



Spécifications des requis du système (SRS)

Version 3.0

Historique des révisions

Date	Version	Description	Auteur
2020-01-21	0.1	premier jet de section 2	Jacob Dorais
2020-01-21	0.2	premier jet de section 4	François-Xavier Legault + Duc Thien Nguyen
2020-01-21	0.3	premier jet de section 3.1 et 3.2	Arthur Garnier + Jérémy Boulet
2020-01-21	0.4	premier jet de section 3.3	Pablo Chaussé
2020-01-25	0.5	Formatage, correction et quelques ajouts d'informations sur différentes sections	François-Xavier Legault
2020-01-25	0.6	Finalisation exigence souhaitable client lourd (3.3)	Arthur Garnier
2020-01-26	0.7	Ajouter des spécifications de section 3.4	Duc-Thien Nguyen
2020-01-26	1.0	revu du document et correction de l'orthographe	Jacob Dorais + François-Xavier Legault
2020-02-04	1.1	correction des exigences (3.2)	Pablo Chaussé
2020-04-09	2.0	Correction des exigences (3.1+)	Arthur Garnier
2020-04-13	3.0	Correction et finalisation	François-Xavier Legault

Table des matières

1. Introduction	4
1.1. But	4
1.2. Définitions, acronymes et abréviations	4
1.3. Vue d'ensemble du document	4
2. Description globale	4
2.1. Caractéristiques des utilisateurs	4
2.2. Interfaces	5
2.2.1. Interfaces utilisateurs	5
2.2.2. Interfaces matérielles	5
2.2.3. Interfaces logicielles	5
2.2.4. Interfaces de communication	5
2.3. Contraintes générales	5
2.4. Hypothèses et dépendances	5
3. Exigences fonctionnelles	5
3.1 Exigences essentielles du client lourd	6
3.2 Exigences souhaitables du client lourd	11
3.3 Exigences essentielles du client léger	13
3.4 Exigences souhaitables du client léger	17
4. Exigences non fonctionnelles	18
4.1. Utilisabilité	18
4.2. Fiabilité	19
4.3. Performance	19
4.4. Maintenabilité	20
4.5. Contraintes de conception	20
4.6. Sécurité	20
4.7. Exigences de la documentation utilisateur en ligne et du système d'assistance	21
4.8. Normes applicables	21

Spécifications des requis du système (SRS)

1. Introduction

1.1. But

Ce SRS décrit le comportement externe de l'application P'inturé . 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 P'inturé .

1.2. Définitions, acronymes et abréviations

Client léger : Application spécifiquement conçue pour être exécutée à partir d'appareils mobiles

(tablette, cellulaire, etc.)

Client lourd : Application spécifiquement conçue pour être exécutée sur un ordinateur personnel.

Dessin: Ensemble de traits SVG qui forme une image

Jeu: Activité de produire des dessins à tour de rôle et de deviner ce que représente le dessin

Joueur humain: Un utilisateur.

Joueur virtuel: Simulation d'un joueur fait par un ordinateur

Partie: Durée du jeu à l'issue de laquelle sera désigné le gagnant

TCP : Protocole de contrôle de transmission

Tour: Durée selon laquelle une seule personne est dessinateur

1.3. Vue d'ensemble du document

Dans la section 2 du document, on y retrouve une description générale du logiciel. Dans la section 3, on y décrit les exigences fonctionnelles du logiciel P'inturé tandis que dans la section 4, ce sont les exigences non fonctionnelles.

2. Description globale

P'inturé est un logiciel en ligne d'un jeu multijoueur contre d'autres joueurs ou contre une intelligence artificielle où à tour de rôle, les joueurs doivent faire un dessin pour représenter un mot et les autres joueurs doivent le deviner le plus rapidement possible. Le logiciel doit permettre aux utilisateurs de créer un compte et de se connecter pour jouer. Les informations seront enregistrées dans un serveur qui établira la connexion entre les joueurs.

2.1. Caractéristiques des utilisateurs

L'application concerne principalement des étudiants en ingénierie logiciel (18-25 ans). Ces utilisateurs nécessitent un accès à une tablette android ou bien un ordinateur windows connecté à

internet. Le minimum des compétences techniques requises par l'utilisateur est la capacité à accéder à l'application ainsi que lire et écrire. L'application ne nécessite aucune expérience particulière pour être bien utilisée.

2.2. Interfaces

2.2.1. Interfaces utilisateurs

L'interface du client léger adapté pour l'utilisation de l'écran tactile a été conçue avec Android Studio. Le client lourd a été conçu avec WPF. Les deux interfaces clients permettent la navigation entre différentes pages. L'interface serveur est une invite de commande.

