*Fais-moi un dessin*

Plan de tests logiciels

Version 2.0

Historique des révisions

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2021-03-10 | 1.0 | Première ébauche | Augustin Bouchard |
| 2021-03-24 | 1.1 | Première finition du plan de tests | Augustin Bouchard |
| 2021-04-19 | 2.0 | Révision du plan de tests pour la remise | Augustin Bouchard |

Table des matières

[**1. Introduction**](#_heading=h.1fob9te) **4**

[**2. Exigences à tester**](#_heading=h.3znysh7) **4**

[**3. Stratégie de test**](#_heading=h.2et92p0) **6**

[3.1. Types de test](#_heading=h.tyjcwt) 6

[3.1.1. Tests de fonction](#_heading=h.3dy6vkm) 6

[3.1.2. Tests d’interface usager](#_heading=h.4d34og8) 7

[3.1.3. Tests d’intégrité des données](#_heading=h.3rdcrjn) 7

[3.1.4. Tests de stress](#_heading=h.44sinio) 7

[3.1.5. Tests de volume](#_heading=h.z337ya) 7

[3.1.6. Tests d’échec/récupération](#_heading=h.1y810tw) 8

[**4. Ressources**](#_heading=h.qsh70q) **8**

[4.1. Équipe de test](#_heading=h.1pxezwc) 8

[Tests de fonction, Tests d’interface usager](#_heading=h.6q40244hgnmy) 8

[Tests d’intégrité des données, Tests d’échec/récupération](#_heading=h.agrmpiix0c8l) 8

[Tests de stress, Tests de volume](#_heading=h.yflqt572rlup) 8

[4.2. Système](#_heading=h.49x2ik5) 8

[Electron](#_heading=h.8rvpina0nex) 8

[Android Pie-9](#_heading=h.1mbawvngdymx) 8

[PostgreSQL](#_heading=h.6t8y8261wgu2) 8

[**5. Jalons du projet**](#_heading=h.2p2csry) **8**

**Plan de tests logiciels**

# 

# 1. Introduction

Le présent document contient les informations nécessaires pour comprendre la méthodologie de test du projet *Fais-moi un dessin.* La première section porte sur les exigences à tester. Les exigences du SRS avec leurs tests respectifs y seront présentés. La deuxième section porte sur les stratégies de test employées pour le projet. Seulement les types de tests jugés pertinents y seront discutés. La troisième section s’attarde plutôt aux ressources humaines, matérielles et systèmes relatives à la discipline de test. Finalement, il sera question de détailler l’effort relatif à chaque jalon de la discipline de test.

# 2. Exigences à tester

|  |  |
| --- | --- |
| **Exigences** | **Tests associés** |
| 3.1.1 Accès au clavardage | * Tests d’interface usager |
| 3.1.2 Clavardage intégré au mode fenêtré | * Tests d’interface usager |
| 3.3.1 Créer, rejoindre, quitter, supprimer des canaux de discussion | * Tests d’interface usager * Tests d’intégrité des données |
| 3.3.8 Vider la boîte de texte après l’envoi d’un message et garder l’accent sur la boîte de message | * Tests d’interface usager |
| 3.3.12 Le message doit être accompagné de l’heure d’envoie | * Tests de stress |
| 3.3.13 Les caractères spéciaux UTF8 sont supportés | * Tests de volume |
| 3.4.1.3 Le profil doit afficher les informations du joueur | * Tests d’interface usager |
| 3.4.2 Une erreur doit être envoyée si le pseudo n’existe pas | * Tests d’échec/récupération |
| 3.4.3 Une erreur doit être envoyée si le mot de passe n’est pas le bon | * Tests d’échec/récupération |
| 3.5.1.2 Un seul joueur peut dessiner à la fois | * Tests de fonction * Tests d’interface usager |
| 3.5.2 Le mode de jeu classique nécessite minimum 2 vrais joueurs pour une partie de 4 joueurs. | * Tests de fonction * Tests d’interface usager * Tests d’intégrité des données * Tests de volume * Tests de stress * Tests d’échec/récupération |
| 3.5.2.1.1 Lorsque le joueur virtuel dessine, l’envoie se fait sans erreur | * Test de volume |
| 3.6.1 Un joueur peut créer manuellement une paire mot-image avec les informations nécessaires | * Tests de fonction * Tests d’intégrité des données * Tests de volume |
| 3.6.3 Le joueur peut importer une image | * Tests d’intégrité des données |
| 3.7.1 Le joueur peut dessiner avec tous les outils sur une surface de dessin | * Tests de fonction * Tests d’interface usager |
| 3.7.6 Les autres joueurs doivent voir exactement le même dessin | * Tests de stress |
| 3.10.1 Le système doit présenter des effets visuels et sonores | * Tests d’interface usager |
| 3.11.1 Le système doit indiquer lorsqu’une réponse donnée par un utilisateur est proche du mot réel | * Tests d’interface usager |
| 3.12.1 L’utilisateur doit pouvoir se connecter s’il ne l’est pas déjà, se déconnecter ou créer un compte | * Tests de stress * Tests d’échec/récupération |
| 3.13.2 Le joueur peut rejoindre ou créer un lobby | * Tests d’interface usager |
| 3.13.4.5 Le joueur rejoint le clavardage du lobby lorsqu’il rejoint le lobby | * Tests d’interface usager * Tests d’échec/récupération |
| 3.14.1 Le joueur peut attribuer un pouce à l’artiste | * Tests d’interface usager |
| 3.17.1 Mode sprint solo | * Tests de fonction * Tests d’intégrité des données * Tests de volume * Tests de stress * Tests d’échec/récupération |
| 3.17.1.2 Lorsque le joueur virtuel dessine, l’envoie se fait sans erreur | * Test de volume |
| 3.17.2 Mode sprint coop | * Tests de fonction * Tests d’intégrité des données * Tests de volume * Tests de stress * Tests d’échec/récupération |
| 3.17.2.2 Lorsque le joueur virtuel dessine, l’envoie se fait sans erreur | * Test de volume |
| 3.17.3.6 Lorsque le joueur virtuel dessine, l’envoie se fait sans erreur | * Test de volume |
| 3.19.1 Leaderboard | * Tests d’intégrité des données |
| 4.1.1 L’interface est cohérente entre les différents clients | * Tests d’interface usager |
| 4.1.4 Les dessins sont cohérents sur les différents clients | * Tests d’interface usager |
| 4.1.5 Le logiciel est maîtrisable en une partie | * Tests d’interface usager |
| 4.3.3 Le logiciel supporte 8 connexions (4 par partie) | * Tests de stress |

