**Chat test**

**Plan de tests**

**Version 0.1**

**2017-07-05**

**Baptiste Fotia**

*[Note: Le texte inséré entre crochets et affiché en italique bleu sert à fournir un guide à l’auteur et doit être effacé avant la publication du document. Le paragraphe suivant ce texte est en style Normal.]*

*[Pour personnaliser les champs automatiques avec Microsoft Word, qui affichent un arrière plan lorsqu’on les sélectionne activer Fichier/Propriétés et modifier les champs Sujet, Auteur, Compagnie et catégorie de l’onglet Résumé, et le champ Version de l’onglet Personnalisation Ces champs peuvent être mis à jour dans le document en les sélectionnant tous ou individuellement et en activant la touche F9. Les champs des en-têtes et des pieds de page doivent être mise à jour séparément.]*

**Historique des modifications du document**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| 2019-12-10 | 0.1 | Premier jet du plan | Baptiste |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table des matières**

1. Introduction 4

1.1 Objectif du document 4

1.2 Portée du document 4

1.3 Historique 4

1.4 Définitions, acronymes et abréviations 4

1.5 Identification des artéfacts 4

1.6 Vue d’ensemble 5

2. Exigences à tester 6

3. Stratégie de tests 6

3.1 Types de tests 6

3.1.1 Tests fonctionnels 6

3.1.2 Tests d’interface utilisateur 6

3.1.3 Tests de données et d’intégrité de base de données 7

3.1.4 Profilage de performance 7

3.1.5 Tests de charge 8

3.1.6 Tests de stress 9

3.1.7 Tests de volumétrie 10

3.1.8 Tests de sécurité et de contrôle d’accès 11

3.1.9 Tests de basculement et de récupération 12

3.1.10 Tests de configuration 14

3.1.11 Tests d’installation 14

3.2 Outils 15

4. Ressources 15

4.1 Travailleurs 15

4.2 Système 17

5. Jalons du projet 17

6. Biens livrables 17

6.1 Modèle de test 17

6.2 Journaux de test 18

6.3 Rapports d’anomalies 18

7. Annexe A: Tâches du projet 19

**Plan de tests**

1. **Introduction**

*[L’introduction donne une vue d’ensemble de tout le document. On y présente toute information que le lecteur a besoin pour comprendre le document. Elle comprend l’objectif du document, sa portée, les définitions, acronymes et abréviations, les références et une vue d’ensemble du document.]*

*Le projet permet de chatter avec d’autres personnes et de commenter des articles. On a une base de données SQLite3 qui contient les informations nécessaires sur les utilisateurs, les articles et les commentaires. Un commentaire est associé à un utilisateur (par son identifiant) et un article n’est pas identifié par un utilisateur.*

* 1. **Objectif du document**

*[Préciser les objectifs de ce document comme, par exemple :*

* *Identifier les informations existantes du projet et les composants qui doivent être testés.*
* *Énumérer les exigences d’évaluation à haut niveau*
* *Recommander et décrire les stratégies de test à employer*
* *Identifier les ressources nécessaires et fournir un estimé de l’effort de test*
* *Identifier les biens livrables pour les tests.]*

* *Nous avons possédons des ressources sur notre système pour chaque entité :*
  + *Les utilisateurs*
  + *Les commentaires*
  + *Les articles*
* *Par ordre de priorité des tests par entité :*
  + *Il est très important de tester en priorité (1) les ressources des utilisateur (vérification des mots de passes, chiffrages, que ce soit le bon mot de passe correspondant à l’utilisateur, etc.)*
  + *Les commentaires doivent être testés (priorité 2) puisqu’ils sont publiés en fonction de l’utilisateur*
  + *La dernière priorité (3) est de tester les articles.*
* *Tester la sécurité du système/projet est très important*

1. **Exigences à tester**

La liste suivante identifie les items, cas d’utilisation, exigences fonctionnelles et exigences non-fonctionnelles, qui ont été désignés comme cibles de test. Cette liste représente ce qui sera testé.

*[Énumérer les principales exigences à tester]*

* Tests de documentation
* Tests d’interface utilisateur
* Tests fonctionnels
* Tests d’interopérabilité ou d’intégration
* Tests de charge
* Tests de sécurité
* Tests multi plates-formes
* Tests de données et d’intégrité de base de données

1. **Stratégie de tests**

*[La stratégie de tests présente l’approche recommandée d’évaluation des cibles des tests. La précédente section, Exigences à tester, décrit de qui sera testé et cette section comment cela sera testé.*

*Fournir, pour chaque type de test, une description du test et comment il sera implémenté et exécuté.*

*Lorsque les tests ne sont pas implémentés et exécutés, expliquer en une phrase pourquoi.*

*La stratégie de tests consiste d’abord à identifier les techniques de tests et à identifier les critères de complétion des tests.*

*Les tests doivent être exécutés dans des environnements sécurisés avec des données connues et contrôlées.]*

