

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

Version 3.1

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
2021-04-14	3.0	Appliquer les correction de l'appelle d'offre	Augustin Bouchard
2021-04-19	3.1	Révision pour la remise du projet	Augustin Bouchard

Table des matières

1. Introduction	5
1.1. But	5
1.2. Définitions, acronymes et abréviations	5
1.3. Vue d'ensemble du document	6
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	6
2.2.4. Interfaces de communication	6
2.3. Contraintes générales	7
2.4. Hypothèses et dépendances	7
3. Exigences fonctionnelles	7
Exigences essentielles	7
3.1. Clavardage - Intégration (Client lourd)	7
3.2. Clavardage - Intégration (Client léger)	7
3.3. Clavardage - Canaux de discussion (Client lourd et léger)	7
3.4 Profil utilisateur et historique (Client lourd et léger)	7
3.5 Modes de jeu (Client lourd et léger)	8
3.6 Création d'une paire mot-image (Client lourd)	9
3.7 Surface de dessin (Client lourd et léger)	9
3.8 Personnalité des joueurs virtuels (Client lourd et léger)	10
3.9 Tutoriel (Client lourd et léger)	10
3.10 Effets visuels et sonores (Client lourd et léger)	10
3.11 Déroulement d'une partie (Client lourd et léger)	11
3.12 Page d'accueil (Client lourd et léger)	11
3.13 Page menu principal des parties (Client lourd et léger)	11
Exigences souhaitables	11
3.14 Interactions de multijoueurs en partie (Client lourd et léger)	11
3.15. Clavardage - Intégration (Client léger)	11
3.16 Profil utilisateur et historique (Client lourd et léger)	11
3.17 Modes de jeu (Client lourd et léger)	12
3.18 Création d'une paire mot-image (Client lourd)	13
3.19 Informations sur les joueurs (Client lourd et léger)	13
3.20 Tutoriel (Client lourd et léger)	13

3.21 Déroulement d'une partie (Client lourd et léger)	13
3.22 Page d'accueil (Client lourd et léger)	14
4. Exigences non fonctionnelles	14
4.1. Utilisabilité	14
4.2. Fiabilité	14
4.3. Performance	14
4.4. Maintenabilité	15
4.5. Contraintes de conception	15
4.6. Sécurité	15
4.7. Exigences de la documentation usager en ligne et du système d'assistance	15
4.8. Normes applicables	15
ANNEXE A: Glossaire	16

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 différents 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.

Chat: Clavardage.

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 conventions permettant le développement de l'application.

Heroku : Serveur distant de déploiement d'applications web. Utilisé pour déployer le serveur du logiciel.

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.

Kotlin: Langage de programmation orienté-objet basé sur Java pour développer l'application android sur Android Studio.

Lobby : Groupe de joueurs associés à une partie.

MS Azure: Microsoft Azure. Plate-forme applicative en nuage de Microsoft. Utilisée pour déployer la base de données.

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.

SVG: Dessin vectoriel.

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

UTF-8: Norme de codage pour les caractères informatiques. UTF-8 inclut toutes les lettres de l'alphabet et plusieurs caractères spéciaux.

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 librairie socket.io pour permettre une communication fluide utile pour le clavier. 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 supposons 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 Le système doit permettre à l'utilisateur d'accéder au clavardage à tout moment
 - 3.2.1.1 Le système doit avoir une interface de clavardage intégrée dans l'application

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

- 3.3.1 Le système doit permettre à l'utilisateur de créer un canal de discussion
- 3.3.2 Le système doit permettre à l'utilisateur de rejoindre un canal de discussion
- 3.3.3 Le système doit permettre à l'utilisateur de quitter un canal de discussion
- 3.3.4 Le système doit détruire un canal de discussion vide
- 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
- 3.3.7 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 système doit présenter un profil contenant une partie privée.
 - 3.4.1.1.1 Le système doit afficher le prénom dans le profil privé
 - 3.4.1.1.2 Le système doit afficher le nom dans le profil privé

