205

Fais-moi un dessin
Spécifications des requis du système (SRS)

Version 2.1

Historique des révisions

Date	Version	Description	Auteur
2021-01-25	1.0	Première Ébauche	Toute l'équipe
2021-01-27	1.1	Ajout des fonctionnalités de la liste d'exigences complète	Toute l'équipe
2021-02-01	1.2	Relecture du SRS et modifications légères	Toute l'équipe
2021-02-03	1.3	Révision finale	Toute l'équipe
2021-02-17	2.0	Appliquer les corrections recommandées	Augustin Bouchard
2021-02-19	2.1	Révision finale pour la réponse à l'appel d'offre	Augustin Bouchard

Table des matières

Spécifications des requis du système (SRS)

1. Introduction

1.1. But

Le SRS décrit le comportement externe du logiciel de jeu interactif *Fais-moi un dessin* selon les spécifications. 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. Les différends requis sont également classés selon leur appartenance aux catégories fonctionnelles ou non fonctionnelles.

1.2. Définitions, acronymes et abréviations

Android : Système d'exploitation mobile

Android studio : Outil de développement d'applications Android

BD : Base de données

Client lourd : Application exécutable Electron compatible spécifiquement avec un ordinateur personnel Windows (PC)

Client léger : Code source étant capable d'être exécuté sur un émulateur répliquant une tablette Samsung Galaxy Tab A 2019.

Electron : Framework basé sur l'engine V8 javascript de Google permettant la conversion d'une application web en exécutable Windows

Framework : Ensemble de bibliothèques, d'outils et de convention permettant le développement de l'application

Joueur réel : Utilisateur humain qui utilise l'application avec son compte

Joueur virtuel : Utilisateur émulé par le serveur qui sera capable de dessiner et d'envoyer des indices aux joueurs réels.

Lobby : Groupe de joueurs associés à une partie

Paire mot-image: Dessin émulé par le joueur virtuel contenant toutes les informations telles que: mot à deviner, indices et le mode d'affichage du dessin

TypeScript : Langage de programmation orienté-objet visant à simplifier l'utilisation de javascript dans une application web

Windows : Système d'exploitation d'un ordinateur

1.3. Vue d'ensemble du document

La première section d'introduction donne les informations générales du projet comme le but, les définitions ainsi que la description globale. Par la suite, les exigences du logiciel sont séparées en deux catégories : fonctionnelle et non fonctionnelle.

2. Description globale

Le logiciel à développer est un jeu *Fais-moi un dessin* disponible sur deux plateformes. Le client-lourd sera basé sur le logiciel développé durant le projet 2 *Polydessin* qui permettait à un utilisateur de faire des dessins vectoriels et de les modifier. Il s'agira donc d'une évolution du projet 2. Un joueur doit dessiner un dessin selon un mot qui lui est donné et les autres doivent par la suite deviner le mot selon le dessin de joueur. Les deux plateformes sont Android et Windows. Les joueurs pourront affronter des personnes en ligne qui jouent sur de différentes plateformes qu'eux

et ils pourront aussi discuter avec ceux-ci via un chat.

2.1. Caractéristiques des usagers

Les utilisateurs visés par cette application sont des ingénieurs logiciels. On peut donc supposer qu'ils sont au moins âgés de 20 ans. On peut aussi assumer qu'ils sont très habiles ou du moins qu'ils connaissent la technologie que nous utiliserons pour développer notre application.

2.2. Interfaces

2.2.1. Interfaces usagers

L'interface utilisateur contient une partie pour les menus, notamment une section pour se connecter et une section pour rejoindre une partie, puis une interface de jeu pour la partie même. L'interface pour le client lourd doit être cohérente et similaire à celle du client léger afin de permettre aux utilisateurs de facilement s'y retrouver d'une interface à l'autre. L'interface du client lourd est développée en HTML et CSS par le *framework* Angular et celle du client léger est développée en Kotlin. Il y a finalement l'interface du serveur, ce dernier sera développé avec Azure donc son interface web sera utilisée par les développeurs lors de de différentes tâches de développement.

2.2.2. Interfaces matérielles

Les interfaces matérielles sont l'écran, le système de son, le clavier et la souris en ce qui concerne le client lourd. Les utilisateurs du client léger quant à eux utilisent l'interface tactile multi-Touch.

2.2.3. Interfaces logicielles

Le client lourd doit être conçu sur Electron en utilisant le framework Angular et le langage TypeScript. Le client léger doit être codé en Kotlin. Le serveur doit être conçu en utilisant le framework Node.js. Le client lourd doit pouvoir fonctionner sous le système d'exploitation Windows 10 version 20H2 et le client léger doit pouvoir fonctionner sous Android Pie 9.0. Le serveur doit pouvoir être hébergé sur le service d'hébergement cloud Azure de Microsoft. La base de données SQL sera aussi déployée sur le service d'hébergement cloud Azure de Microsoft. Le serveur utilisera le framework Express ainsi que la libraire socket.io pour permettre une communication fluide utile pour le clavardage. Au niveau client sur Electron, on utilise également ngx-color-picker pour permettre le choix de la couleur du crayon pour le dessin.

