

**Projet 3 : Fais-moi un dessin
Spécifications des requis du système (SRS)**

Version 1.4

Historique des révisions

Date	Versi on	Description	Auteur
2020-01-19	1.0	Compléter partie interfaces et caractéristiques usagers.	Nicole Joyal
2020-01-21	1.1	Rédaction des exigences fonctionnelles et non fonctionnelles	Équipe - 6
2020-01-23	1.2	Rédaction des exigences fonctionnelles	Karima
2020-01-24	1.3	Révision du document	Équipe - 6
2020-01-26	1.4	Mise en page et correction finale	Équipe - 6
2020-02-05	1.5	Révision du document suite à la correction du cours de LOG3000	Amar Ghaly

Table des matières

1. Introduction	5
1.1. But	5
1.2. Définitions, acronymes et abréviations	5
1.3. Vue d'ensemble du document	5
2. Description globale	6
2.1. Caractéristiques des usagers	6
2.2. Interfaces	6
2.2.1. Interfaces usagers	6
2.2.2. Interfaces matérielles	6
2.2.3. Interfaces logicielles	7
2.2.4. Interfaces de communication	7
2.3. Contraintes générales	7
2.4. Hypothèses et dépendances	7
3. Exigences fonctionnelles	7
3.1. Client lourd	7
3.1.1. Accueil	7
3.1.2. Clavardage - Intégration	8
3.1.3. Clavardage - Canaux de discussion	8
3.1.4. Profil utilisateur et historique	9
3.1.5. Sauvegarde du dessin	9
3.1.6. Mode de jeu - Dessin libre	9
3.1.15. Personnalité des joueurs virtuels	15
3.1.16. Tutoriel	15
3.1.17. Effets sonores et visuels	15
3.1.18. Intégrité du langage	16
3.1.19. Classement	16
3.1.20. Dévoilement des lettres	16
3.2. Client léger	17
3.2.1. Clavardage	17
4. Exigences non fonctionnelles	17
4.1. Utilisabilité	17
4.1.1. Apprentissage intuitif	17
4.2. Fiabilité	17
4.2.1. Robustesse vis-à-vis les paramètres utilisateurs	17
4.2.5. Robustesse vis-à-vis l'envoi des données client-serveur	18
4.2.6. Temps moyen jusqu'à la réparation	18

4.2.7. Disponibilité	18
4.3. Performance	18
4.3.1. Temps de réponse lors du clavierage	18
4.3.2. Temps de réponse lors d'une recherche	18
4.3.3. Capacité de connexions simultanées sur le serveur	18
4.4. Maintenabilité	18
4.4.1. Normes de codage	18
4.4.2. Normes de gestion de versions	18
4.4.3. Convention de nommage	18
4.5. Contraintes de conception	19
4.5.1. Langage de programmation	19
4.5.2. Plateforme de développement	19
4.6. Sécurité	19
4.6.1. Accès aux données utilisateur	19
4.7. Normes applicables	19
4.7.1. Normes par rapport aux données utilisateur	19

Spécifications des requis du système (SRS)

1. Introduction

1.1. But

Le SRS décrit le comportement externe d'une application. Il décrit aussi les exigences non fonctionnelles, les contraintes de conception, ainsi que les autres facteurs nécessaires à la description complète des exigences du logiciel à développer.

1.2. Définitions, acronymes et abréviations

Client lourd – Application avec fonctionnalités complexes et hébergées localement

Client léger – Application avec fonctionnalités simples qui dépendent du serveur pour le traitement et qui s'exécute sur une tablette

Serveur - Machine de traitement des requêtes effectuées du côté client

Manche – Unité de jeu ayant une limite de temps

Partie – Nombre de manches déterminées par l'hôte

Hôte – Créateur de la partie

Joueur virtuel – Joueur contrôlé par l'application pouvant occuper uniquement le rôle de joueur dessinateur

Joueur juge – Rôle pouvant être occupé par un joueur dans le mode de jeu inversé

Il est celui qui choisit le mot que les joueurs dessinateurs devront dessiner et détermine le top 3 des meilleurs dessins.

Joueur dessinateur – Rôle pouvant être occupé par un joueur dans le mode de jeu inversé.

