Équipe EOS Marketplace

# EOS Marketplace V1
# Plan de tests logiciels

**Version 1.0**

# Historique des révisions

*[Dans l'historique de révision, il est nécessaire d'écrire le nom du ou des auteurs ayant travaillé sur chaque version.]*

| Date | Version | Description | Auteur |
|---|---|---|---|
| aaaa-mm-jj | x.x | <Détails précis du travail effectué> | <Nom> |
| 2021-10-05 | 0.1 | Ajout des sections 1 à 3 | Guilhem Dubois |
| 2021-10-06 | 0.2 | Ajout des ressources et jalons | Guilhem Dubois |
| | | | |
| | | | |

# Table des matières

*[Mettre à jour la table des matières avant une remise.]*

# Plan de tests logiciels

## 1. Introduction

*[Décrire le contenu et l'organisation du document.]*

Ce document contient des informations afin de préparer la procédure de tests dans notre projet. La première section porte sur les exigences (telles que définies dans le SRS) qui doivent être testées ainsi que le type de test associé. La deuxième section présente chaque type de test pertinent et les outils utilisés pour les réaliser. La troisième section porte sur les ressources humaines et matérielles utiles dans la réalisation des tests. Finalement, les jalons de tests sont présentés.

## 2. Exigences à tester

*[La liste qui suit identifie ce qui sera testé (cas d'utilisation, exigences fonctionnelles et exigences non-fonctionnels).*

*Note: Cette section est évolutive. Elle vise ultimement à établir la traçabilité entre chaque exigence du SRS et leurs cas de test respectifs. Évidemment, lors de la rédaction initiale du PTL, il est normal de présenter uniquement une liste sommaire des exigences et de cas d'utilisation.]*

| **Exigences** | **Tests associés** |
|---|---|
| *3.1.1 [Essentiel] L'utilisateur doit pouvoir se connecter avec un compte existant.* | Tests de fonction<br>Tests de sécurité et de contrôle d'accès<br>Tests d'échec/récupération |
| *3.1.2 [Essentiel] L'utilisateur doit pouvoir s'enregistrer avec un compte existant Google ou Facebook.* | Tests d'intégrité des données |
| *3.1.5 [Essentiel] L'utilisateur doit pouvoir s'enregistrer avec un compte Anchor Wallet.* | Tests d'interface usager |
| *3.2.1 [Essentiel] L'utilisateur doit pouvoir créer une annonce pour un service.*<br>*3.2.2 [Essentiel] L'utilisateur doit pouvoir créer une annonce pour un bien.* | Tests de fonction<br>Tests d'interface usager<br>Tests de volume<br>Tests d'intégrité des données |
| *3.2.3 [Essentiel] L'utilisateur doit pouvoir voir une annonce.* | Tests d'interface usager |
| *3.3. Carte de Sherbrooke ou Montréal* | Tests de configuration<br>Tests de volume |
| *3.4.1 [Essentiel] L'application doit permettre aux utilisateurs d'afficher les informations publiques d'autres utilisateurs.* | Tests de sécurité et de contrôle d'accès |
| *3.4.2 [Essentiel] L'application doit afficher les* | Tests de sécurité et de contrôle d'accès |