Les principales librairies du client léger sur Android seront les suivantes: hilt-android pour l'injection de dépendances, androidx lifecycle pour les observables et socket io pour les échanges avec le serveur.

2.2.4. Interfaces de communication

Les joueurs doivent se connecter à Internet pour pouvoir jouer au jeu. Donc, dans le cas d'un utilisateur sur tablette Android, il doit être en mesure de pouvoir se connecter à Internet avec le Wi-Fi. Pour l'utilisateur du client lourd sur Windows, il doit être en mesure de pouvoir se connecter à Internet via un câble Ethernet ou bien Wi-Fi. Les communications se feront avec le protocole TCP.

2.3. Contraintes générales

Il doit être en mesure d'accepter 4 personnes dans le même lobby sans délai visible lors des dessins, et ce pour dix lobbys simultanément. La tablette Android possède seulement 2 Gb de RAM, il faudra s'assurer que le client-léger s'exécute fluidement sans prendre la totalité de la mémoire.

2.4. Hypothèses et dépendances

On suppose que le client lourd utilise le système d'exploitation Windows 10 et que le client léger est une tablette de type Galaxy Tab A 2019. Nous supposerons que les utilisateurs possèdent une connexion sur le réseau Internet avec un débit de téléchargement suffisant (10 Mb/s). Le serveur étant déployé sur la plateforme Azure de Microsoft, on supposera qu'il sera accessible 99.9% du temps comme mentionné par Microsoft.

3. Exigences fonctionnelles

Exigences essentielles

3.1. Clavardage - Intégration (Client lourd)

3.1.1 Le système doit permettre à l'utilisateur d'accéder au clavardage à tout moment

3.1.1.1 Le système doit permettre à l'utilisateur d'accéder au clavardage en mode fenêtré

3.1.1.2 Le système doit permettre à l'utilisateur d'accéder au clavardage en mode intégré

3.1.2 Le système doit permettre à l'utilisateur de passer du mode de clavardage intégré au mode fenêtré

3.1.2.1 Le système doit retourner au mode de clavardage intégré lorsque la fenêtre du mode fenêtré est fermée

3.1.2.2 Le système doit conserver la connexion de l'application principale dans la fenêtre de clavardage

3.1.2.2.1 Le système doit reconnaître l'utilisateur et lui permettre d'envoyer des messages à partir de son compte, peu importe le mode (fenêtré ou intégré)

3.1.2.2.2 Le système doit connecter automatiquement l'utilisateur à la fenêtre de clavardage

3.1.2.3 Le système doit permettre le clavardage en dehors d'une partie

3.2. Clavardage - Intégration (Client léger)

3.2.1 Se référer au point 3.1.1

3.2.1.1 L'interface de clavardage doit être intégrée dans l'application

3.2.3 Le système doit notifier l'utilisateur lors de la réception d'un nouveau message par un effet sonore à chaque nouveau message reçu

3.3. Clavardage - Canaux de discussion (Client lourd et léger)

3.3.1 Le système doit permettre à l'utilisateur de créer des canaux de discussion

3.3.2 Le système doit permettre à l'utilisateur de rejoindre des canaux de discussion

3.3.3 Le système doit permettre à l'utilisateur de quitter des canaux de discussion

3.3.4 Le système doit permettre à l'utilisateur de supprimer des canaux de discussion

3.3.5 Le système doit permettre à l'utilisateur de participer à plusieurs canaux de discussion simultanément

3.3.6 Le système doit permettre à l'utilisateur d'afficher l'historique du clavardage de chaque canal de discussion ouvert

3.3.6.1 Par défaut, le système ne doit afficher que les messages envoyés depuis la connexion de l'utilisateur

3.3.7 Le système doit censurer les mots vulgaires en les cachant partiellement par des astérisques

3.3.8 Le système doit vider la boîte de texte après l'envoi d'un message

3.3.9 Le système doit garder l'accent sur la boîte de texte après l'envoi d'un message

3.3.10 Le système doit empêcher l'envoi de message sans caractère visible

3.3.11 Le système doit accompagner le message du nom de l'utilisateur qui a envoyé le message

3.3.12 Le système doit accompagner le message du temps d'envoi selon le format HH:MM:SS

3.3.13 Le système de clavardage doit supporter les caractères spéciaux UTF-8

3.4 Profil utilisateur et historique (Client lourd et léger)

3.4.1 Le système doit permettre à l'utilisateur d'avoir un profil qui lui permet d'être authentifié par le serveur avec un pseudonyme unique et un mot de passe