Il est celui qui dessine le mot choisi par le joueur juge pour la manche.

Jeu – Mot ou expression à deviner par les joueurs lors d'une manche

Lobby – Interface de création d'une partie

1.3. Vue d'ensemble du document

Ce document sera composé de 4 sections distinctes. La première section décrivant ce qu'est un SRS ainsi que le vocabulaire utilisé. La deuxième partie traitera globalement du type de logiciel qui

sera développé et des technologies qui seront utilisées. La troisième partie abordera les exigences fonctionnelles des clients lourds et légers. Finalement, on décrira les exigences non fonctionnelles dans la quatrième partie.

2. Description globale

Le logiciel à développer est une application mobile de type amusement. Le jeu étant de style devinette, les joueurs devront soit tenter de dessiner le mot présenté, soit tenter de deviner le mot représenté par un dessinateur. L'application peut être jouée sur des ordinateurs avec comme système d'exploitation Windows et des tablettes avec comme système d'exploitation Android.

2.1. Caractéristiques des usagers

Les utilisateurs potentiels de l'application seront typiquement des adolescents ou de jeunes adultes et étudiants en génie logiciel. Ils seront, par conséquent, à l'aise avec les fonctionnalités d'un appareil mobile et confortable avec des jeux rapides.

2.2. Interfaces

2.2.1. Interfaces usagers

Il y aura trois interfaces usagers principales à développer. Le client lourd qui représente l'application sur un ordinateur, le client léger représente un appareil portatif tel une tablette et le serveur n'est qu'en fait un tunnel SSH permettant la connexion via le terminal. L'interface usager du client lourd sera développée en WPF (.NET C#), le client léger en Kotlin et le serveur utilisera Node.js et sera codé en TypeScript. Afin d'être plus ergonomique et améliorer l'expérience utilisateur, l'interface du client léger sera adaptée pour supporter les fonctions d'un écran tactile.

Lors de l'accès aux interfaces de clients léger et lourd, l'utilisateur doit d'abord accéder à l'interface de connexion. Si ce dernier ne possède pas de compte, il passe à l'interface de création de comptes. Puis, il pourra accéder au Lobby. À partir du lobby, l'utilisateur peut commencer à jouer, soit sélectionner un mode de jeu, créer un jeu ou participer à un canal de discussion.

2.2.2. Interfaces matérielles

En ce qui concerne le client lourd, les interfaces matérielles supportées seront les écrans d'ordinateur, le clavier et la souris. Pour le client léger, les interfaces matérielles supportées sont les tablettes Android qui possèdent un écran tactile. Dans les deux cas, il est primordial d'avoir un accès réseau à Internet câblé ou sans-fil et un système sonore fonctionnel sur chaque appareil.

2.2.3. Interfaces logicielles

Les interfaces logicielles supportées par l'application sont les systèmes d'exploitation Windows 10 sur ordinateur (client lourd) et Android de version supérieure à 5.1 (client léger). Du côté serveur, il sera hébergé sur Google Cloud Platform (ou Microsoft Azure) et aura une connexion à une base de données MongoDB de type NoSQL.

2.2.4. Interfaces de communication

La communication entre le client lourd et le client léger s'effectuera sur le réseau de Polytechnique Montréal, tandis que la communication entre le client léger et le serveur se fera avec le réseau sans fil *eduroam*.

2.3. Contraintes générales

On estime que la mémoire physique (RAM) du client lourd devra être d'au moins 1 Gb. Il devra aussi avoir assez de mémoire de stockage pour enregistrer les dessins et les messages, on estime au moins 500 Mb. Pour ce qui est de l'application mobile, la mémoire physique pourra être un peu plus limitée, on estime environ 500 Mb. Pour ce qui est de la mémoire physique du client léger, 400 Mb d'espace libre devrait être suffisant. Pour s'assurer du bon fonctionnement du logiciel, on devrait limiter le nombre de clients à 100 personnes.

2.4. Hypothèses et dépendances

Le logiciel sera dépendant d'une base de données (ex. MongoDB) et d'un serveur hébergé sur le service infonuagique de Microsoft Azure, sans lesquels notre application ne pourra fonctionner. L'application nécessite une connexion Internet assez performante afin de communiquer rapidement avec le serveur. Puisque nous traitons des images, nous supposons que la connexion Internet sera suffisamment performante pour envoyer et recevoir les données d'image en temps réel.