2.2.2. Interfaces matérielles

Les interfaces matérielles standards telles que le clavier et la souris seront nécessaires pour l'utilisation du logiciel. Pour la version du client léger, l'écran tactile de l'appareil portable sera nécessaire. Un écran est nécessaire pour pouvoir visionner l'application. Un appareil sonore (peu importe le type) sera nécessaire pour pouvoir entendre les sons dans l'application.

2.2.3. Interfaces logicielles

Pour bien fonctionner, P'inturé aura besoin du logiciel Poly Paint auquel nous avons conservé certaines fonctionnalités et supprimé d'autres. Application testée exclusivement sur Windows 10. Le serveur est exécuté dans le nuage (pas de serveur local). La base de données des utilisateurs et des messages ainsi que le serveur est enregistré par la plateforme d'application infonuagique Heroku (<https://www.heroku.com/>). P'inturé a été testé sur la version Oreo d'Android (8.1). La base de donnée fonctionne avec PostgreSQL. Nous n'avons pas d'engin de conversion SVG, car nous traitons qu'avec des images SVG.

2.2.4. Interfaces de communication

Le logiciel va communiquer avec un serveur implémenté avec Node.JS en TypeScript. Le serveur aura une base de données avec PostgreSQL. La communication se fera via l'interface de socket.io qui utilise le protocole de communication TCP, ainsi que les requêtes HTTP. L'application nécessite un accès à internet des utilisateurs (Le réseau de Polytechnique).

2.3. Contraintes générales

Les parties devront pouvoir contenir de 1 à 8 joueurs. Le serveur devra être capable de supporter au moins 16 joueurs en même temps et au moins 3 parties simultanées. Le serveur devrait utiliser la communication asynchrone pour que tous les utilisateurs aient une expérience de jeux similaire.

2.4. Hypothèses et dépendances

Nous supposons que les utilisateurs seront munis d'une connexion assez rapide pour ne pas être le maillon qui ralentit son expérience de jeu.

3. Exigences fonctionnelles

Cette section décrit les exigences fonctionnelles du logiciel P'inturé .

3.1 Exigences essentielles du client lourd

3.1.1 Clavardage - Intégration

- 3.1.1.1. Le système doit permettre au clavardage d'être accessible à tout moment dès que le client est connecté au serveur.
- 3.1.1.2. Le système doit permettre à l'utilisateur de clavarder avec n'importe quelle personne en ligne
- 3.1.1.3. Le système doit permettre au clavardage d'être disponible en mode intégré.
- 3.1.1.4. Le système doit permettre clavardage d'être disponible en mode fenêtré.
- 3.1.1.5. Le système doit permettre d'alterner entre les deux modes.
- 3.1.1.6. Si la fenêtre externe est fermée, le système doit permettre de la faire réapparaître en mode intégré automatiquement.
- 3.1.1.7. Le système doit permettre de clavarder au cours d'une partie.
- 3.1.1.8. Le système doit permettre de clavarder en dehors d'une partie.
- 3.1.1.9. Le système doit permettre à l'utilisateur d'envoyer des messages contenant au maximum 144 caractères.

3.1.2 Clavardage - Canaux de discussion

- 3.1.2.1. Pendant le jeu, le système doit permettre de créer un canal privé de jeu pour que un utilisateur puisse communiquer.
- 3.1.2.2. Le système doit permettre d'avoir un canal principal par défaut.
 - 3.1.2.2.1. Le système doit permettre au canal principal par défaut de contenir tous les utilisateurs connectés.
- 3.1.2.3. Le système doit permettre à l'utilisateur de créer un canal.
- 3.1.2.4. Le système doit permettre à l'utilisateur de joindre un canal à partir d'une liste.
- 3.1.2.5. Le système doit permettre à l'utilisateur d'être dans plusieurs canaux en même temps.
- 3.1.2.6. Le système doit permettre à l'utilisateur de changer de canal.
- 3.1.2.7. Le système doit permettre à l'utilisateur de quitter un canal rejoint.
- 3.1.2.8. Le système doit permettre à l'utilisateur de consulter l'historique de message d'un canal rejoint.

3.1.3 Profil utilisateur et historique

- 3.1.3.1. Le système doit permettre à un utilisateur de créer un compte à l'aide d'un pseudo et

d'un mot de passe

3.1.3.2. Le système doit permettre à un utilisateur de créer un mot de passe d'au moins 4 caractères et de maximum 20, contenant chiffre, lettre et caractères spéciaux.