| | |
|---|---|
| *informations privées de l'utilisateur.* | |
| *3.4.3 [Essentiel] L'utilisateur doit pouvoir accéder aux profils des autres utilisateurs de l'application.* | Tests d'interface usager |
| *3.4.4 [Essentiel] L'utilisateur doit pouvoir accéder à la liste de ses annonces créées.* | Tests d'interface usager<br>Tests de volume |
| *3.5. Accepter une annonce* | Tests de fonction<br>Tests d'intégrité des données<br>Tests de stress<br>Tests d'échec/récupération |
| *3.5.4 [Essentiel] Les informations de contact (numéro de téléphone et/ou email) deviennent disponibles par les deux utilisateurs impliqués dans le contrat.* | Tests de sécurité et de contrôle d'accès |
| *3.5.5 [Essentiel] L'annonce devient cachée aux autres utilisateurs que les deux impliqués dans le contrat.* | Tests de sécurité et de contrôle d'accès |
| *3.6.1 [Essentiel] L'utilisateur doit pouvoir acheter des jetons de la plateforme.*<br>*3.7.1 [Essentiel] L'utilisateur doit pouvoir payer à l'aide de la plateforme MoonPay.*<br>*3.8.1 [Essentiel] L'utilisateur doit pouvoir payer par la crypto-monnaie EOS.* | Tests de fonction<br>Tests de charge<br>Tests de stress<br>Tests d'échec/récupération |
| *3.9. Création du système de référencement automatique* | Tests de fonction<br>Tests d'interface usager |
| *3.10.2 [Essentiel] Le montant du contrat doit être mis en escrow lors de la création du contrat.* | Tests de sécurité et de contrôle d'accès<br>Tests d'échec/récupération |
| *3.10.3 [Essentiel] La transaction doit être validée par les deux parties prenantes du contrat* | Tests de sécurité et de contrôle d'accès<br>Tests d'échec/récupération |
| *3.11. Intégration d'une messagerie* | Tests de fonction<br>Tests de charge<br>Tests de stress<br>Tests de volume<br>Tests d'échec/récupération |
| *4.1.1. L'interface utilisateur doit être cohérente entre la section de vente et la section d'achat.* | Tests d'interface usager |
| *4.1.2. L'interface utilisateur doit être facile* | Tests d'interface usager |

| | |
|---|---|
| *d'utilisation.* | Tests de configuration |
| *4.1.3. L'interface utilisateur doit être visuellement agréable.* | Tests d'interface usager |
| *4.1.4. Ajouter une annonce ne devrait pas prendre plus de 5 minutes.* | Tests d'interface usager |
| *4.2.5. En cas d'erreur, le serveur doit redémarrer automatiquement.* | Tests d'échec/récupération |
| *4.2.6. Si le serveur est hors ligne, l'application doit afficher un message d'erreur compréhensible.* | Tests d'échec/récupération |
| *4.2.7. En cas d'erreur lors d'une transaction, elle doit être annulée.* | Tests d'échec/récupération |
| *4.3. Performance* | Tests de performance<br>Tests de charge |
| *4.6. Sécurité* | Tests de sécurité et de contrôle d'accès |

## 3. Stratégie de test

*[Cette section présente la stratégie de test qui sera utilisée afin de tester le logiciel.*

*ATTENTION : NE PAS présenter les cas de tests dans cet artefact, mais seulement les différents types de test qui sont prévus.]*

### 3.1. Types de test

*[Pour chaque type de test jugé pertinent, les informations suivantes sont fournies : objectif de test, technique, critère de complétion, considérations spéciales.]*

#### *3.1.1. Tests de fonction*

*[Les tests de fonction s'assurent de l'implémentation appropriée des cas d'utilisation. Ce type de tests est basé sur les techniques de boîte noire.]*

| | |
|---|---|
| Objectif de test: | Tester une fonctionnalité de bout en bout afin de s'assurer que chaque cas d'utilisation est fonctionnel. |

| | |
|---|---|
| Technique: | Définir des actions à effectuer sur l'application en fonction d'un certain cas d'utilisation et comparer les résultats des actions effectuées aux résultats attendus. |
| Critère de complétion: | Les tests sont complétés lorsque les résultats obtenus sont les mêmes que ceux attendus, confirmant que la fonctionnalité est fonctionnelle. |
| Considérations spéciales: | |