3. Exigences fonctionnelles

Parmi les exigences décrites ci-dessous, vous retrouverez des exigences fonctionnelles essentielles et souhaitables. Nous utiliserons alors les notations E et S pour les décrire de façon respective.

3.1. Client lourd

3.1.1. Accueil

- 3.1.1.1. Le système doit permettre à l'utilisateur de se connecter sur la page d'accueil. (E)
- 3.1.1.2. Le système doit permettre à l'utilisateur de se créer un compte. (E)
 - 3.1.1.2.1 Le système doit permettre à l'utilisateur d'entrer son prénom. (E)
 - 3.1.1.2.2 Le système doit permettre à l'utilisateur d'entrer son nom. (E)
 - 3.1.1.2.3 Le système doit permettre à l'utilisateur d'entrer son pseudonyme. (E)
 - 3.1.1.2.4 Le système doit permettre à l'utilisateur d'entrer son mot de passe. (E)

3.1.1.2.5 Le système doit permettre à l'utilisateur de sélectionner un avatar parmi une sélection. (E)

3.1.1.3. Le système doit avertir l'utilisateur s'il y a une erreur de connexion au serveur. (E)

3.1.1.4. Le système doit avertir l'utilisateur s'il y a une erreur d'authentification. (E)

3.1.2. Clavardage - Intégration

3.1.2.1. Le système doit permettre à l'utilisateur de clavarder dès sa connexion à l'application. (E)

(E) 3.1.2.2. Le système doit permettre à l'utilisateur d'accéder au mode de clavardage intégré.

3.1.2.3. Le système doit permettre à l'utilisateur d'accéder au mode de clavardage fenêtré (fenêtre séparée de l'application). (S)

3.1.2.4. Le système doit permettre à l'utilisateur de fermer la fenêtre de clavardage. (S)

3.1.2.5. Le système doit permettre à l'utilisateur de passer au mode intégré lorsque la fenêtre de clavardage est fermée. (S)

3.1.2.6. Le système doit permettre à l'utilisateur de clavarder uniquement avec les autres clients en ligne. (E)

3.1.2.7. Le système doit permettre à l'utilisateur de clavarder avec des utilisateurs en dehors d'une partie. (E)

3.1.3. Clavardage - Canaux de discussion

3.1.3.1. Le système doit permettre à un utilisateur de créer un canal de discussion. (E)

3.1.3.2. Le système doit permettre à un utilisateur de clavarder dans un canal de discussion exclusif à une partie de jeu. (E)

3.1.3.3. Le système doit permettre à un utilisateur de créer un canal de discussion. (E)

3.1.3.4. Le système doit permettre à un utilisateur de choisir la visibilité du canal de discussion (public ou privé). (E)

3.1.3.5. Le système doit permettre à un utilisateur de choisir les utilisateurs qui seront invités à un canal de discussion privé. (E)

3.1.3.6. Le système doit permettre à un utilisateur de joindre un canal de discussion en choisissant un nom dans une liste. (E)

3.1.3.7. Le système doit permettre à un utilisateur de joindre un canal de discussion en choisissant un nom par recherche par mots. (E)

3.1.3.8. Le système doit permettre à un utilisateur de faire partie de plusieurs canaux de discussion. (E)

3.1.3.9. Le système doit permettre à un utilisateur d'inviter un autre client à un canal de discussion. (E)

3.1.3.10. Le système doit permettre à un utilisateur de quitter un canal de discussion. (E)

3.1.3.11. Le système doit permettre à un utilisateur d'accéder à un canal de discussion principal contenant tous les utilisateurs. (E)

3.1.3.12. Le système doit permettre à un utilisateur de voir l'historique complet d'un canal

de

discussion.