- 3.4.1.1.3 Le système doit afficher le courriel dans le profil privé
- 3.4.1.2 Le système doit présenter un profil contenant une partie publique
 - 3.4.1.2.1 Le système doit afficher le pseudonyme dans le profil public
 - 3.4.1.2.2 Le système doit afficher l'avatar dans le profil public
- 3.4.1.3 Le système doit présenter un profil contenant des statistiques sur l'utilisation du jeu.
 - 3.4.1.3.1 Le système doit afficher la statistique du nombre de parties jouées dans le profil
 - 3.4.1.3.2 Le système doit afficher la statistique du pourcentage de victoire dans le profil
 - 3.4.1.3.3 Le système doit afficher la statistique du temps moyen d'une partie dans le profil
 - 3.4.1.3.4 Le système doit afficher la statistique du temps total passé en jeu dans le profil
- 3.4.1.4 Le système doit présenter un profil contenant un historique détaillé
 - 3.4.1.4.1 Le système doit afficher les dates et heures de connexion de l'utilisateur dans le profil
 - 3.4.1.4.2 Le système doit afficher les dates et heures de déconnexion de l'utilisateur dans le profil
 - 3.4.1.4.3 Le système doit afficher l'historique des parties jouées dans le profil
 - 3.4.1.4.3.1 Le système doit afficher la date de la partie dans l'historique.
 - 3.4.1.4.3.2 Le système doit afficher l'heure de la partie dans l'historique.
 - 3.4.1.4.3.3 Le système doit afficher le pseudonyme des joueurs ayant joué dans la partie dans l'historique.
 - 3.4.1.4.3.4 Le système doit afficher le résultat de la partie dans l'historique.
 - 3.4.1.4.3.5 Le système doit afficher le mode de jeu de la partie dans l'historique.
- 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 permettre 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.
 - 3.5.1.4 Le système doit afficher le temps restant d'une manche.
 - 3.5.1.5 Le système doit afficher le nombre de manches jouées.
 - 3.5.1.6 Le système doit afficher le nombre d'indices demandés.
 - 3.5.1.7 Le système doit afficher le nombre de tentatives.
- 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 Le système doit ajouter automatiquement des joueurs virtuels s'il y a moins de 4 joueurs réels.
 - 3.5.2.1.1 Le système doit donner le rôle de dessinateur à un joueur virtuel pour l'entièreté de la partie lorsqu'il n'observe pas.
 - 3.5.2.1.2 Le système doit donner le rôle de devineur à un joueur réel en équipe avec un joueur virtuel pour l'entièreté de la partie lorsqu'il n'observe pas.
 - 3.5.2.2 Le système doit ajouter un dessinateur dans chaque équipe.
 - 3.5.2.3 Le système doit ajouter un devineur dans chaque équipe.
 - 3.5.2.4 Le système doit donner 0 point à l'équipe lorsque le joueur devineur atteint son maximum de tentatives.
 - 3.5.2.4.1 Le système doit donner un droit de répliquer à l'autre équipe.
 - 3.5.2.4.1.1 Le système doit donner 0 point à l'autre équipe si elle se trompe une fois.
 - 3.5.2.4.1.2 Le système doit terminer la manche lorsque chaque équipe a fait ses tentatives.
 - 3.5.2.4.1.3 Le système doit terminer la manche 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 Le système doit permettre à l'utilisateur de choisir le niveau de difficulté facile.
 - 3.5.3.2 Le système doit permettre à l'utilisateur le niveau de difficulté intermédiaire.
 - 3.5.3.3 Le système doit permettre à l'utilisateur le niveau de difficulté difficile.

- 3.5.3.4 Le système doit ajuster la vitesse de dessin des joueurs virtuels selon le niveau de difficulté.
- 3.5.3.5 Le système doit allouer un temps différent aux joueurs selon le niveau de difficulté.
 - 3.5.3.5.1 Le système doit allouer 60 secondes aux joueurs au niveau facile.
 - 3.5.3.5.2 Le système doit allouer 40 secondes aux joueurs au niveau intermédiaire.
 - 3.5.3.5.3 Le système doit allouer 20 secondes aux joueurs au niveau difficile.
- 3.5.3.6 Le système doit allouer un nombre de tentatives différent aux joueurs pour deviner le mot selon le niveau de difficulté.
 - 3.5.3.6.1 Le système doit allouer 5 tentatives au niveau facile.
 - 3.5.3.6.2 Le système doit allouer 3 tentatives au niveau intermédiaire.
 - 3.5.3.6.3 Le système doit allouer 1 tentative au niveau difficile.