### *3.1.2. Tests d'interface usager*

*[Les tests d'interface usager vérifient l'interaction d'un utilisateur avec le logiciel. Le but de ces tests est de s'assurer que l'interface usager fournit à l'utilisateur un accès et une navigation appropriés.]*

| | |
|---|---|
| Objectif de test: | Tester que l'interface de l'application soit intuitive et appropriée selon nos besoins. |
| Technique: | Demander à un utilisateur d'effectuer une action et observer son interaction avec l'interface. Il est aussi possible d'enregistrer cette interaction afin de revenir sur les moments où l'utilisateur a eu plus de difficulté avec l'interface. |
| Critère de complétion: | Le test est complété lorsque l'utilisateur arrive à compléter les actions demandées selon le critère testé (provenant par exemple d'exigences non-fonctionnelles). |
| Considérations spéciales: | Les tests devront être effectués par un utilisateur ne faisant pas partie de notre équipe afin de s'assurer que l'interaction n'est pas influencée par nos connaissances de l'interface. |

### *3.1.3. Tests d'intégrité des données*

*[Les bases de données devraient être testées, sans passer par l'interface usager du logiciel, afin de s'assurer de l'intégrité des données.]*

| | |
|---|---|
| Objectif de test: | S'assurer que les données qui sont gardées dans la base de données soient cohérentes et correctes selon les situations. |
| Technique: | Effectuer une action sur l'application, puis vérifier directement les données dans la base de données (par requêtes). |
| Critère de complétion: | Le test est complet lorsque les données dans la base de données est tel qu'attendu à la suite d'une certaine action. |
| Considérations spéciales: | |

### *3.1.4. Tests de performance*

*[Le profilage de performance est un test de performance où les temps de réponse, les taux de transaction et autres exigences pertinentes sont mesurées et évaluées.]*

| | |
|---|---|
| Objectif de test: | S'assurer que l'application respecte la performance telle que définie dans nos exigences non-fonctionnelles selon différentes situations. |
| Technique: | Simuler différentes situations et enregistrer des données de performance, tel que le temps de réponse du serveur. |

| | |
|---|---|
| Critère de complétion: | Le test d'un critère de performance est complété lorsque le résultat obtenu pour ce critère respecte le niveau de performance attendu. |
| Considérations spéciales: | |

### *3.1.5. Tests de charge*

*[Un test de charge est un test de performance spécifique à des charges de travail (workload) prédéfinies.]*

| | |
|---|---|
| Objectif de test: | S'assurer que l'application garde un niveau de performance satisfaisant à différents niveaux de charge. |
| Technique: | Simuler plusieurs utilisateurs sur l'application en même temps, puis effectuer des actions et enregistrer les données de performance associées. Répéter ces actions sous différents niveaux de charge (par exemple, 2 utilisateurs, 8 utilisateurs, etc.) afin d'observer l'impact de la charge sur la performance. |
| Critère de complétion: | Le test est complété lorsque le résultat obtenu reste satisfaisant sous les différents niveaux de charge testés. |
| Considérations spéciales: | |

### *3.1.6. Tests de stress*

*[Les tests de stress sont un type de test de performance visant à trouver des erreurs liées à la concurrence face à des ressources partagées et rares (mémoire vive, disque dur, réseau, etc.).]*

| | |
|---|---|
| Objectif de test: | S'assurer que l'application fonctionne correctement et reste dans un état stable lorsqu'un stress est appliqué sur l'application et le serveur. |
| Technique: | Simuler des actions un grand nombre de fois dans un court espace temps et s'assurer que l'application gère correctement les requêtes concurrentes. Tester les parties sensibles de l'application afin de découvrir les erreurs critiques pouvant subvenir. |
| Critère de complétion: | Le test est complété lorsque l'application garde un état stable malgré le stress appliqué sur celle-ci. |
| Considérations spéciales: | |