* 1. **Types de tests**

**Tests fonctionnels**

|  |  |
| --- | --- |
| **Stratégie pour les tests fonctionnels** |  |
| Objectif de test : | Vérifier que la fonctionnalité rencontre la bonne cible de test, incluant la navigation, la saisie de données, le traitement et la récupération |
| Technique | Exécuter les cas d’utilisation, chaque chemin du cas ou fonction en utilisant des données valides et non valides tout en vérifiant:   * Les résultats attendus avec des données valides. * Les messages d’erreur et d’avertissement lorsque des données non valides sont utilisées * Que chaque règle d’affaire est appliquée correctement |
| Critère de complétion | * Tous les tests prévus ont été exécutés. * Toutes le anomalies identifiées ont été enregistrées |
| Considérations particulières | Néant |

**Test d’interface utilisateur**

|  |  |
| --- | --- |
| **Stratégie pour les tests d’interface utilisateur** | |
| Objectif de test : | Évaluer:   * La navigation des cibles de test représentent adéquatement les fonctions d’affaire et les exigences, incluant d’écran à écran, de champ à champ, et l’utilisation des modes de déplacement tels que les clés de tabulation, le mouvement de la souris et les clés d’accès rapide. * Les composants de l’écran et les caractéristiques tels les menus, la taille, la position, la cible de saisie et l’état sont conformes aux normes. |
| Technique | Produire des tests pour chaque écran afin de vérifier la navigation et les états de chacun des écrans et composants de l’application. |
| Critère de complétion | * La vérification de chaque écran est concluante en se conformant à la référence ou à une norme acceptable. |
| Considérations particulières | Il est possible que toutes les propriétés personnalisables ou des composants d’un tiers ne soient testées. |

**Test de sécurité et de contrôle d’accès**

|  |  |
| --- | --- |
| **Stratégie pour les tests de sécurité et de contrôle d’accès** |  |
| Objectif de test : | La sécurité au niveau de l’application: Vérifier qu’un acteur ne peut accéder aux seules données et fonctions pour lesquelles son type d’utilisateur lui donne les autorisations.  La sécurité au niveau du système : Vérifier que seuls les acteurs ayant l’autorisation, ont un accès au système et aux application. |
| Technique | Niveau application:   * Identifier et énumérer chaque type d’acteur avec les fonctions et données pour lesquelles il a les autorisations. * Créer des tests pour chaque type d’acteur afin et vérifier chaque autorisation en exécutant des transactions spécifique à chaque utilisateur acteur. * Modifier un type d’utilisateur et ré exécuter les tests pour le même utilisateur. Dans chaque cas, vérifier si les données et fonctions sont disponibles ou non disponibles tel que souhaité.   Niveau système: Voir les considérations particulières. |
| Critère de complétion | Pour chaque type d’acteur connu, les fonctions et données appropriées sont disponibles et si toutes les transactions fonctionnent comme prévu et s’exécutent dans les précédents tests fonctionnels. |
| Considérations particulières | L’accès au système doit être révisé et discuté avec les administrateurs appropriés de système ou de réseau. Ces tests peuvent ne pas être nécessaires s’ils sont déjà une fonction de l’administration du réseau ou du système. |

**Test de charge**

|  |  |
| --- | --- |
| **Stratégie pour les tests de charge** |  |
| Objectif de test : | Vérifier les temps de performance pour des transactions données ou pour des cas d’affaire selon différentes conditions de charge de travail. |
| Technique | Utiliser des tests développer pour les tests fonctionnels ou du cycle d’affaire.  Modifier les fichiers de données pour accroître le nombre de transactions ou les tests pour accroître la fréquence des transactions. |
| Critère de complétion | Une complétion est réussie lorsque des tests à plusieurs transactions et à plusieurs utilisateurs sans défaillance et dans un délai acceptable. |
| Considérations particulières | Les tests de charges doivent être exécutés sur une machine dédiée, à un moment dédié afin de permettre un contrôle complet et des mesures exactes.  Les bases de données utilisées pour les tests de performance doivent avoir une taille courante ou avoir une taille proportionnellement comparable. |

**Tests de données et d’intégrité de base de données**

|  |  |
| --- | --- |
| **Stratégie pour les tests de données et d’intégrité de base de données** |  |
| Objectif de test : | Vérifier les méthodes d’accès à la base de données ainsi que les processus fonctionnent correctement et sans corruption de données. |
| Technique | Appeler chaque méthode et processus d’accès à la base de données en alimentant chacun de données valides, de données invalides et de requêtes de données.  Inspecter la base de données pour vérifier que les données ont été chargées comme prévu, que toutes les opérations de la base de données s’exécutent normalement et réviser les données produites afin de s’assurer que les bonnes données ont été extraites et pour les bonnes raisons. |
| Critère de complétion | Tous les processus et méthodes d’accès fonctionnent tels qu’ils ont été conçus sans corruption de données. |
| Considérations particulières | Les tests peuvent nécessiter un environnement de développement du SGBD ou des pilotes pour charger ou modifier directement des données dans la base de données.  Les processus doivent être activés manuellement.  Des bases de données petites ou de taille minimale devraient être utilisées pour accroître la visibilité d’événements non acceptables. |

