d'UI suggérés.

##### **5.4 Tests de Régression**

* **Objectif** : S'assurer que les nouvelles fonctionnalités n'introduisent pas de bugs dans les fonctionnalités existantes.
* **Résultats** :
  + Aucun bug de régression détecté.
  + Fonctionnalités testées : Authentification, suivi des activités, objectifs de fitness.

##### **5.5 Tests de Performance**

* **Objectif** : Tester la stabilité et la réactivité de l'application sous différentes charges.
* **Résultats** :
  + Temps de réponse : Moins de 2 secondes pour 500 utilisateurs simultanés.
  + Dégradation des performances : Aucune dégradation notable jusqu'à 500 utilisateurs.
  + Problèmes rencontrés : Aucun.

##### **5.6 Tests de Sécurité**

* **Objectif** : Vérifier la sécurité des données utilisateur et la résistance aux attaques.
* **Résultats** :
  + Test d'injection SQL : Pas de vulnérabilité détectée.
  + Test XSS : Pas de vulnérabilité détectée.
  + Problèmes rencontrés : Aucun.

##### **5.7 Tests UI/UX**

* **Objectif** : Évaluer l'expérience utilisateur.
* **Résultats** :
  + Design UI : Réactif et moderne.
  + Navigation : Intuitive et fluide.
  + Problèmes rencontrés : Quelques suggestions pour améliorer la lisibilité des textes sur certains écrans.

#### **6. Conclusions et Recommandations**

* **Conclusions** : L'application répond aux exigences fonctionnelles et non fonctionnelles. Elle est prête pour le déploiement en production avec des performances optimales et une bonne sécurité.
* **Recommandations** :
  + Implémenter les améliorations suggérées pour l'UI/UX.
  + Continuer les tests de performance pour des charges supérieures à 500 utilisateurs.
  + Prévoir des tests de sécurité réguliers pour assurer la protection continue des données utilisateur.

#### **7. Annexes**

* Captures d'écran des tests.
* Logs des tests unitaires et d'intégration.
* Rapports détaillés des tests de performance et sécurité.