3.1.3.13. Le système doit permettre la distinction des utilisateurs dans un canal de discussion à l'aide de leur pseudonyme. (E)

3.1.4. Profil utilisateur et historique

3.1.4.1. Le système doit permettre à l'utilisateur de s'authentifier au serveur. (E)

3.1.4.2. Le système doit vérifier l'unicité du profil de l'utilisateur par son pseudonyme. (E)

3.1.4.3. Le système doit permettre à l'utilisateur de modifier son nom, prénom, mot de passe et avatar. (E)

3.1.4.4. Le système doit permettre à l'utilisateur de voir l'avatar et le pseudonyme des autres clients dans l'interface de jeu. (E)

3.1.4.5. Le système doit permettre à l'utilisateur de consulter ses données statistiques (nombre parties jouées, pourcentage de victoires, temps moyen d'une partie, temps total passé à jouer). (E)

3.1.4.6. Le système doit permettre à l'utilisateur de consulter un historique détaillé présentant des dates et heures de connexion/déconnexion. (E)

3.1.4.7 Le système doit permettre à l'utilisateur de consulter un historique des parties jouées (date, heure, nom des joueurs et résultats pour chaque partie). (E)

3.1.4.8. Le système doit permettre à l'utilisateur de supprimer son compte de façon définitive. (S)

3.1.4.9. Le système doit permettre à l'utilisateur de confirmer la suppression de son compte. (S)

3.1.5. Sauvegarde du dessin

3.1.5.1. Le système permet à l'utilisateur de sauvegarder son dessin localement. (S)

3.1.5.2. Le système offre les formats de sauvegarde suivants : BMP, JPG et PNG. (S)

3.1.6.3. Le système offre cette option uniquement en mode dessin libre. (S)

3.1.6. Mode de jeu - dessin libre

3.1.6.1. Le système doit permettre à l'utilisateur de dessiner sur une zone de dessin en dehors d'une partie de jeu. (E)

3.1.6.2. Le système doit permettre la sauvegarde de son dessin. (E)

3.1.7. Mode de jeu - applicable à tous les modes

3.1.7.1. Le système doit permettre à l'hôte d'imposer une limite de temps pour une manche. (E)

3.1.7.2. Le système doit permettre à l'hôte de choisir le nombre de manches de la partie. (E)

3.1.7.3. Le système doit permettre à l'hôte de choisir s'il y a présence d'un dévoilement de lettres au cours des manches. (E)

3.1.7.4. Le système doit permettre de débiter la partie lorsqu'il y a assez de joueurs. (E)

3.1.7.5. Le système doit permettre au joueur de deviner le mot dessiné à l'aide d'un canal de discussion. (E)

3.1.7.6. Le système doit permettre à l'utilisateur d'affronter un joueur virtuel. (E)

3.1.4.2.6.1 Le système doit permettre au joueur virtuel de prendre uniquement le rôle de dessinateur. (E)

3.1.7.7. Le système doit afficher le compteur de nombre de points amassés par le joueur. (E)

3.1.7.8. Le système doit remettre à zéro le compteur au début d'une nouvelle partie. (E)

3.1.7.9. Le système doit incrémenter la valeur du compteur si l'utilisateur donne la bonne réponse. (E)

3.1.7.10 Le système doit décrémenter la valeur du compteur si l'utilisateur donne la mauvaise réponse. (E)

3.1.7.11. Le système doit permettre à l'utilisateur de quitter le mode de jeu. (E)

3.1.7.12. Le système doit rediriger l'utilisateur vers le lobby lorsqu'il quitte le mode de jeu. (E)

3.1.7.13. Le système doit exclure l'utilisateur du canal de discussion à la fin de la partie. (E)

3.1.7.14. Le système doit permettre à l'utilisateur d'accéder au mode de jeu à partir du lobby. (E)

3.1.7.15. Le système doit offrir une zone de dessin vierge à chaque début de partie. (E)

3.1.7.16. Le système doit permettre à l'utilisateur de visualiser le dessin d'un joueur virtuel. (E)

3.1.4.3.1.1 Le système doit se servir des images présentes dans la banque d'images créées pour les dessins du joueur virtuel. (E)

