Équipe 11

Projet intégrateur de Troisième Année
Plan de tests logiciels

Version 3.0

Historique des révisions

Date	Version	Description	Auteur
2019-03-19	1.0.	J'ai fait presque tout. Les tests d'installation ont été retirés.	Mylène P.
2019-03-21	1.1.	Section 4 "ressource" est complétée. Tout est terminée. Reste à savoir si nous allons retirer les tests de stress.	Mylène P.
2019-03-29	1.2.	Nettoyage du document.	Mylène P.
2019-03-06	2.0	Ajout d'informations dans plusieurs parties	Tom Gautier
2019-03-08	3.0	Revue du document	Tom Gautier

Table des matières

Plan de tests logiciels

1. Introduction

Le plan de tests sert à établir et à détailler différents tests à effectuer afin de pouvoir vérifier l'assurance qualité du projet.

Ce document présentera les exigences à tester, les types de tests utilisés, les outils utilisés, les ressources requises ainsi que les jalons du projet à respecter pour les tests.

2. Exigences à tester

Malgré le fait que nous mentionnons uniquement les catégories de profondeur X.X.X, les tests seront réalisés pour toutes les exigences dans le SRS associé. La liste qui suit identifie ce qui sera testé (cas d'utilisation, exigences fonctionnelles et exigences non-fonctionnels) :

3.1.1. Clavardage - Intégration

3.1.2. Clavardage - Canaux de discussion

3.1.3. Profil utilisateur et galerie

3.1.4. Édition de base collaborative

3.1.5. Édition de formes collaborative

3.1.6. Sauvegarde d'image et chargement

3.1.7. Accessibilité des images

3.1.8. Tutoriel

3.1.9. Amis et liste d'utilisateurs

3.1.10. Accueil

3.1.11. Langues

3.1.12. Connexion

3. Stratégie de test

Cette section présente la stratégie de test qui sera utilisée afin de tester le logiciel. En effet, on ne mentionnera pas ici les cas de tests mais plutôt les stratégies haut niveau pour chaque type de test prévu.

3.1. Types de test

3.1.1. Tests de fonction

Objectif de test:	Valider que les principales exigences fonctionnelles soient satisfaites.
Technique:	Série de tests de type *boîte noire* pour les exigences essentielles du SRS.
Critère de complétion:	La sortie obtenue est égale au résultat attendu.
Considérations spéciales:	

3.1.2. Tests de sécurité et de contrôle d'accès

Objectif de test:	Détecter les failles de sécurité du système.
Technique:	Étant donné que les tests vont concerner principalement la connexion. Nous allons donc tenter de nous connecter avec des informations de connexion invalides. Par exemple, il faudra vérifier si l'utilisateur peut accéder au contenu d'un compte après avoir entré un mot de passe absent ou incorrect.
Critère de complétion:	Le résultat du test doit toujours être un accès refusé lorsque les informations de connexion sont invalides et accepté lorsque celles-ci sont valides.
Considérations spéciales:	

3.1.3. Tests de configuration

Objectif de test:	S'assurer du bon fonctionnement du logiciel sur plusieurs configurations logicielles et matérielles.
Technique:	Vérifier la compatibilité de l'application sur plusieurs appareils (Windows 10 pour le client lourd et Android, spécifiquement sur un Nexus 9 pour le client léger).
Critère de complétion:	Le comportement du système est similaire entre les divers appareils.
Considérations spéciales:	

3.1.4. Tests d'interface usager

Objectif de test:	Les tests d'interface usager permettent d'assurer que les fonctionnalités de manière adéquate afin de répondre aux attentes de l'utilisateur. On veut également assurer une navigation intuitive, fluide et agréable entre les différents écrans.
Technique:	Ces tests sont des cas d'utilisation et seront effectués par des testeurs externes. Cela permettra de tester différemment et de trouver des problèmes qui n'auraient jusqu'à lors pas été découverts. De même, on pourra recueillir leur rétroaction quant à l'utilisabilité et la prise en main du logiciel afin d'améliorer l'expérience utilisateur.
Critère de complétion:	L'utilisateur réalise les tâches spécifiées facilement sans aucune erreur logicielle.
Considérations spéciales:	L'utilisateur devrait pouvoir utiliser les interfaces des deux clients sans difficulté majeure puisque qu'elles doivent être cohérentes entre elles.

3.1.5. Tests de performance

Objectif de test:	Valider que la performance du système correspond à ce qui est attendue dans les requis SRS.
Technique:	Mesurer les temps de réponses pour les cas de tests demandés. Ex.:4.3.3. Le système doit permettre de connecter l'utilisateur à une session collaborative en moins de 30 secondes.
Critère de complétion:	Les métriques obtenues sont conformes aux résultats attendus.
Considérations spéciales:	Il faut effectuer les tests trois fois afin de valider l'authenticité des résultats obtenus.

3.2. Outils

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

Type de test	Outil
Tests de fonction pour le serveur	Jasmine, Mockito
Tests de fonction pour le client léger	JUnit
Tests de fonction pour le client lourd	NUnit
Tests de sécurité et de contrôle d'accès	Rétroactions d'utilisateurs
Tests de configuration	Rétroactions d'utilisateurs
Tests d'interface usager	Rétroactions d'utilisateurs
Tests de performance	Visual Studio, Android Studio

4. Ressources

4.1. Équipe de test

Rôle	Membre de l'équipe	Responsabilités
Concepteur	Tristan Beaulieu	*Client léger* Identifier les besoins et cas de tests Définir les critères de complétions
Concepteur	Olivier Boivin	*Client lourd* Identifier les besoins et cas de tests Définir les critères de complétions
Testeur	Maxim Cousineau	*Client léger* Implémenter et exécuter les tests
Testeur	Nicolas Coutu	*Client lourd* Implémenter et exécuter les tests
Coordonnateur des tests	Tom Gautier	Coordonner les concepteurs et testeurs S'assurer de la correcte exécution des tests Compiler l'information dans le rapport de résultats de tests
Analyste	Mylène Pellemans	Élaborer le plan de tests logiciels Élaborer des cas de test

4.2. Système

Les tests logiciels seront programmés dans le même langage que celui employé pour chacunes des composantes de l'application. C'est-à-dire:

- Client lourd: C#/WPF;
- Client léger: Java;
- Serveur: Typescript sur le cadre logiciel de Node.js;

Les développeurs utilisent les logiciels de programmation Microsoft Visual Studio et Android Studio.

5. Jalons du projet

Jalon	Effort	Date de début	Date de fin
Élaboration du plan de tests	7h	2019-03-15	2019-03-22
Implémentation des tests	10h	2019-03-22	2019-03-31
Tester le projet	15h	2019-03-22	2019-04-08