3.4.1.1 Le profil doit contenir une partie privée.

3.4.1.1.1 Le profil privé contient le prénom

3.4.1.1.2 Le profil privé contient le nom

3.4.1.1.3 Le profil privé contient le courriel

3.4.1.2 Le profil doit contenir une partie publique

3.4.1.2.1 Le profil public contient le pseudonyme

3.4.1.2.2 Le profil public contient l'avatar

3.4.1.3 Le profil doit contenir des statistiques sur l'utilisation du jeu.

3.4.1.3.1 Le profil doit contenir la statistique du nombre de parties jouées

3.4.1.3.2 Le profil doit contenir la statistique du pourcentage de victoire

3.4.1.3.3 Le profil doit contenir la statistique du temps moyen d'une partie

3.4.1.3.4 Le profil doit contenir la statistique du temps total passé en jeu

3.4.1.4 Le profil doit contenir un historique détaillé

3.4.1.4.1 Le profil doit contenir les dates et heures de connexion de l'utilisateur

3.4.1.4.2 Le profil doit contenir les dates et heures de déconnexion de l'utilisateur

3.4.1.4.3 Le profil doit contenir l'historique des parties jouées

3.4.1.4.3.1 L'historique d'une partie jouée doit contenir la date de la partie.

3.4.1.4.3.2 L'historique d'une partie jouée doit contenir l'heure de la partie.

3.4.1.4.3.3 L'historique d'une partie jouée doit contenir le pseudonyme des joueurs ayant joué dans la partie

3.4.1.4.3.4 L'historique d'une partie jouée doit contenir le résultat de la partie

3.4.2 Le système doit envoyer un message d'erreur lorsqu'un pseudonyme n'existe pas dans la base de données lors de l'authentification

3.4.3 Le système doit envoyer un message d'erreur lorsque l'utilisateur entre un mot de passe incorrecte associé au pseudonyme au moment de la connexion

3.5 Modes de jeu (Client lourd et léger)

3.5.1 Le système doit offrir une surface de dessin visible pour tous les membres de la partie

3.5.1.1 Le système doit afficher les joueurs de la partie

3.5.1.1.1 Le système doit afficher le pointage du joueur à côté de son nom

3.5.1.2 Le système doit permet à un seul joueur de modifier le dessin à la fois

3.5.1.3 Le système doit mettre à jour le pointage après chaque tour d'une partie

3.5.2 Le système doit offrir le mode de jeu classique qui doit permettre de jouer à au moins 2 contre 2.

3.5.2.1 Une équipe doit être composée d'au plus 1 joueur virtuel s'il y a moins de 4 joueurs réels.

3.5.2.1.1 Un joueur virtuel doit avoir le rôle de dessiner pour l'entièreté de la partie.

3.5.2.1.2 Un joueur réel en équipe avec un joueur virtuel doit être devineur pour l'entièreté de la partie

3.5.2.2 Dans chaque équipe, il doit y avoir un joueur qui dessine le mot

3.5.2.3 Dans chaque équipe, il doit y avoir un joueur qui devine le mot

3.5.2.4 Si le joueur devineur se trompe, l'équipe n'obtient pas de points

3.5.2.4.1 L'autre équipe doit avoir un droit de réplique

3.5.2.4.1.1 Aucun point doit être accordé si l'autre équipe se trompe aussi

3.5.2.4.1.2 Le tour est terminé si le devineur de chaque équipe se trompe

3.5.24.1.3 Le tour est terminé dès qu'un devineur devine le mot

3.5.2.5 Les rôles (dessinateur et devineur) doivent être inversés après chaque tour.

3.5.3 Le système de jeu doit offrir trois modes de difficulté pour la partie

3.5.3.1 L'utilisateur peut choisir le niveau de difficulté facile pour sa paire mot-image

3.5.3.2 L'utilisateur peut choisir le niveau de difficulté intermédiaire pour sa paire mot-image

3.5.3.3 L'utilisateur peut choisir le niveau de difficulté difficile pour sa paire mot-image

3.5.3.4 La vitesse de dessin des joueurs virtuels dépend de ce niveau de difficulté

3.5.3.5 Le temps alloué aux joueurs dépend du niveau de difficulté

3.5.3.5.1 Le temps alloué aux joueurs au niveau facile est de 60 secondes

3.5.3.5.2 Le temps alloué aux joueurs au niveau intermédiaire est de 40 secondes

3.5.3.5.3 Le temps alloué aux joueurs au niveau difficile est de 20 secondes

3.5.3.6 Le nombre de tentatives allouées aux joueurs pour deviner le mot dépend de ce niveau de difficulté

3.5.3.6.1 Au niveau facile, 5 tentatives doivent être attribuées aux joueurs

3.5.3.6.2 Au niveau moyen, 3 tentatives doivent être attribuées aux joueurs

3.5.3.6.3 Au niveau difficile, 1 tentative doit être attribuée aux joueurs

3.6 Création d'une paire mot-image (Client lourd)

3.6.1 Le système doit permettre aux utilisateurs de créer un jeu manuellement en fournissant les données nécessaires à la création du jeu