### *3.1.7. Tests de volume*

*[Les tests de volume assujettissent le logiciel à de grosses quantités de données afin de déterminer si le logiciel possède une limite causant une panne.]*

| | |
|---|---|
| Objectif de test: | S'assurer que le serveur puisse supporter des grandes requêtes (avec beaucoup de données) sans causer d'erreurs ou de panne. |
| Technique: | Envoyer une grande quantité de données au serveur et observer la réponse du serveur suite à la requête. |
| Critère de complétion: | Le test est complété lorsque le serveur répond correctement, ou selon le cas que l'erreur soit correctement gérée, après l'action effectuée. |

| | |
|---|---|
| Considérations spéciales: | |

### *3.1.8. Tests de sécurité et de contrôle d'accès*

*[Les tests de sécurité et de contrôle d'accès s'intéressent à la sécurité au niveau applicatif (accès d'un utilisateur à certaines données au sein du logiciel) et au niveau système (accès d'un utilisateur au système).]*

| | |
|---|---|
| Objectif de test: | S'assurer que l'application est sécuritaire et est résistante aux essais de récupération de données privées. |
| Technique: | Essayer d'effectuer des actions qui devraient être interdites ou d'accéder à des données qui ne devraient pas être disponibles pour un certain utilisateur. |
| Critère de complétion: | Le test est complété lorsqu'aucune faille n'a été trouvée. |
| Considérations spéciales: | |

### *3.1.9. Tests d'échec/récupération*

*[Les tests d'échec/récupération s'assurent que le logiciel puisse récupérer d'une défaillance (matérielle, logicielle, réseau) sans compromettre l'intégrité des données.]*

| | |
|---|---|
| Objectif de test: | S'assurer que l'application puisse réagir correctement lorsqu'un problème survient et reste dans un état stable (ne laissant pas de problèmes latents). |
| Technique: | Volontairement désactiver une partie de l'application, comme la base de donnée ou la connexion internet, puis essayer différentes actions sur l'application en observant son comportement dans cette situation d'échec. |
| Critère de complétion: | Le test est complété lorsque l'application ne permet pas d'actions illégales à l'utilisateur en cas d'échec. |
| Considérations spéciales: | |

### *3.1.10. Tests de configuration*

*[Les tests de configuration vérifient le bon fonctionnement du logiciel sur plusieurs configurations logicielles et matérielles.]*

| | |
|---|---|
| Objectif de test: | S'assurer que l'application fonctionne sous différents environnements, telle que différents utilisateurs l'utilisent. |
| Technique: | Tester l'application sous différents appareils (firmware et hardware) et s'assurer que l'application fonctionne telle qu'attendu. |
| Critère de complétion: | Le test est complété lorsque l'application fonctionne tel que prévu sous différents appareils. |
| Considérations spéciales: | |

## 3.2. Outils

Les outils suivants seront utilisés au sein de la discipline de test:

| Type de test | Outil |
|---|---|
| Tests de performance<br>Tests de charge | Apache JMeter |
| Tests de configuration | Android Studio Emulator |

## 4. Ressources

*[Cette section présente les ressources humaines et matérielles relatives à la discipline de test.]*

### 4.1. Équipe de test

| Rôle | Membre de l'équipe | Responsabilités |
|---|---|---|
| Testeur de fonctionnalités | Gabriel Dambreville | Tests de fonction, tests d'intégrité des données |
| Testeur d'assurance qualité | Guilhem Dubois | Tests de performance, tests de charge, tests de stress, tests de volume, tests de sécurité et de contrôle d'accès, tests d'échec/récupération |
| Testeur d'interface | Arthur Garnier | Tests d'interface usager, tests de configuration |

### 4.2. Système

*[Décrire les ressources systèmes (environnement, configuration) nécessaires pour la discipline de test.]*

| Applications | environnement | Configuration |
|---|---|---|
| Client | Android Emulator | Android 11.0 |
| Serveur | Google Cloud | Node.js |
| Base de donnée | Atlas MongoDB | MongoDB |

## 5. Jalons du projet

*[Détailler les jalons relatifs à la discipline de test.]*

| Jalon | Effort | Date de début | Date de fin |
|---|---|---|---|
| Tests d'exigences fonctionnelles | 20h | 24 novembre | 30 novembre |
| Tests d'exigences non-fonctionnelles | 20h | 1er décembre | 7 décembre |