3.1.3.3. Le système doit permettre à un utilisateur de s'authentifier à un profil créé.

3.1.3.4. Le système doit permettre à seul utilisateur de se connecter par profil.

3.1.3.5. Le système doit permettre à l'utilisateur d'avoir un profil qui contient prénom, nom et pseudo.

3.1.3.6. Le système doit permettre à l'utilisateur de choisir son avatar pour son profil selon une liste proposée par défaut

3.1.3.7. Le système doit permettre à l'utilisateur de choisir son avatar pour son profil en important un fichier jpg ou png de son ordinateur.

3.1.3.8. Le système doit permettre à un utilisateur de consulter les statistiques d'utilisation du jeu, sur son profil.

3.1.3.9. Le système doit permettre des statistiques pour chaque profil utilisateur, contenant le nombre de parties jouées, le pourcentage de victoires, le temps moyen d'une partie, et le temps total passé à jouer

3.1.3.10. Le système doit permettre à un utilisateur de consulter le pseudo et l'avatar d'un profil d'un autre utilisateur

3.1.3.11. Le système doit enregistrer l'historique détaillé du profil contenant un relevé de ses connexions et déconnexions avec dates et heures correspondantes.

3.1.3.12. Le système doit permettre d'afficher à l'utilisateur un historique des parties jouées sur son profil, affichant l'heure, la date, le temps de jeu de la partie et le résultat.

3.1.4 Création d'un jeu

3.1.4.1. Le système doit permettre aux utilisateurs de créer un jeu.

3.1.4.2. Le système doit permettre à l'utilisateur d'indiquer un indice par rapport au mot à deviner.

3.1.4.3. Le système doit permettre à l'utilisateur créant un jeu de dessiner un mot qui pourra être deviné au cours d'une partie.

3.1.4.4. Le système doit permettre d'enregistrer le dessin fait par l'utilisateur pour pouvoir le redessiner par un joueur virtuel.

3.1.4.5. Le système doit permettre à l'utilisateur d'avoir un aperçu de l'image comme s'il s'agissait du joueur virtuel qui dessinait.

3.1.4.6. Le système doit permettre à l'utilisateur de choisir la difficulté du jeu créer.

3.1.4.7. Le système doit permettre de définir la vitesse de dessin des joueurs virtuels, le temps

alloué et le nombre de tentatives alloué au joueur, selon la difficulté.

3.1.4.8. Le système doit permettre les modes de dessins suivants: classique, aléatoire, panoramique et centré.

3.1.4.8.1. Le système doit permettre au mode aléatoire de faire apparaître les traits du dessin d'un joueur virtuel de manière aléatoire.

3.1.4.8.2. Le système doit permettre au mode panoramique doit faire apparaître les traits du dessin d'un joueur virtuel selon leur position sur l'axe cartésien.

3.1.4.8.3. Le système doit permettre au mode panoramique de proposer 4 directions (droite gauche, gauche droite, haut bas, bas haut)

3.1.4.8.4. Le système doit permettre au mode centré de faire apparaître les traits du dessin d'un joueur virtuel dans l'ordre de leur distance par rapport au centre de l'image

3.1.4.9. Le système doit permettre à l'utilisateur d'importer une image (bmp, jpg, png) au lieu d'un dessin manuel de l'utilisateur pour la création d'un jeu.

3.1.4.10. Le système doit permettre de produire un dessin à partir de l'image importée.

3.1.4.10.1. Le système doit permettre à l'utilisateur de choisir les différentes variables de l'engin de conversion utilisé.

3.1.4.11. Le système doit permettre à l'utilisateur de pouvoir sélectionner l'ordre de séquence d'apparition d'un trait du dessin selon le mode défini au niveau de difficulté précédent.

3.1.4.12. Le système doit permettre à l'utilisateur de choisir l'ordre de séquence parmi aléatoire, panoramique et centré seulement, lors de la création d'un jeu en important une image de son ordinateur.

3.1.4.13. Le système doit permettre à l'utilisateur d'utiliser la banque de dessin «Quick Draw» de Google pour choisir l'image et le mot qui serviront à créer un jeu.

3.1.4.14. Le système doit permettre à l'utilisateur de choisir son image sur «Quick Draw» en utilisant un bouton qui permet de changer le mot et le dessin

3.1.4.15. Le système doit permettre à l'utilisateur de fournir un indice lors de la création d'un jeu.

3.1.4.16. Le système doit permettre à l'utilisateur de choisir un mode de dessin lors de la création d'un jeu.

3.1.5 Personnalité des joueurs virtuels

3.1.5.1. Le système doit permettre aux joueurs virtuels d'avoir une personnalité différente, tirée d'une liste prédéfinie

3.1.5.2. Le système doit permettre aux joueurs virtuels dans une même partie d'avoir une personnalité différente.

3.1.5.3. Le système doit permettre à un joueur virtuel de publier un message au début de la

partie, dans le canal de la partie.

3.1.5.4. Le système doit permettre à un joueur virtuel de publier un message à la fin de chaque tour dans le canal de la partie.