3.6.1.1 L'utilisateur doit fournir le mot recherché

3.6.1.2 L'utilisateur doit fournir au moins un indice

3.6.1.3 L'utilisateur doit dessiner l'image dans une zone de dessin

3.6.1.3.1 La zone de dessin doit contenir les mêmes fonctionnalités que lors d'une partie

3.6.1.3.2 Le système doit gérer le dessin sous le format SVG

3.6.1.4 L'utilisateur doit pouvoir choisir un niveau de difficulté associé à sa paire mot-image

3.6.2 Le système doit permettre à l'utilisateur de choisir le mode de dessin lors de la création d'un jeu manuellement

3.6.2.1 L'utilisateur doit pouvoir choisir le mode conventionnel où chaque trait est dessiné dans le même ordre que lors de la création du dessin par l'utilisateur

3.6.2.1.1 Par défaut, le système doit sélectionner le mode de dessin conventionnel

3.6.2.2 L'utilisateur peut choisir le mode aléatoire où les traits apparaissent dans un ordre aléatoire

3.6.2.2.1 L'ordre d'apparition des traits doit être différent à chaque partie en mode aléatoire

3.6.2.3 L'utilisateur peut choisir le mode panoramique où chaque trait apparaît dans l'ordre de leur position selon une direction choisie

3.6.2.3.1 L'utilisateur doit pouvoir choisir de faire apparaître les traits de droite à gauche

3.6.2.3.2 L'utilisateur doit pouvoir choisir de faire apparaître les traits de gauche à droite

3.6.2.3.3 L'utilisateur doit pouvoir choisir de faire apparaître les traits de haut à bas

3.6.2.3.4 L'utilisateur doit pouvoir choisir de faire apparaître les traits de bas à haut

3.6.2.4 L'utilisateur doit pouvoir choisir le mode centré où les traits apparaissent dans l'ordre de leur distance du centre de l'image

3.6.2.4.1 L'utilisateur doit pouvoir choisir de faire apparaître les traits de l'intérieur vers l'extérieur

3.6.2.4.2 L'utilisateur doit pouvoir choisir de faire apparaître les traits de l'extérieur vers l'intérieur

3.6.3 Le système doit permettre à l'utilisateurs d'importer une image de son ordinateur plutôt que de dessiner l'image lors de la création d'un jeu

3.6.3.1 L'image importée doit être de format matriciel (type bitmap)

3.6.3.2 L'image importée doit être convertie en format SVG

3.6.3.2.1 L'utilisateur peut configurer les différentes variables de l'engin de conversion

3.6.3.3 Seul les modes présentés en 3.6.2.3 doivent être disponibles pour une image importée

3.6.4 Le système doit permettre aux utilisateur de prévisualiser le dessin

3.7 Surface de dessin (Client lourd et léger)

3.7.1 L'utilisateur doit pouvoir dessiner sur une surface de dessin à l'aide d'une option crayon

3.7.1.1 L'utilisateur doit pouvoir changer l'épaisseur du crayon

3.7.1.2 L'utilisateur doit pouvoir changer la couleur du crayon

3.7.1.3 L'utilisateur doit pouvoir changer l'opacité du crayon

3.7.1.4 L'utilisateur doit pouvoir sélectionner une des 10 dernières couleurs

3.7.2 L'utilisateur doit pouvoir effacer les traits de crayon à l'aide d'une option efface

3.7.2.1 L'utilisateur doit pouvoir changer l'épaisseur de l'efface

3.7.2.2 Le système doit permettre d'effacer en brosse

3.7.2.3 Lorsque l'efface passe au-dessus d'un coup de crayon, le coup de crayon devient temporairement rouge

3.7.2.3.1 Si le coup de crayon est déjà rouge, le coup de crayon doit avoir une teinte de rouge plus foncée.

3.7.2.4 Lorsqu'on efface avec un clic, on retire seulement le trait le plus au-dessus (chevauchement)

3.7.3 Les traits de crayon dessinés par l'utilisateur doivent pouvoir être retirés et remis un à un à l'aide d'une option annuler/refaire

