
Équipe 203

PolyDraw
Plan de tests logiciels
Version 1.0

Historique des révisions

Date	Version	Description	Auteur
2020-04-10	1.0	Compléter le plan de tests.	Pascal-Alexandre Morel Cédric Tessier Allan Beddouk Martin Pouliot

Table des matières

1. Introduction	4
2. Exigences à tester	4
3. Stratégie de test	12
3.1. Types de test	12
3.1.1. Tests de fonction	12
3.1.2. Tests d'interface usager	13
3.1.3. Tests d'intégrité des données	13
3.1.4. Tests de performance	13
3.1.5. Tests de charge	14
3.1.6. Tests de stress	14
3.1.7. Tests de volume	14
3.1.8. Tests de sécurité et de contrôle d'accès	14
3.1.9. Tests d'échec/récupération	14
3.2. Outils	15
4. Ressources	15
4.1. Équipe de test	15
4.2. Système	16
5. Jalons du projet	16

Plan de tests logiciels

1. Introduction

Ce document contient les exigences que l'équipe aura à tester pour s'assurer de la qualité du logiciel. L'équipe aura à repasser sur les exigences du SRS et du complément pédagogique pour s'assurer qu'elles respectent les critères qui leur ont été définis. Pour faire ainsi, un type de test sera associé à chaque exigence pertinente qui a été extraite des documents mentionnés. Par la suite, ces types de tests seront définis. Leurs objectifs, leurs critères de complétion et la technique utilisée pour valider ces types de tests seront alors précisés. Puis, les outils utilisés pour chaque type de test seront mentionnés. Il sera ensuite question des ressources qui aideront à la complétion des tests. L'équipe de testeur et les systèmes utilisés pour les tests seront définis. Finalement, l'effort lié à l'activité des tests par jalon sera illustré dans un tableau.

2. Exigences à tester

Exigences	Tests associés
3.1.1.4 Le système doit s'assurer que le nom de compte de l'utilisateur est composé de 4 à 12 caractères alphanumériques.	Test de fonction
3.1.1.5 Le système doit s'assurer que le mot de passe de l'utilisateur doit être composé de 8 à 64 caractères.	Test de fonction
3.1.1.6 Le système doit avertir l'utilisateur d'un critère non respecté.	Test d'interface usager
3.1.1.7 Le système doit vérifier l'unicité du nom d'utilisateur.	Test d'intégrité des données
3.1.1.8 Le système doit avertir l'utilisateur après une tentative de connexion échouée.	Test d'interface usager
3.1.1.9 Le système doit empêcher l'utilisateur de se connecter si sa session est déjà ouverte.	Test de fonction, test d'intégrité des données
3.1.1.10 Le système doit informer l'utilisateur d'une session déjà ouverte.	Test d'interface usager
3.1.1.11 Le système doit accepter les caractères	Test de fonction

spéciaux pour le mot de passe de l'utilisateur.	
3.1.1.12 Le système doit permettre à un utilisateur de se connecter automatiquement (se souvenir de moi)	Test de fonction, test d'interface usager
3.2.1.5 Le système doit permettre à l'utilisateur de créer une partie.	Test de fonction, test d'interface usager
3.2.1.6 Le système doit permettre à l'utilisateur de choisir le mode de jeu mêlée générale pour la création d'une partie.	Test de fonction, test d'interface usager
3.2.1.7 Le système doit permettre à l'utilisateur de choisir le nombre de rondes avant la fin de la partie pour la création d'une partie.	Test de fonction, test d'interface usager
3.2.1.8 Le système doit permettre à l'utilisateur de choisir entre 2 et 5 joueurs maximum pour la création d'une partie mêlée générale.	Test de fonction, test d'interface usager
3.2.1.9 Le système doit permettre à l'utilisateur ayant créé une partie (l'hôte) de la débiter dès qu'un autre joueur humain rejoint la partie en attente pour les modes mêlée générale et coopératif.	Test de fonction
3.2.1.10 Le système doit permettre à l'hôte de retirer un joueur de la partie en attente.	Test de fonction
3.2.1.11 Le système doit permettre à l'hôte d'ajouter un joueur virtuel à une partie mêlée générale en attente.	Test de fonction
3.2.1.12 Le système doit empêcher un utilisateur de joindre une partie pleine	Test de fonction, test d'intégrité des données
3.2.1.13 Le système doit afficher la langue de chaque partie affichée	Test d'interface usager
3.3.1.1 Le système doit permettre à l'utilisateur de choisir le mode de jeu sprint solo pour la création d'une partie.	Test de fonction, test d'interface usager
3.3.1.2 Le système doit permettre à l'utilisateur de choisir le mode de jeu sprint coopératif pour la création d'une partie.	Test de fonction, test d'interface usager