3.1.5.5. Le système doit permettre à un joueur virtuel de donner un indice lorsque demandé

3.1.6 Modes de jeu

3.1.6.1. Après la connexion, le système doit permettre de rediriger l'utilisateur vers un lobby

3.1.6.2. Le système doit permettre à un utilisateur dans un lobby de choisir un mode de jeu.

3.1.6.3. Le système doit présenter les groupes publics existants d'un mode de jeu.

3.1.6.4. Le système doit permettre à l'utilisateur de joindre un groupe public existant

3.1.6.5. Le système doit permettre à l'utilisateur de créer un groupe public

3.1.6.6. Le système doit permettre à l'utilisateur de créer un groupe privé

3.1.6.7. Le système doit permettre à l'utilisateur de joindre un groupe privé où il est invité

3.1.6.8. Le système doit permettre à l'utilisateur d'inviter des utilisateurs à son groupe privé ou public

3.1.6.9. Une fois dans un groupe, le système doit permettre de créer un canal de discussion du groupe

3.1.6.10. Une fois dans un groupe, le système doit permettre d'amener l'utilisateur dans une salle d'attente

3.1.6.11. Le système doit permettre de rendre disponible le bouton pour démarrer la partie une fois que le nombre minimum de joueurs est atteint.

3.1.6.12. Le système doit permettre un mode mêlé générale

3.1.6.13. Le système doit permettre un minimum de 2 joueurs pour jouer au mode mêlé général

3.1.6.14. Le système doit permettre d'attribuer des points aux joueurs qui trouvent la bonne réponse

3.1.6.15. Le système doit permettre de calculer un nombre de points relatif au temps écoulé et au nombre de joueurs ayant déjà eu la bonne réponse

3.1.6.16. Le système doit permettre d'attribuer des points au joueur qui dessine

3.1.6.17. Le système doit permettre d'attribuer un nombre de points pour le joueur qui dessine, dépendamment du nombre de joueurs ayant deviné avec succès

3.1.6.18. Le système doit permettre au joueur qui dessine d'avoir accès aux outils et à l'interface

de dessin

3.1.6.19. Le système doit permettre d'afficher aux autres joueurs les traits de dessin en temps réel

3.1.6.20. Le système doit permettre l'ajout d'un joueur virtuel à la partie

3.1.6.21. Le système doit permettre à un joueur virtuel de participer à un tour au même titre qu'un joueur humain.

3.1.6.22. Le système doit permettre aux joueurs virtuels de prendre uniquement le rôle de dessinateur.

3.1.7 Effets visuels et sonores

3.1.7.1. Le système doit pouvoir jouer un effet sonore lors d'une tentative de résolution

3.1.7.1.1. Le système doit jouer un son différent selon le résultat de la tentative de résolution

3.1.7.2. Le système doit pouvoir jouer un son de fin de partie selon la victoire ou la défaite de l'utilisateur

3.1.7.3. Le système doit pouvoir jouer un son selon les différentes actions dans l'espace de messagerie

3.1.7.3.1. Le système doit jouer un son particulier lors de la réception d'un message.

3.1.7.3.2. Le système doit jouer un son particulier lors de l'envoi d'un message

3.1.7.4. Le système doit pouvoir afficher une bulle de notification indiquant le nombre de nouveaux messages reçus par canal.

3.1.7.5. Le système doit pouvoir afficher une image indiquant le rôle du joueur lors d'une partie.

3.1.7.5.1. Le système doit pouvoir afficher un crayon en premier plan lorsque le rôle d'un joueur est assigné à "*dessinateur*"

3.1.7.5.2. Le système doit pouvoir afficher une paire de lunettes en premier plan lorsque le rôle d'un joueur est assigné à "*devineur*"

3.1.8 Tutoriel

3.1.8.1. Le système doit permettre de présenter un tutoriel non interactif.

3.1.8.2. Le système doit permettre au tutoriel de montrer une suite d'images explicatives avant le début du jeu.

3.1.8.2.1. Le système doit montrer une suite d'images décrivant les différents mécanismes du jeu.

3.1.8.2.2. Le système doit montrer une suite d'images décrivant la particularité du mode

de jeu choisi.

3.1.8.3. Le système doit permettre au tutoriel de s'exécuter pour chaque mode de jeu.

3.1.8.4. Le système doit activer automatiquement le tutoriel lorsqu'un utilisateur utilise l'application pour la première fois.

3.1.8.5. Le système doit permettre à l'utilisateur de revoir le tutoriel en tout temps grâce à un bouton dédié.

3.1.9 Type de parties

3.1.9.1. le système doit permettre à un créateur de partie de rendre la partie privée.