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 Le système doit obliger l'utilisateur à fournir le mot recherché.
 - 3.6.1.2 Le système doit obliger l'utilisateur à fournir au moins un indice.
 - 3.6.1.3 Le système doit fournir une zone de dessin pour dessiner l'image.
 - 3.6.1.3.1 Le système doit fournir la même zone de dessin d'une partie avec les mêmes outils.
 - 3.6.1.4 Le système doit obliger l'utilisateur à fournir 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 Le système doit permettre à l'utilisateur de 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 Le système doit, par défaut, sélectionner le mode de dessin conventionnel.
 - 3.6.2.2 Le système doit permettre à l'utilisateur de choisir le mode aléatoire où les traits apparaissent dans un ordre aléatoire.
 - 3.6.2.2.1 Le système doit afficher les traits dans un ordre différent à chaque partie en mode aléatoire.
 - 3.6.2.3 Le système doit permettre à l'utilisateur de choisir le mode panoramique où chaque trait apparaît dans l'ordre de leur position selon une direction choisie.
 - 3.6.2.3.1 Le système doit permettre de choisir de faire apparaître les traits de droite à gauche.
 - 3.6.2.3.2 Le système doit permettre de choisir de faire apparaître les traits de gauche à droite.
 - 3.6.2.3.3 Le système doit permettre de choisir de faire apparaître les traits de haut à bas.
 - 3.6.2.3.4 Le système doit permettre de choisir de faire apparaître les traits de bas à haut.
 - 3.6.2.4 Le système doit permettre à l'utilisateur de choisir le mode centré où les traits apparaissent dans l'ordre de leur distance du centre de l'image.
 - 3.6.2.4.1 Le système doit permettre de choisir de faire apparaître les traits de l'intérieur vers l'extérieur.
 - 3.6.2.4.2 Le système doit permettre de choisir de faire apparaître les traits de l'extérieur vers l'intérieur.
- 3.6.3 Le système doit permettre à l'utilisateur d'importer une image de son ordinateur plutôt que de dessiner l'image lors de la création d'un jeu.
 - 3.6.3.1 Le système doit permettre l'importation d'une image de format matriciel (type bitmap).
 - 3.6.3.2 Le système doit convertir l'image importée en format SVG.
 - 3.6.3.2.1 Le système doit permettre de choisir les différentes variables de l'engin de conversion sauf conventionnel.
- 3.6.4 Le système doit permettre à l'utilisateur de prévisualiser le dessin.

3.7 Surface de dessin (Client lourd et léger)

- 3.7.1 Le système doit permettre de dessiner sur une surface de dessin à l'aide d'une option crayon.
 - 3.7.1.1 Le système doit permettre de changer l'épaisseur du crayon.
 - 3.7.1.2 Le système doit permettre de changer la couleur du crayon à l'aide d'une palette complète.
 - 3.7.1.3 Le système doit permettre de changer l'opacité du crayon.
 - 3.7.1.4 Le système doit permettre de sélectionner une des 10 dernières couleurs.
- 3.7.2 Le système doit permettre d'effacer les traits de crayon à l'aide d'une option efface.

