

**PAINseau**

**Projet 3**  
**Spécifications des requis du système (SRS)**

**Version 2.0**

## Historique des révisions

Date	Version	Description	Auteur
2021-01-26	1.0	Rédaction initiale du SRS.	Équipe 107
2021-02-02	1.1	Révision du SRS suite aux modifications finales à la liste d'exigence	Équipe 107
2021-02-17	2.0	Révision du SRS suite à la première correction	Équipe 107

# Table des matières

1. Introduction	1
1.1. But	1
1.2. Définitions, acronymes et abréviations	1
1.3. Vue d'ensemble du document	3
2. Description globale	3
2.1. Caractéristiques des usagers	3
2.2. Interfaces	3
2.2.1. Interfaces usagers	3
2.2.2. Interfaces matérielles	3
2.2.3. Interfaces logicielles	3
2.2.4. Interfaces de communication	3
2.3. Contraintes générales	4
2.4. Hypothèses et dépendances	4
3. Exigences fonctionnelles	4
Exigences fonctionnelles pour client léger et client lourd	4
3.1 Clavardage	4
3.2 Profils	5
3.3 Modes de jeu	5
3.4 Paire mot-image	7
3.5 Personnalité des joueurs virtuels	8
3.6 Effets visuels et sonores	8
3.7 Tutoriel	8
3.8 Langues	9
3.9 Couleurs/Thèmes	9
3.10 Leaderboard	9
3.11 Système de points/niveau	9
3.12 Album de dessins	9
3.13 Exigences fonctionnelles pour client léger seulement:	10
4. Exigences non-fonctionnelles	11
4.1. Utilisabilité	11
4.2. Fiabilité	11
4.3. Performance	11
4.4. Maintenabilité	11
4.5. Contraintes de conception	11
4.6. Sécurité	12
4.7. Exigences de la documentation usager en ligne et du système d'assistance	12

# Spécifications des requis du système (SRS)

## 1. Introduction

### 1.1. But

Ce document a pour but de fournir une description détaillée du comportement externe du logiciel *Fais-moi un dessin* et de ses spécifications. Il décrit les exigences non fonctionnelles et fonctionnelles, les contraintes de conception, ainsi que les autres facteurs nécessaires à la description complète des exigences du logiciel à développer. Ce document s'adresse principalement à PolyApps afin d'obtenir leur approbation, et servira à titre de référence lors du développement du logiciel en question.

### 1.2. Définitions, acronymes et abréviations

[E] : (Essentiel) Exigence obligatoire à implémenter.

[S] : (Souhaitable) Exigence non-obligatoire à implémenter.

Administrateurs: Groupe d'utilisateurs comprenant les membres de l'équipe et possédant des droits propres à eux (ex: ils peuvent modifier les profils des utilisateurs, les dessins sauvegardés dans la base de données, retirer les droits d'administrateur, etc.)

Android : Système d'exploitation mobile.

Angular : Cadriciel open-source permettant de programmer des applications Web qui sera utilisé pour le client lourd.

Base de données: Collection organisée de données permet d'organiser, d'analyser et de stocker ces données.

BSON: Format binaire d'échange de données informatiques pour le stockage de données en MongoDB.

CI: Ensemble de pratiques consistant d'une intégration régulière des changements apportés à un code source, suivie de tests automatisés. (en anglais « Continuous Integration »)

CLI: Interface en ligne de commande (en anglais « Command Line Interface »)

Client léger: Application permettant l'accès à un serveur par l'entremise d'un réseau. Exécutée sur des appareils mobiles Android de l'utilisateur

Client lourd: Application permettant l'accès à un serveur par l'entremise d'un réseau. Exécutée sur des PC (Windows 10).

Dark: Thème de couleur de notre application avec une polarité de contraste négative (texte de police claire sur un fond sombre)

Electron : Environnement permettant de développer des applications multi-plateformes de bureau avec des technologies web.

Équipe active : L'équipe présentement en train de dessiner et de deviner l'image.

Équipe en attente : L'équipe adverse en train d'attendre son tour pour deviner et dessiner l'image.

Express: Cadriciel utilisé pour construire notre application du côté serveur.

Exynos : Série de systèmes sur puce créée par la marque Samsung.

GB : Gigabytes

Heroku : Plateforme permettant le déploiement d'applications web en hébergeant des serveurs en ligne.

HTTP: Protocole de communication client-serveur (en anglais «Hypertext Transfer Protocol »)

JavaScript: Langage de programmation utilisé pour le développement Web.

Joueur virtuel: Joueur qui est simulé.

Kotlin : Langage de programmation utilisé pour le client léger, permettant de programmer des applications mobiles pour Android.

Light: Thème de couleur de notre application avec une polarité de contraste positive (texte de police sombre sur un fond clair)

MB : Megabytes

Merge request: Requête pour fusionner une première branche à une seconde.

MongoDB : MongoDB est un système de gestion de base de données orienté documents, avec lequel nous stockerons des informations pertinentes à notre logiciel.

ms : milliseconde

NodeJS: Plateforme logicielle libre (« open source ») en JavaScript, utilisée pour le développement d'applications côté serveur et réseau.

Partie: Durée de temps totale incluant tous les tours des 2 équipes.

PC: Ordinateur personnel (« Personal Computer » en anglais)

Potrace : Potrace est un logiciel open source multiplateforme qui convertit les images bitmap en graphiques vectoriels.

RAM : Mémoire vive dans laquelle on peut enregistrer les informations traitées par un appareil informatique.

Socket.io: Bibliothèque JavaScript utilisée pour les applications Web en temps réel, afin de permettre une communication bidirectionnelle entre le côté client et le serveur.

Système: Inclut l'interface et le logiciel *Fais-moi un dessin*.