3.2 Exigences souhaitables du client lourd

3.2.1 Création d'un jeu

3.2.1.1. Le système doit permettre à l'utilisateur d'entrer un mot ou une expression afin de proposer des images.

3.2.1.2. Le système doit permettre à l'utilisateur de sélectionner une image parmi la série d'images affichées.

3.2.1.3. Le système doit permettre à l'utilisateur de consulter plus de résultats.

3.2.1.4. Le système doit permettre de pouvoir convertir l'image choisie par l'utilisateur en dessin.

3.2.1.5. Le système doit permettre à l'utilisateur de fournir un ensemble de paramètres pour l'engin de conversion

3.2.2 Tutoriel

3.2.2.1. Le système doit permettre de proposer un tutoriel interactif guidant l'utilisateur.

3.2.2.1.1 Le système doit permettre de proposer de l'information selon les cliques et la position de la souris de l'utilisateur

3.2.2.1.2 Le système doit permettre de bloquer le déroulement de la partie, afin de donner de l'information à l'utilisateur.

3.2.3 Modes de jeu

3.2.3.1. Le système doit permettre un mode supplémentaire intitulé sprint coopératif

3.2.3.2. Le système doit permettre à jusqu'à trois joueurs de se réunir dans le mode sprint coopératif.

3.2.3.3. Le système doit permettre à un joueur du sprint coopératif de deviner le mot dessiné par le joueur virtuel

3.2.3.4. Le système doit permettre un point collectif quand un des joueurs du groupe devine le mot

3.2.3.5. Le système doit permettre un nombre maximal de tentatives au groupe selon la difficulté du jeu

3.2.4 Personnalité des joueurs

3.2.4.1. Le système doit permettre à un joueur virtuel d'être capable de faire référence à des événements passés

3.2.4.1.1. Le système doit permettre à un joueur virtuel de faire référence à une ancienne partie jouée avec l'utilisateur.

3.2.4.1.2. Le système doit permettre à un joueur virtuel de faire référence aux autres joueurs rencontrés par l'utilisateur.

3.2.4.2. Le système doit permettre à un joueur virtuel d'être capable d'utiliser les statistiques d'un joueur pour formuler un commentaire personnalisé

3.2.4.3. Le système doit permettre à un joueur virtuel de posséder une personnalité: compétitif, joueur, timide...

3.2.4.3.1. Le système doit permettre à un joueur virtuel de s'exprimer différemment selon la personnalité définie.

3.2.4.3.2. Le système doit permettre à un joueur virtuel de s'exprimer dans le canal de jeu.

3.2.5 Motivation - compétition

3.2.5.1. Le système doit permettre d'afficher un écran de fin de jeux indiquant le résultat de la partie.

3.2.5.1.1. Le système doit permettre d'afficher le résultat des différents joueurs dans

l'ordre (leaderboard) à la fin d'une partie

3.2.5.1.2. Le système doit permettre d'afficher la durée de la partie à la fin d'une partie.

3.2.5.1.3. Le système doit permettre de partager le résultat à la fin d'une partie sur Facebook ou Twitter.

3.2.5.2. Le système doit permettre de pouvoir féliciter un joueur ayant fait un pointage élevé

3.2.5.3. Le système doit permettre de féliciter un utilisateur avec une boîte de dialogue suggérant de partager son score.

3.2.5.3.1. Dans le cas d'une partie multijoueur, le système doit féliciter le top 3.

3.2.5.3.2. Le système établit un seuil de pointage minimum définissant un score élevé

3.2.6 Profil

3.2.6.1. Le système doit permettre une progression du profil avec des niveaux

3.2.6.2. Le système doit permettre de donner à l'utilisateur un accès à de petits avantages de design.

3.2.6.2.1. Le système doit permettre des niveaux qui seraient basés sur le nombre de points totaux accumulé et le nombre de temps joué (niveau d'expérience de l'utilisateur).

3.2.6.2.2. Le système doit permettre d'afficher le niveau à côté du pseudonyme.

3.2.6.2. Le système doit permettre différents éléments de design selon le niveau de l'utilisateur

3.2.6.3. Le système doit permettre à l'utilisateur d'obtenir différentes couleurs pour le contour de l'avatar (bronze, argent, or)

3.2.6.3.1. Le système doit permettre à l'utilisateur de choisir parmi différentes polices de caractère pour l'affichage du pseudonyme.

3.2.6.3.2. Le système doit permettre à l'utilisateur de choisir le mode sombre.

3.2.7 Clavardage

3.2.7.1 Le système doit permettre à l'utilisateur d'insérer les émoticônes personnalisées de l'application dans l'espace de clavardage

3.2.7.1.1. Le système doit permettre l'insertion d'émoticône dans le chat à l'aide du format

suivant : “:nomÉmoticône:”

3.3 Exigences essentielles du client léger

3.3.1 Clavardage - Intégration

3.3.1.1. Le système doit permettre à l'utilisateur d'avoir accès à la zone de clavardage à tout moment dès que le client est connecté au serveur.