3.3.1.3 Le système doit permettre à l'utilisateur de choisir la taille des équipes pour la création d'une partie en mode sprint coopératif.	Test de fonction, test d'interface usager
3.3.1.4 Le système doit restreindre la taille des équipes entre 2 et 4 joueurs pour la création d'une partie en mode sprint coopératif.	Test de fonction, test d'interface usager
3.4.1.1 Le système doit permettre à l'utilisateur de visualiser les informations publiques de son compte.	Test de fonction, test d'interface usager
3.4.1.2 Le système doit permettre à l'utilisateur de visualiser les informations privées de son compte.	Test de fonction, test d'interface usager
3.4.1.4 Le système doit permettre à l'utilisateur de visualiser l'historique de ses dernières connexions.	Test de fonction, test d'interface usager
3.4.1.5 Le système doit fournir un historique de connexion ayant une déconnexion couplée à chaque connexion, à l'exception de la session actuelle.	Test de fonction, test d'interface usager
3.4.1.6 Le système doit permettre à l'utilisateur de visualiser l'historique de ses parties	Test d'interface usager
3.4.1.7 Le système doit afficher la liste des participants humains de chaque partie dans l'historique des parties.	Test de fonction, test d'interface usager
3.4.1.8 Le système doit afficher le gagnant d'une partie mêlée générale dans l'historique des parties.	Test de fonction, test d'interface usager
3.4.2.1 Le système doit permettre à l'utilisateur de changer le thème de couleurs de l'interface parmi 4 choix (client léger).	Test de fonction, test d'interface usager
3..5.1.6 Le système doit assurer la validité du mot d'un jeu créé en le cherchant dans un dictionnaire anglais ou français.	Test de fonction
3.5.1.7 Le système doit empêcher l'utilisateur d'ajouter un mot vulgaire en cherchant dans une liste de mots vulgaires pour la création d'un jeu.	Test de fonction
3.5.1.8 Le système doit permettre à l'utilisateur	Test de fonction, test d'interface usager

d'ajouter un indice dynamiquement pour la création d'un jeu.	
3.5.1.9 Le système doit s'assurer que l'utilisateur soumet au moins un indice pour le jeu.	Test de fonction
3.5.1.12 Le système doit s'assurer qu'un indice ne contienne pas le mot à deviner.	Test de fonction
3.5.1.13 Le système doit permettre à l'utilisateur de fournir des images de type BMP, JPG ou PNG.	Test de fonction
3.5.1.22 Le système doit permettre à l'utilisateur d'arrêter la prévisualisation d'un dessin.	Test de fonction
3.6.1.1 Le système doit permettre à l'utilisateur de suivre un tutoriel non interactif après la création de son compte, au premier démarrage de l'application.	Test de fonction, test d'interface usager
3.6.1.6 Le système doit permettre à l'utilisateur de revoir le tutoriel en tout temps.	Test de fonction, test d'interface usager
3.7.6 Le système doit s'assurer qu'un joueur virtuel puisse dessiner lorsque vient son tour lors d'une partie sprint coopératif.	Test de fonction, test de performance
3.7.7 Le système doit s'assurer qu'un joueur virtuel puisse envoyer un message personnalisé selon les statistiques d'un utilisateur.	Test de fonction, test d'interface usager
3.7.8 Le système doit assigner une personnalité unique à un joueur virtuel qui se reflétera dans son style de dessin.	Test de fonction
3.7.9 Le système doit assigner une personnalité unique à un joueur virtuel qui se reflétera dans son message.	Test de fonction
3.8.1.1 Le système doit permettre à l'utilisateur de clavarder dans une fenêtre intégrée à l'application.	Test de fonction, test d'interface usager
3.8.1.2 Le système doit permettre à l'utilisateur de participer à plusieurs conversations en même temps.	Test de fonction, test d'interface usager