- 3.7.2.1 Le système doit permettre de changer l'épaisseur de l'efface.
- 3.7.2.2 Le système doit permettre d'effacer en brosse.
- 3.7.2.3 Le système doit changer temporairement la couleur du trait survolé par l'efface à rouge.
 - 3.7.2.3.1 Le système doit donner une teinte de rouge plus foncé au trait s'il est déjà rouge.
- 3.7.2.4 Le système doit retirer seulement le trait le plus au-dessus (chevauchement) lorsqu'on efface avec un clic.
- 3.7.3 Le système doit retirer et remettre un à un les traits de crayon dessinés à l'aide d'une option annuler/refaire.
 - 3.7.3.1 Le système doit permettre d'annuler les traits dans l'ordre du plus récent.
 - 3.7.3.2 Le système doit permettre de refaire les traits annulés.
 - 3.7.3.3 Le système doit rendre indisponible l'option refaire dès la création d'un nouveau trait.
 - 3.7.3.4 Le système doit commencer un trait de crayon au moment où le crayon touche la surface de dessin.
 - 3.7.3.5 Le système doit terminer un trait de crayon au moment où le crayon quitte la surface de dessin.
- 3.7.4 Le système doit permettre d'afficher une grille sur sa surface de dessin.
 - 3.7.4.1 Le système doit afficher la grille pour l'utilisateur qui dessine seulement.
 - 3.7.4.2 Le système doit permettre de changer la grosseur du quadrillage de la grille.
- 3.7.5 Le système doit permettre de choisir la couleur désirée à l'aide d'un outil de sélection de couleurs.
 - 3.7.5.1 Le système doit garder la couleur choisie en mémoire dans une liste de présélections.
 - 3.7.5.2 Le système doit permettre de choisir l'opacité de la couleur.
- 3.7.6 Le système doit s'assurer que la surface de dessin soit 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 système doit forcer le joueur virtuel à fournir des indices lorsque demandé.
 - 3.8.1.2 Le système doit forcer le joueur virtuel à encourager son coéquipier.
 - 3.8.1.3 Le système doit forcer le joueur virtuel à féliciter son coéquipier.
 - 3.8.1.4 Le système doit forcer le joueur virtuel à publier un message au début de la partie.
 - 3.8.1.5 Le système doit forcer le joueur virtuel à 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 Le système doit avoir des joueurs virtuels avec des caractères uniques.
 - 3.8.2.2 Le système doit forcer un joueur virtuel à conserver son caractère durant toute la partie.
 - 3.8.2.3 Le système doit avoir un joueur virtuel avec une personnalité calme.
 - 3.8.2.4 Le système doit avoir un joueur virtuel avec une personnalité enfantine.
 - 3.8.2.5 Le système doit avoir un joueur virtuel avec une personnalité encourageante .
 - 3.8.2.6 Le système doit avoir un joueur virtuel avec une personnalité colérique.
- 3.8.3 Le système doit permettre au joueur virtuel d'avoir des conversations dynamiques
 - 3.8.3.1 Le système doit forcer le joueur virtuel à référer des parties jouées dans le canal de discussion.
 - 3.8.3.2 Le système doit forcer le joueur virtuel à référer des joueurs rencontrés dans le canal de discussion.
 - 3.8.3.3 Le système doit forcer le joueur virtuel à référer le pourcentage de parties gagnées des joueurs pour formuler des commentaires personnalisés dans le canal de discussion.

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 effet de changement de gradient de couleur lors du choix de lobby.

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 Le système doit indiquer que le mot est « 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 système doit permettre au joueur d'entrer sa tentative de mot à deviner dans le canal de discussion de la partie.
 - 3.11.2.1 Le système doit réserver le canal de discussion de la partie aux joueurs de la partie.
 - 3.11.2.2 Le système doit envoyer le message 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 Le système doit permettre à l'utilisateur de quitter le logiciel.
- 3.12.2 Le système doit inviter l'utilisateur à entrer son pseudonyme et son mot de passe pour se connecter à son profil sur la page d'accueil.
- 3.12.3 Le système doit permettre à l'utilisateur d'accéder à une page d'inscription pour créer un profil.
 - 3.12.3.1 Le système doit permettre à l'utilisateur de choisir un pseudonyme.
 - 3.12.3.2 Le système doit permettre à l'utilisateur de choisir un email.
 - 3.12.3.3 Le système doit permettre à l'utilisateur de choisir un mot de passe.
- 3.12.4 Le système doit permettre à l'utilisateur d'accéder au menu principal des parties.

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

- 3.13.1 Le système doit permettre à l'utilisateur de sélectionner le mode de jeu auquel il souhaite accéder.
- 3.13.2 Le système doit permettre à l'utilisateur de sélectionner la difficulté souhaitée de la partie.
- 3.13.3 Le système doit permettre à l'utilisateur de créer un lobby.
- 3.13.4 Le système doit permettre à l'utilisateur de rejoindre un lobby.
 - 3.13.4.1 Le système doit permettre à l'utilisateur de consulter la liste des joueurs réels dans le lobby.
 - 3.13.4.2 Le système doit ajouter le bon nombre de joueurs virtuels pour combler un lobby.
 - 3.13.4.3 Le système doit ajouter automatiquement l'utilisateur au canal de discussion de la partie lorsqu'il rejoint le lobby.
 - 3.13.4.4 Le système doit permettre à tous les utilisateurs de démarrer une partie.
 - 3.13.4.5 Le système doit permettre à l'utilisateur de commencer la partie 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 vers le haut.
 - 3.14.1.2 Le système doit permettre aux utilisateurs d'attribuer un pouce vers le bas.
 - 3.14.1.3 Le système doit collecter les pouces dans les statistiques de chaque joueur.
 - 3.14.1.4 Le système doit afficher un pouce à côté du nom de l'utilisateur.
 - 3.14.1.5 Le système doit faire disparaître un pouce 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 Le système doit faire disparaître l'indicateur visuel lorsque tous les canaux contenant de nouveaux messages ont été 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 Le système doit permettre à l'utilisateur de 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 Le système doit envoyer un code par courriel à l'utilisateur lors de chaque connexion d'un 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 Le système doit permettre à l'utilisateur de décider s'il utilise le « Two-Factor Authentication » dans les paramètres de son compte.