3.3.1.2. Le système doit permettre à l'utilisateur de clavarder en mode intégré.

3.3.1.3. Le système doit permettre à l'utilisateur de clavarder avec n'importe quelle personne en ligne.

3.3.1.4. Le système doit permettre à l'utilisateur de clavarder en cours de partie.

3.3.1.5. Le système doit permettre à l'utilisateur de clavarder hors partie.

3.3.1.6. Le système doit afficher les noms des utilisateurs avec leurs messages.

3.3.1.7. Le système doit afficher l'avatar à côté du nom dans l'interface de clavardage.

3.3.1.8. Le système doit permettre à l'utilisateur d'envoyer des messages textuels de moins de 144 caractères.

3.1.1.9. Le système doit présenter une liste des utilisateurs connectés.

3.1.1.10. Le système doit inclure une barre de recherche pour trouver un utilisateur à la zone de clavardage.

3.3.2 Clavardage - Canaux de discussion

3.3.2.1. Le système doit permettre à l'utilisateur de créer plusieurs canaux de discussion .

3.3.2.2. Le système doit permettre à l'utilisateur de supprimer plusieurs canaux de discussion .

3.3.2.3. Le système doit permettre à l'utilisateur de participer à plusieurs discussions en même temps.

3.3.2.4. Le système doit permettre à l'utilisateur de joindre un canal de discussion.

3.2.2.5. Le système doit inclure une barre de recherche pour trouver un canal de discussion à la

zone de clavardage.

3.2.2.6. Le système doit rendre disponible à l'utilisateur un canal principal avec la totalité des utilisateurs.

3.3.3 Profil utilisateur et historique

3.3.3.1. Le système doit permettre à l'utilisateur de créer son profil.

3.3.3.2. Le système doit obliger l'utilisateur à s'authentifier avec son profil pour se connecter.

3.3.3.3. Le système doit stocker les informations du profil.

3.3.3.4. Le profil contient une partie privée (prénom, nom, mot de passe).

3.3.3.5. Le profil contient une partie publique (pseudo, avatar).

3.3.3.6. Le système enregistre des statistiques dans le profil

3.3.3.6.1. Les statistiques d'utilisation du jeu contiennent le nombre de parties jouées, le pourcentage de victoires, le temps moyen d'une partie, et le temps total passé à jouer.

3.3.3.7. Le système enregistre un historique dans le profil

3.3.3.7.1. L'historique détaillé du profil contient un relevé de ses connexions et déconnexions avec dates et heures correspondantes dans un format lisible, un historique des parties jouées, autant seul qu'avec d'autres joueurs, et des résultats.

3.3.3.8. Le système doit évaluer la qualité du mot de passe pour l'accepter.

3.3.3.9. Le système doit s'assurer que le pseudo soit unique pour l'accepter.

3.3.3.10. Pour la sélection d'avatars, le système doit offrir d'importer une photo ou de prendre une photo.

3.3.3.11. Un seul client à la fois peut être connecté à un compte.

3.3.3.12. Le système doit rendre le pseudo et l'avatar visible aux autres utilisateurs.

3.3.3.13. La partie privée du profil n'est pas visible par les autres utilisateurs.

3.3.4 Modes de jeu

- 3.3.4.1. Après la connexion, le système doit diriger l'utilisateur vers un lobby.
- 3.3.4.2. Le système doit permettre à l'utilisateur de choisir un mode de jeu à partir du lobby.
- 3.3.4.3. Le système doit présenter les groupes publics existants d'un mode de jeu.
- 3.3.4.4. Le système doit permettre à l'utilisateur de rejoindre un groupe public existant.
- 3.3.4.5. Le système doit permettre à l'utilisateur de créer un groupe public.
- 3.3.4.6. Le système doit permettre à l'utilisateur de créer un groupe privé.
- 3.3.4.7. Le système doit permettre à l'utilisateur de rejoindre un groupe privé en entrant un mot de passe.
- 3.3.4.8. Une fois dans un groupe, le système doit créer un canal de discussion incluant les membres du groupe.
- 3.3.4.9. Une fois dans un groupe, le système doit amener les utilisateurs dans une salle d'attente.
- 3.3.4.10. Le système doit rendre disponible le bouton pour démarrer la partie une fois que le nombre minimum de joueurs est atteint.
- 3.3.4.11. Le système doit offrir le mode mêlée générale.
 - 3.3.4.11.1. Le mode mêlée générale requiert un minimum de 2 joueurs.
 - 3.3.4.11.2. Le système doit attribuer des points aux joueurs qui trouvent la bonne réponse.
 - 3.3.4.11.3. Le système doit calculer un nombre de points relatif au temps écoulé et au nombre de joueurs ayant déjà eu la bonne réponse.
 - 3.3.4.11.4. Le système doit attribuer des points au joueur qui dessine.
 - 3.3.4.11.5. Le nombre de points pour le joueur qui dessine doit être dépendant du nombre de joueurs ayant deviné avec succès.
 - 3.3.4.11.6. Le système doit donner au joueur qui dessine l'accès aux outils et à l'interface de dessin.