3.8.1.3 Le système doit permettre à l'utilisateur de créer une conversation.	Test de fonction, test d'interface usager
3.8.1.4 Le système doit permettre à l'utilisateur de rejoindre une conversation.	Test de fonction, test d'interface usager
3.8.2.1 Le système doit permettre à l'utilisateur de clavarder dans une fenêtre séparée de l'application.	Test de fonction, test d'interface usager
3.8.2.1 Le système doit permettre à l'utilisateur de clavarder dans une fenêtre séparée de l'application.	Test de fonction, test d'interface usager
3.8.3.1 Le système doit notifier l'utilisateur à l'aide d'un indicateur visuel lors de la réception de message.	Test de fonction, test d'interface usager, test de volume
3.8.3.2 Le système doit notifier l'utilisateur à l'aide d'un effet sonore lors de la réception de message.	Test de fonction, test d'interface usager
3.9.1.1 Le système doit permettre d'afficher tout l'historique d'une conversation à un utilisateur.	Test de fonction, test d'interface usager
3.10.1.1 Le système doit permettre à un dessinateur de dessiner un trait.	Test de fonction, test de performance
3.10.1.2 Le système doit permettre à un dessinateur d'effacer un trait au complet.	Test de fonction, test d'interface usager
3.10.1.3 Le système doit permettre à un utilisateur d'effacer librement une partie du trait	Test de fonction, test d'interface usager
3.10.1.4 Le système doit permettre au dessinateur de sélectionner la pointe d'un trait.	Test de fonction, test d'interface usager
3.10.1.5 Le système doit permettre au dessinateur de sélectionner la taille d'un trait.	Test de fonction, test d'interface usager
3.10.1.6 Le système doit permettre au dessinateur de sélectionner la couleur d'un trait.	Test de fonction, test d'interface usager
3.10.1.7 Le système doit ajouter un utilisateur à une conversation lorsque la partie débute.	Test de fonction, test d'interface usager
3.10.1.9 Le système doit permettre à un joueur virtuel de tracer un dessin.	Test de fonction, test de performance
3.10.1.10 Le système doit modifier la vitesse de	Test de fonction

dessin d'un joueur virtuel en fonction de la difficulté d'un jeu.	
3.10.1.11.1 Le système doit permettre à un joueur, qui regarde un dessin, d'écrire le mot correspondant au dessin.	Test de fonction, test d'interface usager
3.10.1.11.2 Le système doit donner des points à un joueur qui a deviné le mot correspondant au dessin affiché.	Test de fonction, test d'interface usager
3.10.1.11.3 Le système doit jouer un effet sonore lorsque le mot est deviné par un joueur.	Test de fonction, test d'interface usager
3.10.1.11.4 Le système doit donner des points à un joueur qui dessine en fonction du nombre de joueurs ayant deviné le mot correspondant au dessin.	Test de fonction, test d'interface usager
3.10.1.11.5 Le système doit changer de dessinateur à chaque tour.	Test de fonction, test d'interface usager
3.10.1.11.6 Le système doit changer de dessinateur si tous les autres joueurs ont deviné le mot correspondant au dessin.	Test de fonction, test d'interface usager
3.11.1.12.3 Le système doit définir le nombre d'essais maximum collectif selon la difficulté d'un jeu.	Test de fonction, test d'interface usager
3.12.1.2.6 Le système doit afficher le nombre d'essais restant à l'équipe pour le dessin en cours.	Test d'interface usager
3.11.1.1.3 Le système doit changer le dessin une fois le nombre d'essais maximum pour deviner les mots est atteint.	Test de fonction, test d'interface usager
3.11.1.1.8 Le système doit ajouter du temps à la partie en fonction du niveau de difficulté du dessin lorsque le mot est trouvé.	Test de fonction, test d'interface usager
3.11.1.1.10 Le système doit ajouter un temps bonus au joueur lorsqu'il devine un mot correctement.	Test de fonction, test d'interface usager
3.11.1.1.11 Le système doit permettre à l'utilisateur de demander un indice.	Test de fonction, test d'interface usager
3.11.1.1.12 Le système doit pénaliser l'utilisateur en retirant du temps lorsque l'utilisateur demande	Test de fonction, test d'interface usager