3.1.8. Mode de jeu - mêlée générale

- 3.1.8.1. Le système doit permettre à l'utilisateur de choisir parmi trois mots aléatoires à dessiner. (E)
- 3.1.8.2. Le système doit permettre à l'utilisateur de dessiner le mot choisi dans un délai de temps prédéfini. (E)
- 3.1.8.3 Le système doit permettre à l'utilisateur de dessiner à l'aide d'un outil crayon ou stylo sur la zone de dessin. (E)
- 3.1.8.4. Le système doit permettre à l'utilisateur de corriger son dessin à l'aide d'un outil d'efface. (E)
- 3.1.8.5 Le système doit permettre à l'utilisateur de modifier les paramètres du crayon ou du stylo (couleur et épaisseur). (E)
- 3.1.8.6 Le système doit permettre à l'utilisateur de créer une partie en mode de jeu mêlée générale. (E)
- 3.1.8.7. Le système doit permettre la redirection de l'utilisateur vers l'interface de jeu une fois la partie créée. (E)
- 3.1.8.8. Le système doit permettre à l'utilisateur d'alterner entre le rôle de dessinateur et deviner. (E)
- 3.1.8.9. Le système doit permettre à l'utilisateur qui devine de voir les traits du dessin en temps réel. (E)

3.1.9. Mode de jeu - sprint solo

- 3.1.9.1. Le système doit permettre à l'utilisateur de voir le temps restant de la partie. (E)
- 3.1.9.2. Le système doit permettre à l'utilisateur de voir son nombre de tentatives restant. (E)
- 3.1.9.3. Le système doit permettre de décrémenter les compteurs de tentatives en cas d'échec. (E)
- 3.1.9.4. Le système doit accorder à l'utilisateur un temps bonus lorsqu'il donne la bonne réponse. (E)
- 3.1.9.5. Le système doit permettre à l'utilisateur de passer à une prochaine manche lorsqu'il donne une bonne réponse. (E)
- 3.1.9.6. Le système doit terminer la partie lorsque l'utilisateur consomme toutes ses tentatives. (E)
- 3.1.9.7. Le système doit terminer la partie lorsque l'utilisateur consomme tout son temps accordé. (E)

3.1.10. Mode de jeu - sprint coopératif

- 3.1.10.1 Le système doit permettre à l'utilisateur de consulter les groupes existants en mode de jeu coopératif. (E)

3.1.10.2. Le système doit permettre à l'utilisateur de se joindre à un groupe existant incomplet (E).

3.1.10.3. Le système doit permettre à l'utilisateur d'accéder au canal de discussion de son groupe. (E)

3.1.10.4. Le système doit afficher qu'un groupe est complet lorsqu'il est composé de 4 joueurs. (E)

3.1.10.5. Le système doit empêcher que l'utilisateur se joigne à un groupe complet. (E)

3.1.10.6. Le système doit comptabiliser le nombre de tentatives de manière collective à un groupe. (E)

3.1.11. Mode de jeu - inversé

3.1.11.1. Le système doit permettre au joueur juge de choisir le mot à dessiner. (S)

3.1.11.2. Le système doit présenter à l'utilisateur une banque de mots parmi laquelle il peut sélectionner le mot à dessiner. (S)

3.1.11.3. Le système doit permettre au joueur juge de sélectionner un temps pour la manche. (S)

3.1.11.4. Le système doit permettre au joueur dessinateur de dessiner le mot choisi. (S)

3.1.11.5. Le système doit permettre au joueur juge de classer les trois meilleurs dessins à la fin de la manche. (S)

3.1.11.6. Le système doit permettre au joueur juge de visualiser tous les dessins lorsque le temps alloué est écoulé. (S)

3.1.11.7. Le système doit incrémenter le compteur de points en fonction de la position dans le classement défini par le joueur juge. (S)

3.1.11.8. Le système doit allouer du temps au joueur juge pour déterminer son classement. (S)

3.1.11.9. Le système doit permettre le changement du joueur juge à chaque manche. (S)

3.1.11.10. Le système doit permettre au joueur juge de visualiser les traits de dessin d'un joueur dessinateur. (S)

3.1.11.12. Le système doit limiter le temps accordé au joueur juge pour choisir les trois meilleurs dessins. (S)

3.1.11.13 Le système doit décrémenter le compteur de points du joueur juge lorsque le temps accordé est écoulé. (S)

3.1.11.14. Le système doit distribuer de manière équitable les points restants aux autres joueurs en cas d'échec de décision de la part du joueur juge. (S)