3.3.4.11.7. Le système doit afficher aux autres joueurs les traits en temps réel .

3.3.4.11.8. Le système doit permettre l'ajout d'un ou de plusieurs joueurs virtuels à la partie.

3.3.4.11.9. Les joueurs virtuels participent aux tours au même titre que les joueurs humains.

3.3.4.11.10. Les joueurs virtuels ne peuvent pas deviner ce que représentent les dessins. Ils prennent donc uniquement le rôle de dessinateur.

3.3.5 Effets visuels et sonores

3.3.5.1. Le système doit contenir au minimum trois groupes distincts d'effets sonores et visuels.

3.3.6 Tutoriel

3.3.6.1. Le système doit présenter un tutoriel interactif nécessitant des actions de l'utilisateur.

3.3.6.1.1. Le système doit proposer de l'information selon les clicks et la position de la souris de l'utilisateur

3.3.6.1.2. Le système doit pouvoir bloquer le processus de la partie afin de donner de l'information à l'utilisateur.

3.3.6.2. Le système doit permettre à l'utilisateur de compléter le tutoriel après une certaine progression

3.3.6.2.1. Le système doit permettre une progression dans le tutoriel selon l'accomplissement des objectifs donnés à l'utilisateur

3.3.6.3. Le système doit permettre à l'utilisateur de comprendre les différentes mécaniques du jeu à travers les différents objectifs données dans le tutoriel.

3.3.6.4. Le système doit activer le tutoriel automatiquement à la première connexion d'un utilisateur.

3.3.6.5. Le tutoriel doit être disponible en tout temps par la suite.

3.3.7 Type de parties

3.3.7.1. Le système doit permettre aux créateurs de parties de rendre la partie privée.

3.3.7.2. Le système doit permettre aux créateurs de parties d'inviter d'autres utilisateurs à leur partie.

3.4 Exigences souhaitables du client léger

3.4.1 Modes de jeu

3.4.1.1. Le système doit offrir le mode de sprint coopératif en plus des autres modes de jeu. Le mode sprint coopératif est comme le mode de sprint solo, mais à plusieurs joueurs humains qui collaborent afin de deviner les mots ou expressions.

3.4.1.2. Le système doit permettre jusqu'à 4 joueurs de rejoindre la même partie pour deviner un maximum de mots ou expressions dessinés par un joueur virtuel.

3.4.1.3. Le système accorde un point au joueur qui entre la bonne réponse en premier.

3.4.1.4. En fonction du niveau de difficulté choisi, les joueurs ont un nombre maximum collectif d'essais.

3.4.2 Clavardage

3.4.2.1: Une notification sera envoyée au client lors de la réception de nouveaux messages (par indication visuelle et effet sonore).

3.4.2.2 Le système permet à l'utilisateur d'insérer les émoticônes personnalisées de l'application dans l'espace de clavardage à l'aide du format suivant : `":nomÉmoticones:"`.

4. Exigences non fonctionnelles

La section sert à démontrer les exigences qui ne sont pas fonctionnelles, par exemple l'utilisabilité, la performance, etc.

4.1. Utilisabilité

4.1.1. L'application doit inclure un tutoriel descriptif, qui explique en détail comment utiliser les fonctionnalités que les utilisateurs utilisent le plus souvent, pour permettre à ceux-ci de comprendre toutes les fonctionnalités.

4.1.2. L'interface de notre application doit être intuitive et simple, avec le moins de boutons possibles et placés à des positions et avec des formes et couleurs habituelles (bouton "oui" à droite et de couleur vert, bouton "non" à gauche et de couleur rouge, section pour entrer du texte

en bas, etc.).

4.1.3. Un utilisateur peut créer son profil en moins d'une minute, sans trop d'opérations obligatoires de sa part et le moins possible d'informations obligatoires.

4.1.4. Un utilisateur ne devrait pas avoir besoin de passer plus d'une minute dans le tutoriel afin de devenir familier avec les différentes fonctionnalités de l'application.