3.7.3.1 L'utilisateur doit pouvoir annuler les traits dans l'ordre du plus récent

3.7.3.2 L'utilisateur doit pouvoir refaire les traits annulés

3.7.3.3 Refaire est rendu indisponible dès la création d'un nouveau trait

3.7.3.4 Un trait de crayon commence au moment ou le crayon touche la surface de dessin

3.7.3.5 Un trait de crayon termine au moment ou le crayon quitte la surface de dessin

3.7.4 L'utilisateur doit pouvoir afficher une grille sur sa surface de dessin

3.7.4.1 La grille doit être présente pour l'utilisateur seulement

3.7.4.2 L'utilisateur doit pouvoir changer la grosseur du quadrillage de la grille

3.7.5 L'utilisateur doit pouvoir choisir la couleur désirée à l'aide d'un outil de sélection de couleurs

3.7.5.1 La couleur choisie doit être gardée en mémoire dans une liste de présélections

3.7.5.2 L'utilisateur doit pouvoir choisir l'opacité de la couleur

3.7.6 La surface de dessin doit être la même pour les utilisateurs du client léger et du client lourd de manière à ce que tous les dessins soient exactement les mêmes pour les utilisateurs de n'importe quelle plateforme

3.8 Personnalité des joueurs virtuels (Client lourd et léger)

3.8.1 Le système doit permettre au joueur virtuel d'avoir des communications simples

3.8.1.1 Le joueur virtuel doit fournir des indices lorsque demandé

3.8.1.2 Le joueur virtuel doit encourager son coéquipier

3.8.1.2 Le joueur virtuel doit féliciter son coéquipier

3.8.1.3 Le joueur virtuel doit publier un message au début de la partie et à la fin de chaque tour

3.8.1.3 Le joueur virtuel doit publier un message à la fin de chaque tour

3.8.2 Le système doit permettre aux joueurs virtuels d'avoir des personnalités variées

3.8.2.1 Les joueurs virtuels doivent avoir différents caractères uniques

3.8.2.2 Un joueur virtuel doit conserver son caractère durant toute la partie

3.8.2.3 Les personnalités peuvent être prétentieux, colérique, comique, etc.

3.8.3 Le système doit permettre au joueur virtuel d'avoir des conversations dynamiques

3.8.3.1 Le joueur virtuel doit pouvoir référer des parties jouées

3.8.3.2 Le joueur virtuel doit pouvoir référer des joueurs rencontrés

3.8.3.3 Le joueur virtuel doit pouvoir utiliser des statistiques des joueurs pour formuler des commentaires personnalisé

3.9 Tutoriel (Client lourd et léger)

3.9.1 Le système doit afficher le tutoriel automatiquement pour la première connexion à l'application

3.9.2 Le système doit permettre de refaire le tutoriel à partir du menu principal

3.9.3 Le système doit présenter un tutoriel composé d'une série d'images expliquant le fonctionnement de l'application

3.9.3.1 Le système doit permettre de passer à la prochaine instruction/image.

3.10 Effets visuels et sonores (Client lourd et léger)

3.10.1 Le système doit présenter un rétroaction sonore après chaque tentative de résolution

3.10.2 Le système doit présenter un effet de particules lors d'une victoire.

3.10.3 Le système doit présenter un décompte (3...2...1) avant chaque manche d'une partie

3.11 Déroulement d'une partie (Client lourd et léger)

3.11.1 Le système doit indiquer lorsqu'une réponse donnée par un utilisateur est proche du mot réel

3.11.1.1 Un mot doit être considéré comme « proche » lorsqu'il a un caractère différent du mot réel

3.11.1.2 Le système doit ajouter un message seulement visible par l'utilisateur en question dans la zone de réponse

3.11.2 Le joueur doit pouvoir entrer sa tentative de mot à deviner dans le canal de discussion de la partie

3.11.2.1 Le canal de discussion de la partie est réservé aux joueurs de la partie.

3.11.2.2 Le message doit être envoyé à tous les autres joueurs de la partie seulement si le mot est mauvais

3.12 Page d'accueil (Client lourd et léger)

3.12.1 L'utilisateur doit pouvoir quitter le logiciel

3.12.2 La page d'accueil doit inviter l'utilisateur à entrer son pseudonyme et son mot de passe pour se connecter à son profil