3.1.12. Mode de jeu - 1 contre 1

3.1.12.1. Le système doit permettre à l'utilisateur de se joindre à une partie de jeu un contre un si la partie est composé d'un seul joueur. (S)

3.1.12.2 Le système doit permettre au joueur virtuel d'être le seul à pouvoir dessiner le mot choisi. (S)

3.1.12.3. Le système doit incrémenter le compteur du nombre de points du joueur ayant deviné le mot en premier. (S)

3.1.13. Création d'un jeu - Manuelle

3.1.13.1. Le système doit permettre à l'utilisateur d'entrer un mot ou une expression à dessiner. (E)

3.1.13.2. Le système doit imposer à l'utilisateur d'entrer au moins un indice relié au mot ou à l'expression. (E)

3.1.13.3. Le système doit permettre à l'utilisateur d'entrer plusieurs indices associés au mot ou l'expression s'il le désire. (E)

3.1.13.4. Le système doit permettre à l'utilisateur de dessiner le mot ou l'expression qu'il a choisi dans une zone de dessin. (E)

3.1.13.5. Le système doit permettre à l'utilisateur de dessiner à l'aide d'un outil crayon ou stylo sur la zone de dessin. (E)

3.1.13.6. Le système doit permettre à l'utilisateur de corriger son dessin à l'aide d'un outil d'efface. (E)

3.1.13.7. Le système doit permettre à l'utilisateur de modifier les paramètres du crayon ou du stylo (couleur et épaisseur). (E)

- 3.1.13.8. Le système doit permettre à l'utilisateur de sélectionner un niveau de difficulté associé à son image. (E)
- 3.1.13.9. Le système doit déterminer la vitesse de traçage du dessin. (E)
- 3.1.13.10. Le système doit déterminer le temps alloué pour deviner le mot. (E)
- 3.1.13.11. Le système doit déterminer le nombre de tentatives permises. (E)
- 3.1.13.12. Le système doit déterminer le temps bonus accordé au joueur en fonction du niveau de difficulté sélectionné par l'utilisateur. (E)
- 3.1.13.13. Le système doit permettre à l'utilisateur de choisir le mode de reproduction des images effectuées par le joueur virtuel. (E)
- 3.1.13.14. Le système doit offrir le mode classique où les traits de dessins sont ajoutés dans le même ordre que lors de la création du dessin par l'utilisateur. (E)
- 3.1.13.15. Le système doit offrir le mode aléatoire où les traits de dessins sont ajoutés de manière aléatoire. (E)
- 3.1.13.16. Le système doit offrir le mode panoramique où les traits de dessins sont ajoutés dans l'ordre de leur position sur les axes cartésiens. (E)
- 3.1.13.17. Le système doit offrir le mode centré où les traits de dessins sont ajoutés dans l'ordre de leur distance au centre de l'image. (E)
- 3.1.13.18. Le système doit offrir le mode centré de l'intérieur vers l'extérieur où les traits les plus proches du centre apparaissent en premier. (E)
- 3.1.13.19. Le système doit offrir le mode centré de l'extérieur vers l'intérieur où les traits les plus loin du centre apparaissent en premier. (E)

3.1.14. Création d'un jeu - assistée

Le système doit permettre à l'utilisateur tout ce qui est offert lors de la création de jeu manuelle en plus des exigences suivantes.

- 3.1.14.1. Le système doit permettre à l'utilisateur de choisir une image bitmap de son ordinateur. (E)
- 3.1.14.2. Le système doit permettre à l'utilisateur de choisir les paramètres de conversion de l'outil Potrace. (E)