3.16.2.4 Le système doit désactiver le « Two-Factor Authentication » 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 où il faut trouver le plus de mots ou expressions possibles.

3.17.1.1 Le système de jeu doit offrir trois modes de difficulté pour la partie. (Se référer à 3.5.3)

3.17.1.2 Le système doit dessiner le mot à deviner par un joueur virtuel.

3.17.1.3 Le système doit allouer un nombre d'essais maximum pour deviner le mot ou l'expression.

3.17.1.3.1 Le système doit allouer le nombre d'essais selon la difficulté du dessin.

3.17.1.4 Le système doit afficher le temps restant pour deviner le mot à l'écran en tout temps.

3.17.1.5 Le système doit afficher le nombre d'essais restants à l'écran en tout temps.

3.17.1.6 Le système doit afficher le score actuel du joueur à l'écran en tout temps.

3.17.1.7 Le système doit donner un point au joueur pour chaque bonne réponse.

3.17.1.8 Le système doit donner du temps supplémentaire au joueur pour chaque bonne réponse.

3.17.1.8.1 Le système doit ajouter le temps au temps affiché à l'écran.

3.17.1.8.2 Le système doit ajouter le temps selon la difficulté de la paire mot-image.

3.17.1.9 Le système doit passer au prochain dessin si le joueur dépasse le nombre d'essais accordés.

3.17.1.9.1 Le système doit donner 0 points si le joueur dépasse le nombre d'essais accordés.

3.17.1.9.1 Le système doit donner 0 temps bonus si le joueur dépasse le nombre d'essais accordés.

3.17.1.10 Le système doit terminer la partie lorsque le joueur dépasse le nombre de temps accordé.

3.17.2 Le système de jeu doit offrir un mode de jeu sprint coopératif où plusieurs joueurs collaborent pour deviner les mots ou expressions