3.12.3 L'utilisateur doit pouvoir accéder à une nouvelle page pour créer un profil

3.12.3.1 L'utilisateur doit pouvoir choisir un pseudonyme

3.12.3.2 L'utilisateur doit pouvoir choisir un email

3.12.3.3 L'utilisateur doit pouvoir choisir un mot de passe

3.12.4 L'utilisateur doit pouvoir accéder au Menu principal des parties

3.13 Page menu principal des parties (Client lourd et léger)

3.13.1 L'utilisateur doit pouvoir sélectionner le mode de jeu auquel il souhaite accéder
3.13.2 L'utilisateur doit pouvoir sélectionner la difficulté souhaitée de la partie
3.13.3 L'utilisateur doit pouvoir créer un lobby
3.13.4 L'utilisateur doit pouvoir rejoindre un lobby
- 3.13.4.1 L'utilisateur doit pouvoir consulter la liste des joueurs virtuels dans le lobby
- 3.13.4.2 L'utilisateur doit pouvoir consulter la liste des joueurs réels dans le lobby
- 3.13.4.3 L'utilisateur doit pouvoir avoir l'option d'ajouter un joueur virtuel
- 3.13.4.4 L'utilisateur doit pouvoir avoir l'option d'enlever un joueur virtuel
- 3.13.4.5 L'utilisateur doit automatiquement être ajouté au canal de discussion de la partie lorsqu'il rejoint le lobby
- 3.13.4.6 Tous les utilisateurs ont le contrôle du lobby
- 3.13.4.7 L'utilisateur doit pouvoir commencer la partie avec un bouton lorsque le nombre minimum de joueurs est atteint.

Exigences souhaitables

3.14 Interactions de multijoueurs en partie (Client lourd et léger)

3.14.1 Le système doit permettre aux utilisateurs d'attribuer un pouce à un autre utilisateur pour un dessin
- 3.14.1.1 Le système doit permettre aux utilisateurs d'attribuer un pouce vert
- 3.14.1.2 Le système doit permettre aux utilisateurs d'attribuer un pouce rouge
- 3.14.1.3 Les pouces doivent être collectés dans les statistiques de chaque joueur
- 3.14.1.4 Les pouces vert et rouge doivent apparaître à côté du nom du joueur qui dessine
- 3.14.1.5 Les pouces vert et rouge doivent disparaitre après 3 secondes

3.15. Clavardage - Intégration (Client léger)

3.15.1 Le système doit notifier l'utilisateur lors de la réception d'un nouveau message par un indicateur visuel lorsque le clavardage est fermé
- 3.15.1.1 L'indicateur visuel doit disparaître lorsque tous les canaux contenant de nouveaux messages ont étés ouverts

3.16 Profil utilisateur et historique (Client lourd et léger)

3.16.1 Le système doit permettre à l'utilisateur de se connecter avec un compte Google
- 3.16.1.1 L'utilisateur doit choisir son nom d'utilisateur lors de sa première connexion

3.16.2 Le système doit permettre l'utilisateur d'utiliser un « Two-Factor Authentication »
- 3.16.2.1 Lors de chaque connexion d'un utilisateur, le système doit envoyer un code par courriel à l'utilisateur
- 3.16.2.2 Le système doit permettre à l'utilisateur d'entrer le code reçu par courriel pour se connecter
- 3.16.2.3 L'utilisateur peut décider s'il utilise le « Two-Factor Authentification » dans les paramètres de son compte
- 3.16.2.4 Le « Two-Factor Authentication » doit être désactivé par défaut

3.17 Modes de jeu (Client lourd et léger)

3.17.1 Le système de jeu doit offrir un mode de jeu sprint solo ou il faut trouver le plus de mots ou expressions possibles
- 3.17.1.1 La partie doit être limitée dans le temps selon la difficulté
- 3.17.1.2 Le dessin à deviner doit être créé par un joueur virtuel
- 3.17.1.3 Le joueur a un nombre d'essais maximum pour deviner le mot ou l'expression
  - 3.17.1.3.1 Le nombre d'essais est déterminé en fonction de la difficulté du dessin
- 3.17.1.4 Le temps restant pour deviner le mot doit être affiché à l'écran en tout temps
- 3.17.1.5 Le nombre d'essais restants doit être affiché à l'écran en tout temps
- 3.17.1.6 Le score actuel du joueur doit être affiché en tout temps
- 3.17.1.7 Le joueur gagne un point pour chaque bonne réponse

3.17.1.8 Le joueur gagne du temps pour chaque bonne réponse
3.17.1.8.1 Le temps est ajouté au temps affiché à l'écran
3.17.1.8.2 Le temps ajouté dépend de la difficulté de la paire mot-image
3.17.1.9 Si le joueur dépasse le nombre d'essais accordés, il passe au prochain dessin
3.17.1.9.1 Si le joueur dépasse le nombre d'essais accordés, il obtient zéro
3.17.1.9.1 Si le joueur dépasse le nombre d'essais accordés, il obtient zéro temps bonus
3.17.1.10 Si le joueur dépasse le nombre de temps, la partie est terminée