# 3. Stratégie de test

## 3.1. Types de test

### 3.1.1. Tests de fonction

|  |  |
| --- | --- |
| Objectif de test: | S’assurer que le logiciel implémente les fonctionnalités appropriées aux différents cas d’utilisation. |
| Technique: | Définir des tests, des actions ou des cas d’utilisation ainsi que les résultats attendus pour chacun d’entre eux. |
| Critère de complétion: | Les tests de fonction sont complétés lorsque tous les résultats concordent avec les résultats attendus. |
| Considérations spéciales: |  |

### 3.1.2. Tests d’interface usager

|  |  |
| --- | --- |
| Objectif de test: | S’assurer que les fonctionnalités de l’application fonctionnent selon leurs spécifications. L’interface doit être cohérente et permettre à l’utilisateur de faire les actions désirées et de faire les navigations appropriées. |
| Technique: | * Effectuer les tests manuellement (observer, noter, essayer les boutons, naviguer dans l’application) * Effectuer les tests pendant un enregistrement, puis visionner l'enregistrement pour noter et évaluer le logiciel |
| Critère de complétion: | Les tests suivent une logique vrai ou faux. Un test est complété lorsqu’il remplit le critère spécifié. |
| Considérations spéciales: | Les tests devraient être exécutés par quelqu’un d’autre |

### 3.1.3. Tests d’intégrité des données

|  |  |
| --- | --- |
| Objectif de test: | S’assurer que les données enregistrées sur la base de données soient soient les bonnes et soient conformes aux spécifications. |
| Technique: | * Faire des requêtes directement sur la base de donnée pour vérifier les résultats * Vérifier le contenu des tables sur la base de donnée |
| Critère de complétion: | Un test d’intégrité des données est complété lorsque la table ou la requête retourne toutes les informations attendues avec les bons types. |
| Considérations spéciales: |  |

### 3.1.4. Tests de stress

|  |  |
| --- | --- |
| Objectif de test: | S’assurer que le logiciel opère de la même façon en situation de concurrence. S’assurer qu’il n’y a pas de pertes de données. |
| Technique: | Être plusieurs à exécuter les mêmes tâches en même temps. |
| Critère de complétion: | Un test de stress est complété lorsque toutes les actions exécutées sur chaque client sont cohérentes et qu’il n’y pas de diminution de performance observable. |
| Considérations spéciales: | Les tests de stress doivent être réalisés à plusieurs. |

### 

### 

### 

### 

### 

### 3.1.5. Tests de volume

|  |  |
| --- | --- |
| Objectif de test: | S’assurer que le l’envoie de grosses quantités de données ne créent pas d’erreurs ou de panne. |
| Technique: | Envoyer des grosses quantités de données au serveur et déterminer si ces données sont toutes transférées. |
| Critère de complétion: | Un test de volume est complété lorsque les données reçues sont les mêmes que celles qui ont été envoyées. |
| Considérations spéciales: |  |

### 3.1.6. Tests d’échec/récupération

|  |  |
| --- | --- |
| Objectif de test: | S’assurer que les échecs et les interruptions ne compromettent pas l’intégrité des données. |
| Technique: | Simuler des échecs ou des interruptions et noter l’état des données enregistrées par la suite. |
| Critère de complétion: | Un test d’échec/récupération est complété lorsque les données dans la base de données sont conformes au résultat attendu en cas d’échec. |
| Considérations spéciales: |  |

# 4. Ressources

## 4.1. Équipe de test

|  |  |  |
| --- | --- | --- |
| **Rôle** | **Membre de l’équipe** | **Responsabilités** |
| Comportements attendus | Simon Ayotte | Tests de fonction, Tests d’interface usager |
| Données et récupération | Félix Dumont | Tests d’intégrité des données, Tests d’échec/récupération |
| Briseur d’application | Mark Weber-Sadler | Tests de stress, Tests de volume |

## 

## 

## 

## 4.2. Système

|  |  |  |
| --- | --- | --- |
| **Applications** | **environnement** | **Configuration** |
| Client-lourd | Windows 10, macOs | Electron |
| Client-léger | Android | Android Pie-9 |
| Base de donnée | MCAzure | PostgreSQL |

# 5. Jalons du projet

|  |  |  |  |
| --- | --- | --- | --- |
| **Jalon** | **Effort** | **Date de début** | **Date de fin** |
| Test de l’application sans les jeux | 16h | 7 avril 2021 | 13 avril 2021 |
| Test des jeux | 8h | 14 avril 2021 | 19 avril 2021 |
| Tests non-fonctionnels | 8h | 14 avril 2021 | 19 avril 2021 |