3.1.14.3. Le système doit être capable de reconnaître un fichier de type BMP, JPG et PNG.
(E)

3.1.15. Personnalité des joueurs virtuels

3.1.15.1. Le joueur virtuel doit envoyer un message en début et fin de partie. (E)

3.1.15.2. Le joueur virtuel doit pouvoir envoyer un message dans un canal de discussion.
(E)

3.1.15.3. Le joueur virtuel doit répondre aux demandes d'indice des joueurs membres de son équipe. (E)

3.1.15.4. Le joueur virtuel doit avoir une personnalité unique au cours d'une partie. (E)

3.1.16. Tutoriel

3.1.16.1. Le système doit activer le tutoriel automatiquement lors de la première connexion de l'utilisateur. (E)

3.1.16.2. Le système doit permettre à l'utilisateur de passer à la prochaine partie du tutoriel.
(E)

3.1.16.3. Le système doit permettre à l'utilisateur de visualiser un tutoriel non interactif (suite d'images et courts vidéos explicatifs de type .gif). (E)

3.1.16.4. Le système doit permettre à l'utilisateur de revoir le tutoriel au besoin. (E)

3.1.17. Effets sonores et visuels

3.1.17.1. Le système doit lancer un effet sonore. (E)

3.1.17.2. Le système doit offrir un effet sonore de victoire lorsque l'utilisateur gagne une manche. (E)

3.1.17.3. Le système doit offrir un effet sonore de défaite lorsque l'utilisateur perd une manche. (E)

3.1.17.4. Le système doit offrir un effet sonore après une tentative de résolution. (E)

3.1.17.5. Le système doit offrir un effet sonore à l'approche de la fin d'une partie. (E)

3.1.17.6. Le système doit présenter une animation de rétroaction quand l'utilisateur dessine. (E)

3.1.17.7. Le système doit présenter un effet de transition lors du passage d'une vue à une autre. (E)

3.1.17.8. Le système doit présenter un effet lors d'une victoire de partie. (E)

3.1.17.9. Le système doit permettre à l'utilisateur de modifier ses préférences d'effets sonores. (E)

3.1.17.10. Le système doit permettre à l'utilisateur de modifier le volume des effets sonores par incrément de 25 %, soit 0, 25, 50, 75 et 100%. (S)

3.1.17.11. Le système doit permettre à l'utilisateur de couper complètement le volume des effets sonores. (S)

3.1.17.12. Le système doit permettre la sauvegarde des préférences de volume définies par l'utilisateur. (S)

3.1.17.13. Le système doit activer les effets sonores par défaut. (E)

3.1.18. Intégrité du langage

3.1.18.1. Le système doit censurer les mots grossiers dans les canaux de discussion. (S)

3.1.18.2 Le système doit changer le mot grossier par un mot plus complaisant lorsqu'un mot grossier est détecté. (S)

3.1.19. Classement

3.1.19.1. Le système doit afficher le nombre de points de l'utilisateur. (E)

3.1.19.2. Le système doit mettre à jour le classement des joueurs après chaque partie. (E)

3.1.19.3. Le système doit permettre à l'utilisateur d'accéder au classement des dix meilleurs joueurs. (E)

3.1.19.4. Le système doit permettre à l'utilisateur d'accéder à sa position dans le classement des joueurs. (E)

3.1.20. Dévoilement des lettres

3.1.20.1. Le système doit dévoiler les lettres du mot durant la partie de façon aléatoire. (E)

3.1.20.2. Lorsque le temps est écoulé, le système doit avoir dévoilé 50 % du mot s'il n'a pas été deviné. (E)

3.1.20.3. Le système doit permettre à l'hôte d'activer le dévoilement de lettres lors de la création de la partie. (E)

3.1.20.4. Le système doit permettre à l'hôte de désactiver le dévoilement de lettres lors de la création de la partie. (E)

3.2. Client léger

Essentiellement, les exigences fonctionnelles du client léger sont identiques à celles du client lourd à l'exception de la section **3.1.5 Création d'un jeu**, qui est entièrement absente du client léger, et de l'intégration du clavardage.

3.2.1. Clavardage

3.2.1. Le système doit permettre à l'utilisateur de clavarder dès sa connexion à l'application. (E)

3.2.2. Le système doit permettre à l'utilisateur d'accéder au mode de clavardage intégré uniquement. (E)

3.2.3. Le système doit permettre à l'utilisateur de clavarder avec des utilisateurs en dehors d'une partie. (E)

4. Exigences non fonctionnelles

4.1. Utilisabilité

4.1.1. Apprentissage intuitif

4.1.1.1. Le logiciel doit avoir un tutoriel simple et concis.

4.1.1.1.1. Le tutoriel doit prendre en moyenne 1 à 3 minutes à compléter.

4.1.1.1.2 Le tutoriel doit être accessible en tout temps à l'utilisateur.

4.1.1.2. Le logiciel doit contenir des icônes explicatives.

4.2. Fiabilité

4.2.1. Robustesse vis-à-vis des paramètres utilisateur

4.2.1.1. Le système doit maintenir la cohérence des données d'une partie (classement et pointage) lorsqu'un changement d'informations est effectué au compte de l'utilisateur.