3.17.2 Le système de jeu doit offrir un mode de jeu sprint coopératif ou plusieurs joueurs collaborent pour deviner les mots ou expressions
3.17.2.1 La partie est limitée dans le temps
3.17.2.2 Le dessin à deviner doit être dessiné par un joueur virtuel
3.17.2.3 L'équipe a un nombre d'essais maximum collectif pour deviner le mot
3.17.2.3.1 Le nombre d'essais est défini en fonction de la difficulté du dessin
3.17.2.4 Le temps restant pour deviner le mot doit être affiché à l'écran en tout temps
3.17.2.5 Le nombre d'essais restants doit être affiché à l'écran en tout temps
3.17.2.6 Le score actuel des joueurs doit être affiché en tout temps
3.17.2.7 Une seule personne doit deviner le mot pour passer au prochain dessin
3.17.2.8 L'équipe gagne un point pour chaque bonne réponse
3.17.2.9 L'équipe gagne du temps pour chaque bonne réponse
3.17.2.9.1 Le temps est ajouté au temps affiché à l'écran
3.17.2.9.2 Le temps ajouté dépend de la difficulté de la paire mot-image
3.17.2.10 Si l'équipe dépasse le nombre d'essais accordés, le système passe au prochain dessin
3.17.2.10.1 Si l'équipe dépasse le nombre d'essais accordés, elle n'obtient aucun point
3.17.2.10.1 Si l'équipe dépasse le nombre d'essais accordés, elle n'obtient aucun bonus de temps

3.17.3 Le système de jeu doit offrir un mode de jeu *battle royal*
3.17.3.1 Le jeu doit permettre au créateur de la partie de choisir le nombre de vies par personne
3.17.3.2 Chaque personne doit deviner le mot le plus vite possible
3.17.3.2.1 La dernière personne à deviner le mot doit perdre une vie
3.17.3.3 Un joueur qui n'a plus de vie doit tomber en mode spectateur - Souhaitable
3.17.3.4 Le joueur qui se retrouve seul à la fin est le joueur gagnant
3.17.3.5 Chaque personne qui devine le mot gagne un point
3.17.3.6 Le joueur gagnant gagne plus de point que les autres joueurs

3.18 Création d'une paire mot-image (Client lourd)

3.18.1 Le système doit permettre aux utilisateurs d'utiliser la banque de dessin « Quick Draw » de Google plutôt que de dessiner l'image lors de la création d'un jeu
3.18.1.1 Le système doit proposer un mot au hasard de « Quick Draw »"
3.18.1.2 Le système doit afficher le dessin correspondant au mot choisi
3.18.1.3 L'utilisateur peut demander une nouvelle proposition

3.19 Informations sur les joueurs (Client lourd et léger)

3.19.1 Le système doit permettre aux utilisateurs de consulter un tableau de classement général de tous les utilisateurs de l'application
3.19.1.1 Le tableau doit afficher les utilisateurs en ordre décroissant de l'utilisateur avec le plus de points à celui qui en a le moins
3.19.1.2 Le tableau doit afficher le nom du joueur
3.19.1.3 Le tableau doit afficher le nombre de points du joueur
3.19.1.4 Le tableau doit afficher le nombre de victoires du joueur
3.19.1.5 Le tableau doit afficher le nombre de victoires pour chaque mode de jeu du joueur
3.19.1.6 Le tableau doit afficher le nombre de parties jouées par le joueur
3.19.1.7 Le tableau doit afficher le nombre de points d'artiste du joueur (pouces vert et rouge)
3.19.2 Le système doit permettre aux utilisateurs d'avoir accès à sa liste d'amis

3.19.2.1 L'utilisateur doit pouvoir visualiser le profil de ses amis
3.19.2.2 L'utilisateur doit pouvoir inviter un ami dans un lobby
3.19.2.3 L'utilisateur doit pouvoir joindre un ami dans son lobby
3.19.2.4 L'utilisateur doit pouvoir clavarder avec un ami
3.19.2.5 L'utilisateur doit pouvoir ajouter un ami
3.19.2.6 L'utilisateur doit pouvoir supprimer un ami

3.20 Tutoriel (Client lourd et léger)

3.20.1 Le tutoriel doit forcer l'utilisateur à cliquer sur les différents boutons de l'application afin de progresser dans le tutoriel

3.20.1.1 Un message doit s'afficher pour indiquer la tâche à faire
3.20.1.2 Le prochain message doit s'afficher automatiquement lorsque l'utilisateur accomplit la tâche
3.20.1.3 Le message courant doit disparaitre automatiquement lorsque l'utilisateur accomplit la tâche

3.21 Déroulement d'une partie (Client lourd et léger)

3.21.1 Le système doit permettre de rejoindre une partie en mode spectateur

3.21.1.1 Chaque spectateur doit voir la partie se dérouler en temps réel
3.21.1.2 Le système doit empêcher les spectateurs d'entrer des réponses
3.21.1.3 L'utilisateur peut rejoindre une partie en tant que spectateur en passant par la liste de lobby
3.21.1.4 Si la liste d'amis est implémentée, l'utilisateur peut rejoindre une partie en tant que spectateur en passant par la liste d'amis