**Tests multi plates-formes**

|  |  |
| --- | --- |
| Objectif de test : | Vérifier que le système/projet soit compatible sur toutes les principales plateformes du marché. |
| Technique | Identifié la plateforme avant exécution du système/projet pour déterminer les différents appels systèmes.  Charger les bons binaires en fonction de la plateforme. |
| Critère de complétion | Les appels de la plateforme n’interfèrent pas le système/projet.  Vérifier que l’on dispose des bonnes versions des différents binaires. |
| Considérations particulières | Néant |

**Tests de documentation**

|  |  |
| --- | --- |
| Objectif de test : | Vérifier que la documentation soit la plus compréhensible possible avec les autres membres qui suivent le projet |
| Technique | Écrire la documentation. |
| Critère de complétion | Aucune documentation existe pour ce système/projet. |
| Considérations particulières | Assurer le suivi et le maintien de la documentation. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Récapitulatif des tests de documentation** | | | | |
| **Numéro** | **Nom du test** | **Couverture Tests** | **Couverture du code** | **Test prévu** |
| 1 | Écriture de la documentation | 0% | 0% | NON Car aucune documentation existe pour le moment. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Récapitulatif des tests d’interface** | | | | |
| **Numéro** | **Nom du test** | **Couverture Tests** | **Couverture du code** | **Test prévu** |
| 1 | Interface utilisateur respectant les règles d'UX Design | 0% | 0% | Aucun fichier CSS ne permet de designer le système/projet. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Récapitulatif des tests fonctionnels** | | | | |
| **Numéro** | **Nom du test** | **Couverture Tests** | **Couverture du code** | **Test prévu** |
| 1 | Connection à la base de données | 0% | 5% | Oui. Le code est déjà écrit pour se connecter à la base mais aucun test n'est prévu. |
| 2 | Respect des valeurs d'entrées et de sorties selon l'utilisation des fonctions définis | 0% | 10% | Oui. Les fonctions sont déjà écrites mais aucun test ne permet de vérifier les valeurs d'entrées et de sorties. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Récapitulatif des tests de charge** | | | | |
| **Numéro** | **Nom du test** | **Couverture Tests** | **Couverture du code** | **Test prévu** |
| 1 | Vérifier le nombre de personne connecté | 0% | 10% | Oui. Nous utilisons une base Sqlite3 qui ne peut supporter un certain nombre d'utilisateur simultanément dans le système/projet. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Récapitulatif des tests de sécurité** | | | | |
| **Numéro** | **Nom du test** | **Couverture Tests** | **Couverture du code** | **Test prévu** |
| 1 | Les mots de passes ne sont pas stocké en clair dans la base de donnée | 0% | 0% | Oui. Actuellement, les mots de passes sont écrit en clair et doivent être chiffré. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Récapitulatif des tests de multi plateforme** | | | | |
| **Numéro** | **Nom du test** | **Couverture Tests** | **Couverture du code** | **Test prévu** |
| 1 | Exécuter le projet sur diverse plateformes | 0% | 0% | Oui. Installer les binaires et services nécessaires dans les plateformes autres que celles prévuent. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Récapitulatif des tests de données et d’intégrité de base de données** | | | | |
| **Numéro** | **Nom du test** | **Couverture Tests** | **Couverture du code** | **Test prévu** |
| 1 | Le mot de passe de l'utilisateur n'a pas changé | 0% | 10% | Oui. Il faudra créer un champ de confirmation pour le mot de passe. |
| 2 | Le commentaire/article de l'utilisateur est différent | 0% | 10% | Oui. |

1. **Annexe A: Tâches du projet**

Voici la liste des tâches liées à l’exécution des tests :

* Planifier les tests
  + Identifier les exigences de test
  + Évaluer les risques
  + Identifier les ressources nécessaires aux tests
  + Développer une stratégie de test
  + Créer le calendrier de tests
  + Générer le plan de test
* Concevoir les tests
  + Analyser la charge de travail
  + Identifier et décrire les cas de test
  + Identifier et structurer les procédures de test
  + Réviser la couverture des tests
* Implémenter les tests
  + Enregistrer ou programmer les scripts de test
  + Identifier les fonctionnalités propres au test dans les modèles de conception et d’implémentation.
  + Établir des ensembles de données externes
* Exécuter les tests
  + Exécuter les procédures de test
  + Évaluer l’exécution des tests
  + Récupérer les tests suspendus
  + Vérifier les résultats
  + Examiner les résultats non prévus
  + Enregistrer les anomalies
* Évaluer les tests
  + Évaluer la couverture des cas de tests
  + Évaluer la couverture du code
  + Analyser les anomalies
  + Déterminer si les critères de complétion et de succès des tests ont été atteints.