4.2.1.2. Le système doit maintenir la cohérence des données des canaux de discussion lorsqu'un changement d'informations est effectué au compte de l'utilisateur.

4.2.5. Robustesse vis-à-vis de l'envoi des données client-serveur

4.2.5.1. L'envoi de données incorrectes (par requête ou formulaire) doit être détecté et pris en charge correctement (message d'erreur et demande de renvoi d'information).

4.2.6. Temps moyen jusqu'à la réparation

4.2.6.1. Une panne serveur, causée par des problèmes de connexion, devrait être réglée dans les 5 heures qui suivent vu le contrat de niveau de service de Microsoft Azure et Google Cloud.

4.2.6.2. Une panne serveur causée par autre chose que des problèmes de connexion devrait être réglée dans les 5 heures qui suivent.

4.2.7. Disponibilité

4.2.7.1. Le système doit être disponible 97 % du temps.

4.3. Performance

4.3.1. Temps de réponse lors du clavardage

4.3.1.1. Un message envoyé doit apparaître dans un délai inférieur à 110 millisecondes.

4.3.2. Temps de réponse lors d'une recherche

4.3.2.1. Le système doit traiter une recherche en moins de 2 secondes.

4.3.3. Capacité de connexions simultanées sur le serveur

4.3.3.1. Le serveur doit pouvoir supporter 20 connexions simultanées.

4.4. Maintenabilité

4.4.1. Normes de codage

4.4.1.1. Le code et les commentaires doivent être rédigés en anglais.

4.4.1.2. Chaque fonctionnalité/classe doit comporter une seule responsabilité principale.

4.4.2. Normes de gestion de versions

4.4.2.1. Chaque commit doit inclure les initiales du développeur.

4.4.2.2. Les développeurs doivent intégrer sur une branche de développement avant d'intégrer sur la branche *master*.

4.4.3. Convention de nommage

4.4.3.1. Les variables et les méthodes doivent commencer par une lettre minuscule et une

lettre majuscule pour les mots suivants. Par exemple : « maxLength ».

4.4.3.2. Les constantes à nom composé devront séparer les parties de leurs noms par le caractère « _ » en plus d'utiliser seulement des lettres en majuscule. Par exemple : « MAX_LENGTH ».

4.5. Contraintes de conception

4.5.1. Langage de programmation

4.5.1.1. Le client lourd doit être codé dans les langages C#/.NET (WPF).

4.5.1.2. Le client léger doit être codé dans le langage Kotlin.

4.5.1.3. Le serveur doit être développé en utilisant Node.js et TypeScript.

4.5.2. Plateforme de développement

4.5.2.1. La plateforme de développement pour le client lourd doit être Visual Studio.

4.5.2.2. La plateforme de développement pour le client léger doit être Android Studio.

4.5.2.3. La plateforme de développement pour le serveur doit être Visual Studio Code.

4.6. Sécurité

4.6.1. Accès aux données utilisateur

4.6.1.1. Une base de données protégée doit contenir les données utilisateur.

4.6.1.2. Chaque mot de passe doit être haché dans la base de données.

4.6.1.3. Seul un administrateur a accès à la base de données.

4.6.1.4. Les statistiques liées aux profils des joueurs sont privées et contenues dans la base de données.

4.7. Normes applicables

4.7.1. Normes par rapport aux données utilisateur

4.7.1.1. Les données utilisateur sont uniquement utilisées à des fins d'amélioration de l'expérience utilisateur.

4.7.1.2. Après la période du projet, les données des utilisateurs doivent être effacées de toutes les ressources du projet.

4.7.1.3. Toutes les données de l'utilisateur doivent être effacées quand il efface son profil.