3.22 Page d'accueil (Client lourd et léger)

3.22.6 Se référer à 3.16.1

4. Exigences non fonctionnelles

4.1. Utilisabilité

4.1.1 Le logiciel Fais-moi un dessin doit offrir une interface cohérente sur les clients lourd et léger

4.1.2 Le logiciel Fais-moi un dessin doit présenter une interface facile d'utilisation

4.1.2.1 Les boutons doivent être assez gros pour appuyer facilementt

4.1.2.2 Le texte doit être facilement lisible

4.1.3 Le logiciel Fais-moi un dessin doit offrir une interface visuellement agréable

4.1.3 Les contrastes doivent être suffisamment élevés pour discerner les différents boutons

4.1.4 Le logiciel Fais-moi un dessin doit offrir un dessin cohérent sur les deux clients

4.1.5 Le logiciel Fais-moi un dessin doit être maîtrisable en une partie

4.1.6 Le logiciel Fais-moi un dessin doit être intuitif

4.1.6.1 Les boutons doivent indiquer leur utilité simplement

4.2. Fiabilité

4.2.1. Le serveur doit être accessible 99% du temps

4.2.2 Le serveur doit fonctionner correctement

4.2.3 Le serveur doit fonctionner sans redémarrage

4.2.4 Les clients doivent fonctionner correctement

4.2.5 Les clients doivent fonctionner sans redémarrage

4.2.6 Le système doit offrir un temps moyen entre les pannes d'un mois

4.2.7 Le système doit avoir un temps moyen de 12 heures jusqu'à la réparation

4.3. Performance

4.3.1 Le serveur doit être en mesure de fournir une expérience de jeu sans latence

4.3.2 La synchronisation des plateaux de jeu doit être primordiale pour assurer une bonne expérience de jeu

4.3.3 Le serveur doit pouvoir supporter simultanément la connexion de 8 utilisateurs, c.-à-d. 4 dans deux lobbys différents

4.3.4 Les parties doivent débuter simultanément sur tous les clients de la même partie

4.3.5 Les parties doivent terminer simultanément sur tous les clients de la même partie

4.3.6 Le système doit permettre à l'utilisateur de jouer et de clavarder simultanément sans latence

4.3.7 Le système doit confirmer la validité d'un message de partie dès qu'il est envoyé

4.4. Maintenabilité

4.4.1 Le code devra suivre les normes de programmation utilisée à l'École Polytechnique Montréal

- *Ces normes de programmation sont une adaptation du document « C++ Programming Style Guidelines »*

4.4.2 Le code des fonctionnalités en cours de développement doivent être publié sur GitLab dans une branche nommée <fonctionnalité>-<#numéro ticket jira> ex : architecture-serveur-5

4.4.3 Les messages de commit Git doivent suivre le format suivant [initiales-nom](type): <commentaire>

avec comme types :

- *dev - développement*
- *deb - debug*
- *ref - refactoring*
- *aut - autre*
- *tes - test*

ex : [SA](dev): Modification UI menu principal. Ajout du bouton pour fermer l'application.

4.4.4 Les remises de sprint doivent être faites sur la branche dev

4.4.5 La remise finale doit être fait sur la branche master

4.5. Contraintes de conception

4.5.1 Le client lourd doit utiliser le framework angular

4.5.1.2 Le framework angular doit utiliser HTML, CSS et TypeScript

4.5.2 Le client léger doit utiliser le langage Kotlin, Java et XML

4.5.3 L'environnement de programmation pour le client léger est Android Studio

4.5.4 Le serveur utilisera NodeJs

4.5.3.1 Le langage du serveur doit être JavaScript

4.5.5 Le framework pour créer un exécutable avec Angular est Electron

4.5.6 Android studio est utilisé comme émulateur android

4.6. Sécurité

4.6.1 Le système doit cacher les mots de passe par défaut

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

4.7.1 Le guide de l'utilisateur doit être disponible sur le site internet de l'application
4.7.2 Le tutoriel doit être disponible sur le site internet de l'application

4.8. Normes applicables

4.8.1 Le code écrit doit passer par un processus de merge request sur la branche dev et 2 personnes doivent approuver la requête

ANNEXE A: Glossaire

[Si vous n'utilisez pas cette annexe, veuillez la supprimer complètement.]

Terme	Description
Two-Factor authentication	Authentification en deux étapes: La première étape implique la connexion directe sur le client, ensuite l'utilisateur reçoit un courriel incluant un code qu'il devra entrer sur le client pour le connecter à l'application.
Bitmap	Dessin matriciel