3.17.2.1 Le système de jeu doit offrir trois modes de difficulté pour la partie. (Se référer à 3.5.3)

3.17.2.2 Le système doit dessiner le mot à deviner par un joueur virtuel.

3.17.2.3 Le système doit allouer un nombre d'essais maximum pour deviner le mot ou l'expression.

3.17.2.3.1 Le système doit allouer le nombre d'essais selon la difficulté du dessin.

3.17.2.4 Le système doit afficher le temps restant pour deviner le mot à l'écran en tout temps.

3.17.2.5 Le système doit afficher le nombre d'essais restants à l'écran en tout temps.

3.17.2.6 Le système doit afficher le score actuel du joueur à l'écran en tout temps.

3.17.2.7 Le système doit passer au prochain dessin lorsqu'une personne devine le mot.

3.17.2.8 Le système doit donner un point à l'équipe pour chaque bonne réponse.

3.17.2.9 Le système doit donner du temps supplémentaire à l'équipe pour chaque bonne réponse.

3.17.2.9.1 Le système doit ajouter le temps au temps affiché à l'écran.

3.17.2.9.2 Le système doit ajouter le temps selon la difficulté de la paire mot-image.

3.17.2.10 Le système doit passer au prochain dessin si l'équipe dépasse le nombre d'essais accordés.

3.17.2.10.1 Le système doit donner 0 points si l'équipe dépasse le nombre d'essais accordés.

3.17.2.10.1 Le système doit donner 0 temps bonus si l'équipe dépasse le nombre d'essais accordés.

3.17.2.11 Le système doit terminer la partie lorsque le joueur dépasse le nombre de temps accordé.

3.17.3 Le système de jeu doit offrir un mode de jeu *battle royal*

3.17.3.1 Le système doit allouer un nombre de vies par personne selon la difficulté de la partie.

3.17.3.1.1 Le système doit allouer 5 vies par personne pour la difficulté facile.

3.17.3.1.2 Le système doit allouer 3 vies par personne pour la difficulté intermédiaire.

3.17.3.1.3 Le système doit allouer 1 vie par personne pour la difficulté difficile.

3.17.3.2 Le système doit donner des points selon l'ordre de personnes qui devine le mot.

- 3.17.3.2.1 Le système doit donner 2 points à la première personne qui devine le mot.
- 3.17.3.2.2 Le système doit donner 1 point à la deuxième personne qui devine le mot.
- 3.17.3.2.3 Le système doit donner 0 point à la dernière personne.
- 3.17.3.2.4 Le système doit enlever une vie à la dernière personne.
- 3.17.3.3 Le système doit mettre un joueur qui n'a plus de vie en mode spectateur. - Souhaitable
- 3.17.3.4 Le système doit attribuer la victoire au joueur qui se retrouve seul à la fin.
- 3.17.3.5 Le système doit allouer 60 secondes pour une manche.
- 3.17.3.6 Le système doit dessiner le mot à deviner par un joueur virtuel.

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.
 - 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 Le système doit permettre à l'utilisateur de 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 système doit afficher le tableau en ordre décroissant de l'utilisateur avec le plus de victoires Classique et Bataille Royale combinés. .
 - 3.19.1.4 Le système doit afficher le nombre de victoires du joueur Classique et Bataille Royale combinés.
 - 3.19.1.2 Le système doit afficher le rang du joueur.
 - 3.19.1.2 Le système doit afficher le nom du joueur.
 - 3.19.1.3 Le système doit afficher le nombre de points du joueur de tous les modes combinés.
 - 3.19.1.4 Le système doit afficher le nombre de victoires du joueur.
 - 3.19.1.5 Le système doit afficher le nombre de victoires pour chaque mode de jeu du joueur.
 - 3.19.1.6 Le système doit afficher le nombre de parties jouées par le joueur.
 - 3.19.1.7 Le système doit afficher le nombre de points d'artiste du joueur (pouces vert et rouge).
- 3.19.2 Le système doit permettre à l'utilisateur d'avoir accès à sa liste d'amis.
 - 3.19.2.1 Le système doit permettre à l'utilisateur de visualiser le profil d'un ami.
 - 3.19.2.2 Le système doit permettre à l'utilisateur d'inviter un ami dans un lobby.
 - 3.19.2.3 Le système doit permettre à l'utilisateur de joindre un ami dans son lobby.
 - 3.19.2.4 Le système doit permettre à l'utilisateur de clavarder avec un ami.
 - 3.19.2.5 Le système doit permettre à l'utilisateur d'ajouter un ami.
 - 3.19.2.6 Le système doit permettre à l'utilisateur de supprimer un ami.

3.20 Tutoriel (Client lourd et léger)

- 3.20.1 Le système doit forcer l'utilisateur à cliquer sur les différents boutons de l'application afin de progresser dans le tutoriel.
 - 3.20.1.1 Le système doit afficher un message pour indiquer la tâche à faire.
 - 3.20.1.2 Le système doit afficher le prochain message lorsque l'utilisateur accomplit la tâche.
 - 3.20.1.3 Le système doit faire disparaître le message courant 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 Le système doit permettre à chaque spectateur de 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 Le système doit permettre à l'utilisateur de rejoindre une partie en tant que spectateur en passant par la liste de lobby.
 - 3.21.1.4 Le système doit permettre à l'utilisateur de rejoindre une partie en tant que spectateur en passant par la liste d'amis si elle est implémentée.

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 facilement.

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.

4.1.3.1 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 fonctionnalité simplement.

4.2. Fiabilité

4.2.1. Le serveur doit être accessible 85% du temps.

4.2.2 Le serveur doit fonctionner correctement.

4.2.3 Le serveur doit fonctionner sans redémarrage.

4.2.4 Le client doit fonctionner correctement.

4.2.5 Le client doit 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 Les plateaux de jeu doivent être synchronisés.

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 Une partie doit débiter avec un délai maximal de 4 secondes sur tous les clients de la même partie.

4.3.5 Une partie doit terminer avec un délai maximal de 4 secondes sur tous les clients de la même partie.

4.3.6 Le système doit permettre à l'utilisateur de jouer et de clavier avec un délai maximal de transition de 0.5 seconde.

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 protéger le mot de passe de l'utilisateur.

4.6.2 Le système doit protéger l'adresse courriel de l'utilisateur.

4.6.3 Le système doit protéger l'historique de connexion de l'utilisateur.

4.6.4 Le système doit protéger l'id de l'utilisateur.

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

4.7.1 Le tutoriel doit être disponible sur 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

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