un indice.	
3.11.1.2.3 Le système doit définir le nombre d'essais maximum collectif selon la difficulté d'un jeu.	Test de fonction, test d'interface usager
3.11.1.2.4 Le système doit changer le dessin une fois le nombre d'essais maximum pour deviner les mots est atteint.	Test de fonction, test d'interface usager
3.11.1.2.6 Le système doit afficher le nombre d'essais restant à l'équipe pour le dessin en cours.	Test de fonction, test d'interface usager
3.11.1.2.7 Le système doit ajouter du temps à la partie en fonction du niveau de difficulté du dessin lorsque le mot est trouvé.	Test de fonction, test d'interface usager
3.11.1.2.8 Le système doit ajouter des points à l'équipe lorsqu'un mot est trouvé.	Test de fonction, test d'interface usager
3.11.1.2.9 Le système doit diminuer de 1 la quantité d'essais restante à l'équipe pour chaque tentative de mot échouée.	Test de fonction, test d'interface usager
3.11.1.2.10 Le système doit ajouter un temps bonus aux joueurs lorsqu'il devine un mot correctement.	Test de fonction, test d'interface usager
3.11.1.2.12 Le système doit pénaliser l'utilisateur en retirant du temps lorsque l'utilisateur demande un indice.	Test de fonction, test d'interface usager
3.12.1.1 Le système doit pouvoir traduire l'interface utilisateur de l'application en fonction de la langue du système (Anglais ou Français) lors de la première utilisation.	Test de fonction, test d'interface usager
3.12.1.2 Le système doit pouvoir traduire l'interface utilisateur de l'application grâce à un bouton.	Test de fonction, test d'interface usager
3.12.1.3 Le système doit traduire le message envoyé par un joueur virtuel en fonction de la langue de la partie.	Test de fonction, test d'interface usager
4.1.2.1 La création d'une nouvelle partie doit se faire en moins de 1 minute.	Test d'interface usager

4.2.1.1 Le système doit être disponible 99.9% du temps par année.	Test de performance, test de charge, test de volume, test de stress
4.1.2.2 La création d'un nouveau compte doit prendre moins de 1 minute et 30 secondes.	Test d'interface usager
4.1.2.3 La connexion au compte doit se faire en moins de 5 secondes.	Test de performance, test de charge, test de volume
4.1.2.4 L'utilisateur doit pouvoir utiliser le clavier en moins de 30 secondes.	Test d'interface usager
4.1.2.5 L'utilisateur doit pouvoir consulter son profil en moins de 10 secondes.	Test d'interface usager
4.1.2.6 La durée du tutoriel doit être de moins de 2 minutes	Test d'interface usager
4.1.2.7 L'utilisateur doit changer le thème de l'application en moins de 15 secondes.	Test d'interface usager
4.1.3.1 Les éléments de l'interface utilisateur de chaque plateforme doivent utiliser le même style.	Test d'interface usager
4.2.1.4 Le système doit redémarrer le serveur s'il tombe en panne.	Tests d'échec/récupération
4.3.1.1 Le temps requis pour recevoir le message d'un autre joueur sur un canal de discussion est d'une seconde au maximum.	Test de performance, test de volume, test de charge
4.3.1.3 Le temps requis pour que le système valide le nom d'utilisateur et le mot de passe d'un utilisateur est d'une seconde au maximum.	Test de performance, test de volume, test de charge
4.3.1.4 Le serveur doit fournir une expérience de jeu sans latence visible.	Test de performance, test de charge, test de volume
4.3.1.5 Le serveur doit supporter simultanément la connexion de 8 utilisateurs.	Test de performance, test de charge, test de volume, test de stress
4.3.1.6 Le serveur doit répondre en moins de 1.5 seconde pour les requêtes REST	Test de performance, test de charge, test de volume
4.3.2.3 Le client léger doit pouvoir changer le thème de l'interface sans ajouter de délai aux autres tâches.	Test de performance

4.6.1.4 Les journaux doivent être libres d'informations confidentielles.	Test de sécurité et de contrôle d'accès
4.6.1.5 Les journaux ne doivent pas contenir de secret comme des mots de passe.	Test de sécurité et de contrôle d'accès
4.6.2.1 Les mots de passe doivent être cryptés à l'aide de l'algorithme bcrypt dans la base de données pour assurer leur sécurité et leur confidentialité.	Test de sécurité et de contrôle d'accès
4.6.2.3 Les jetons sont ensuite utilisés pour authentifier les utilisateurs déjà connectés.	Test de fonction
4.6.2.4 Les jetons sont utilisés pour toutes les autres requêtes qui nécessitent de l'authentification.	Test de sécurité et de contrôle d'accès, test de fonction
4.6.3.1 Les jetons doivent être générés avec des fonctions de génération d'octets aléatoire avec une entropie élevée.	Test de fonction, test de sécurité et de contrôle d'accès