4.1.5. Tous les éléments de l'application doivent avoir une continuité, les boutons doivent toujours avoir la même forme et l'interface doit garder le même thème de couleur dans toutes ses sections.

4.1.6. Le temps d'exécution pour l'envoi d'un message devront être presque instantané (100 ms maximum).

4.1.7. Le temps d'apparition du dessin quand l'utilisateur fait un trait doit être presque instantané (10 ms maximum).

4.1.8. L'interface et les boutons doivent être réactifs; une bulle d'information doit être affichée quand le curseur survole certains éléments et les boutons doivent changer de couleur lorsque le curseur les survole, cela afin de guider l'utilisateur et de l'aider dans sa compréhension du fonctionnement du logiciel.

4.2. Fiabilité

4.2.1. La base de données doit être active et prête pour recevoir des requêtes pour un minimum de 99% du temps. Une assurance de 99.5% est assurée par le logiciel Heroku que nous utilisons pour l'hébergement de notre base de données.

4.2.2. Le système et notre serveur devront être hors d'usage au maximum 1% du temps en minimisant le temps moyen de réparation après une panne.

4.3. Performance

4.3.1. Le temps de réponse doit être presque instantané (100 ms maximum) pour la transmission d'un message envoyé par l'utilisateur dans le chat.

4.3.2. Le temps de réponse pour l'affichage d'un trait de dessin lorsque l'utilisateur dessine doit être presque instantané (10 ms maximum).

4.3.3. L'application doit être au maximum 1Go en termes de mémoire utilisée afin de ne pas en

prendre trop sur le système sur lequel l'application est installée.

4.3.4. L'application doit offrir un minimum de 60 rafraîchissements d'image par seconde pour le client lourd et le client léger. Pour le client léger, le matériel visé est la tablette Galaxy Tab A, et pour le client léger, un ordinateur comportant au minimum 1 Go de mémoire vive et un processeur à 2 coeurs.

4.3.5. Un nombre maximal de 8 utilisateurs simultanés doivent être supportés par le serveur pour une partie.

4.4. Maintenabilité

4.4.1. Un administrateur doit avoir accès à la base de données pour gérer les informations personnelles de chacun des utilisateurs.

4.4.2. Les constantes doivent être écrites en majuscule (ex: CONSTANTES_EXEMPLE).

4.4.3. Les variables qui sont constantes doivent être déclarées constante (ex: const variable)

4.4.4. Le code doit être développé en anglais.

4.4.5. Le code ne doit pas avoir de répétition ni de redondance.

4.4.6. Chaque fonction doit avoir uniquement une fonctionnalité.

4.4.7. "Tabs" doivent être équivalent à 4 "Espaces".

4.5. Contraintes de conception

4.5.1. Le client lourd doit être programmé dans le cadre de WBF.

4.5.2. Le client lourd doit être programmé dans le langage de programmation C#.

4.5.3. Le client lourd doit être développé sur Visual Studio 2019.

4.5.4. Le client léger doit être programmé avec Android Studio .

4.5.5. Le client léger doit être programmé dans le langage de programmation Kotlin.

4.5.6. Le serveur doit être construit sur Node.js.

4.5.7. Le serveur doit être programmé dans le langage de programmation Typescript.

4.5.8. Le serveur doit être développé sur VScode.

4.5.9. La base de données doit être Postgresql et hébergé sur Heroku.

4.6. Sécurité

4.6.1. Les mots de passe doivent être cryptés et sauvegardés dans la base de données.

4.6.2. Les connexions à la base de données doivent être restreintes à seulement les développeurs pour éviter le vol d'information.

4.6.3. Le serveur doit vérifier la forme (les informations envoyées doivent suivre un format défini afin d'éviter, par exemple, l'injection de code lors des requêtes) des requêtes afin d'éviter les failles de sécurité.

4.6.4. L'avatar doit être seulement changeable que par l'utilisateur et non par d'autres personnes.

4.7. Exigences de la documentation utilisateur en ligne et du système d'assistance

4.7.1. Le tutoriel doit être activable à nouveau même si l'utilisateur l'a déjà fait.

4.8. Normes applicables

4.8.1. La branche de "master" doit être bloquée pour ajouter des modifications. Il est obligatoire de travailler sur des branches séparées.

4.8.2. Un "pull request" doit être créé pour avoir la possibilité d'ajouter des codes dans la branche master.

4.8.3. Afin que le "pull request" soit accepté, le "CI" doit passer les tests ainsi que deux "code review" effectués par les développeurs de l'équipe.

4.8.4. Les développeurs doivent respecter la convention des noms de branches et avoir des messages de "commits" descriptifs et courts.

4.8.5. Les développeurs doivent mettre à jour la feuille de temps et les demandes sur le redmine, chaque jour.