3. Stratégie de test

3.1. Types de test

3.1.1. Tests de fonction

Objectif de test:	S'assurer que l'implémentation des fonctionnalités correspond à ce qui était demandé. Il faut que les fonctionnalités retournent les bonnes données pour un scénario particulier.
Technique:	Il faudra vérifier que pour chaque cas d'utilisation et fonction, les données renvoyées ou manipulées sont valides. Il faudra aussi vérifier que lorsque des données invalides sont rentrées, les messages d'erreurs ou d'avertissement sont appropriés.
Critère de complétion:	Tous les résultats attendus doivent concorder avec nos résultats attendus, le SRS et le complément pédagogique. Les défauts rencontrés doivent également être corrigés.
Considérations spéciales:	Lors du test de fonction, il faudra s'assurer de couvrir toutes les branches et de valider les résultats obtenus lors de l'entrée de données invalides.

3.1.2. Tests d'interface usager

Objectif de test:	Il faut vérifier que l'utilisateur peut naviguer aux différentes vues de l'application grâce à des actions spécifiques. L'interface doit aussi afficher des dialogues d'erreur lorsque nécessaire. Puis, les données voulues doivent être affichées correctement. Finalement, l'interface doit être conviviale et intuitive (bonne expérience utilisateur).
Technique:	Naviguer entre différentes vues et s'assurer que nous sommes redirigés à la bonne vue. Effectuer des actions qui sont empêchées par le client ou par le serveur et s'assurer qu'un dialogue d'erreur est affiché. Vérifier que les vues affichent les données spécifiées par le SRS. Évaluer l'expérience utilisateur de sujets.
Critère de complétion:	La navigation entre chaque vue a le comportement attendu. Chaque action invalide fait ouvrir un dialogue d'erreur avec une description de l'erreur. Chaque vue contient les informations spécifiées dans le SRS. L'expérience utilisateur évaluée doit satisfaire les critères de satisfaction définis.
Considérations spéciales:	Chaque interface doit être disponible en anglais et en français.

3.1.3. Tests d'intégrité des données

Objectif de test:	Il faut s'assurer que les données en BD soient valides et qu'elles ne soient pas corrompues lors d'accès en lecture et en écriture à la base de données.
Technique:	Faire des accès à la base de données en écriture pour y insérer des données valides ou invalides. Il faudra ensuite vérifier si la base de données a été modifiée comme désiré. Les ajouts de données valides devraient se retrouver en BD en faisant des lectures des tables. Les ajouts de données invalides devraient être refusés.
Critère de complétion:	Des accès concurrents à la base de données qui ajoutent des données valides et invalides doivent laisser la BD dans un état cohérent et attendu à la fin des tests.

3.1.4. Tests de performance

Objectif de test:	Tester la performance de l'application et des fonctionnalités. Le but étant de s'assurer que l'application répond à nos attentes et qu'elle soit la plus optimisée possible.
Technique:	Mesurer les temps de traitements et de réponse pour des opérations coûteuses. Aussi, tester le comportement de l'ordinateur, pourcentage de CPU, RAM, etc.
Critère de complétion:	Les performances respectent les critères établis et la latence de l'application respecte les critères de base.
Considérations spéciales:	Les tests seront différents sur les deux clients. En effet, les caractéristiques de la tablette sont différentes d'un portable classique.

3.1.5. Tests de charge

Objectif de test:	Tester des séquences et cas d'utilisation utilisant les différentes fonctionnalités
Technique:	Établir une liste d'étapes à effectuer et tester notamment des séquences «sensibles» ou qui pourraient causer des bogues potentiels.
Critère de complétion:	Les séquences exécutées doivent donner le comportement attendu.

3.1.6. Tests de stress

Objectif de test:	Tester l'application et la pousser dans des cas limites. Le but étant de pouvoir s'assurer que l'application ait un bon comportement dans des situations «stressantes».
Technique:	Varié l'utilisation de clients, utiliser le plus de clients légers/lourds possible.
Critère de complétion:	L'application fonctionne de la même façon même dans des situations limites. Les performances, et utilisations de ressources doivent rester dans un ordre de grandeur raisonnable.

3.1.7. Tests de volume

Objectif de test:	En complément des tests de stress, le but est de confronter l'application a des gros volumes de données.
Technique:	Utiliser plusieurs clients en parallèle qui vont exécuter des séquences avec de lourds volumes de données.
Critère de complétion:	L'application ne plante pas et donne des résultats satisfaisants face à ce genre de situations.

3.1.8. Tests de sécurité et de contrôle d'accès

Objectif de test:	S'assurer que les informations de l'utilisateur soient en sécurité.
Technique:	Tenter d'accéder à des informations sensibles de l'utilisateur grâce à des outils de sécurité.
Critère de complétion:	L'accès d'informations sensibles a échoué.

3.1.9. Tests d'échec/récupération

Objectif de test:	S'assurer que le logiciel soit stable même en cas de bogue, défaillance.
Technique:	Redémarrer le serveur pendant une session courante de clients

Critère de complétion:	L'application déconnecte l'utilisateur correctement, et l'état est réinitialisé correctement.
Considérations spéciales:	Les tests seront différents sur les deux clients. En effet, les caractéristiques (lifecycle par exemple) de la tablette sont différentes d'un portable classique.

3.2. Outils

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

Type de test	Outil
Test de fonction	Débogueur Visual Studio, Débogueur Android Studio, Débogueur Goland
Test d'interface usager	Windows 10, Tablette Galaxy Tab A, Émulateur Android 9.0
Test d'intégrité des données	Datagrip (visualiser le contenu de la base de données)
Test de performance	Windows 10, Tablette Galaxy Tab A, Émulateur Android 9.0 Profileurs Visual Studio et Android Studio Go benchmark
Test de charge	Go Benchmark
Test de stress	Go race
Test de volume	Windows 10, Tablette Galaxy Tab A, Émulateur Android 9.0
Test de sécurité et de contrôle d'accès	Kali Linux, BurpSuite
Test de d'échec/récupération	Windows 10, Tablette Galaxy Tab A, Émulateur Android 9.0

4. Ressources

4.1. Équipe de test

Rôle	Membre de l'équipe	Responsabilités
Test Manager	Pascal-Alexandre Morel	Il fournit de l'aide administrative sur les tests. Il s'assure que les tests ont été effectués et il fournit de la direction lorsque nécessaire.
Test designer du client lourd	Pascal-Alexandre Morel	Il identifie, implémente et priorise les cas d'utilisation de tests du client lourd.
Testeur du client lourd	Cédric Tessier	Il exécute les tests et il enregistre les résultats.

Test designer du client léger	Philippe Côté	Il identifie, implémente et priorise les cas d'utilisation de tests du client léger.
Testeur du client léger	Samuel Saito-Gagné	Il exécute les tests et il enregistre les résultats.
Test designer du serveur	Martin Pouliot	Il identifie, implémente et priorise les cas d'utilisation de tests du client lourd.
Testeur du serveur	Allan Beddouk	Il exécute les tests et il enregistre les résultats.

4.2. Système

Les tests du client lourds seront exécutés sur deux ordinateurs différents afin d'assurer une meilleure compatibilité du logiciel. Le premier ordinateur est un ThinkPad Lenovo X1 Carbon 7th Gen. Son système d'exploitation est Windows 10 et l'ordinateur possède 16 GB de RAM ainsi qu'un écran 1080p.

Le deuxième ordinateur est une machine virtuelle équipée de 4 coeurs et 4 fils d'exécution (3.2 à 4.1 GHz). Son système d'exploitation est Windows 10 et elle possède 12 GB de RAM. La machine est associée à un écran 1440p afin de vérifier que l'interface est dynamique en fonction de la résolution.

Les tests du client léger seront effectués sur une Galaxy Tab A (2019) qui correspond à la même tablette que celle utilisée pour l'évaluation finale. Elle possède 2GB de RAM, une résolution de 1920x1200 ainsi qu'un écran de 10.1".

Les tests du serveur sont exécutés sur une machine virtuelle Ubuntu 18.04 LTS sur un serveur de Polytechnique Montréal et sur une machine virtuelle Azure pour la remise finale. Dans le cas, où une erreur survient sur le serveur, celle-ci se retrouvait dans les journaux de l'application et l'équipe était automatiquement avisée du problème. Également, plusieurs tests ont été exécutés sur un Macbook Pro 2014 avec 8GB de RAM.

5. Jalons du projet

Jalon	Effort	Date de début	Date de fin
Planification des cas de tests	3h	8 avril 2020	8 avril 2020
Conception des cas de tests	3h	9 avril 2020	9 avril 2020
Implémenter les cas de tests	3h	10 avril 2020	10 avril 2020
Exécuter les tests	6h	11 avril 2020	12 avril 2020
Enregistrer les résultats des tests	1h	12 avril 2020	12 avril 2020