Tablette: Dispositif mobile doté d'un système d'exploitation mobile et d'un écran tactile.

TCP: Protocole standard pour la communication entre multiples dispositifs électroniques à distance (« Transmission Control Protocol » en anglais).

Tour: Durée de temps durant laquelle un joueur dessine et son coéquipier tente de deviner le dessin.

TS-LINT : Outil d'analyse statique du code source Typescript pour détecter les erreurs de maintenance, fonctionnalité et lisibilité.

Typescript: Langage de programmation libre et open source qui est utilisé pour le client lourd.

VSCode : Visual Studio Code est un éditeur de code extensible développé par Microsoft pour Windows, Linux et macOS.

WUXGA : Le WUXGA ou Wide Ultra Extended Graphics Array est une norme d'affichage correspondant à un UXGA large dont la définition est de 1 920×1 200 pixels, soit 2 304 000 pixels.

XML: Méta langage de balisage qui permet de définir différents langages (« Extensible Markup Language » en anglais)

### 1.3. Vue d'ensemble du document

Ce présent document se divise en trois sections: la section 2 présente une description générale du logiciel, soit les caractéristiques des usagers, les différentes interfaces, les contraintes de conception et les hypothèses et les dépendances. Puis, les sections 3 et 4 discutent des exigences fonctionnelles et non fonctionnelles du logiciel.

## 2. Description globale

Le logiciel *Fais-moi un dessin* est un jeu interactif, accessible par Android ou PC, où il faut deviner le dessin d'autrui et vice-versa. Cette application permet à plusieurs usagers de jouer en même temps pour deviner le dessin des autres et ainsi accumuler des points. Ces parties multijoueurs sont basées sur un serveur qui doit pouvoir supporter plusieurs parties simultanément. Une fonction de clavardage entre les joueurs est également disponible dès qu'un usager se connecte sur le serveur.

### 2.1. Caractéristiques des usagers

Il existe plusieurs usagers potentiels qui pourront utiliser ce logiciel. La catégorie principale regroupe les gens qui utiliseront notre application, soit nos clients. Ces derniers peuvent se retrouver dans n'importe quelle tranche d'âge et n'ont pas nécessairement de connaissances précises ou spécifiques à notre logiciel. Pour plus de précisions, consultez la section 2.4 *Hypothèses et dépendances*.

### 2.2. Interfaces

#### 2.2.1. Interfaces usagers

La première interface utilisateur est celle du client lourd, qui sera développée en HTML avec Angular 8. Un menu principal permettra à l'utilisateur de naviguer dans les options de son choix. L'interface utilisateur pour le client léger sera très similaire, mais sur tablette Android, et sera développée en Kotlin 1.4.21.

#### 2.2.2. Interfaces matérielles

Pour le client lourd, les interfaces matérielles associées sont l'écran de l'ordinateur, le clavier et la souris. Pour le client léger, l'écran tactile de la tablette constituera l'interface matérielle principale sur laquelle l'utilisateur pourra sélectionner les options et dessiner, tout à partir de plusieurs points de contact.

#### 2.2.3. Interfaces logicielles

Le client lourd sera développé sur Windows 10 et le client léger sur Android. Quant à l'interface logicielle du serveur, ce dernier sera hébergé sur Heroku. Le cadriciel Angular 8 sera utilisé pour implémenter l'interface du logiciel, et le client lourd utilisera aussi le cadriciel Electron. La base de données MongoDB sera utilisée pour stocker les dessins ainsi que les informations sur les profils des utilisateurs et l'historique des messages de clavardage à l'aide de MongoDB Atlas. Le logiciel Potrace sera utilisé pour permettre à l'utilisateur de convertir des images en dessin.

#### 2.2.4. Interfaces de communication

Le logiciel communique à l'aide d'un serveur Windows implémenté en Node.js. Des requêtes HTTP seront exécutées

pour le logiciel et la communication entre les clients et le serveur se fera à l'aide de socket.io, qui utilisera le protocole TCP. La communication se fera par internet puisque le serveur est hébergé sur Heroku.

## 2.3. Contraintes générales

4 joueurs doivent être en mesure de jouer une partie en même temps et 10 joueurs devraient être capables de se connecter simultanément sur le serveur sans que ce dernier ait de problème de performance. La limite d'un BSON document dans MongoDB est de 16MB et les dessins ne peuvent donc dépasser cette taille. Le logiciel doit utiliser un maximum de 1GB de mémoire vive et doit utiliser moins de 500MB d'espace pour le client léger. Quant au client lourd, le logiciel doit utiliser un maximum de 2GB de mémoire vive et doit utiliser moins de 1GB d'espace sur le disque dur.

## 2.4. Hypothèses et dépendances

Les fonctionnalités pour dessiner utilisées dans notre logiciel sont basées sur celles développées lors du projet de LOG2990 et constituent donc une dépendance. Nous émettons l'hypothèse que tous les utilisateurs ont au moins des connaissances de base avec un ordinateur et une tablette (en mesure d'utiliser le clavier, la souris et l'écran tactile de manière aisée). Nous supposons aussi que les joueurs ont accès à une connexion au réseau suffisamment rapide et stable pour que l'application performe en bonne et due forme. Nous supposons également que les utilisateurs utilisant un client léger ont une tablette avec au minimum une taille d'écran de 10.1", une résolution d'écran 1920 x 1200 (WUXGA), un processeur Octa Core (Dual 1.8GB + Hexa 1.6GHz) Lassen O+ (Exynos 7904A), une mémoire RAM de 2GB, une mémoire de 32GB avec une version d'Android 9.0 Pie. Quant au client lourd, nous supposons qu'il a au minimum une résolution d'écran 1920 x 1080, une mémoire RAM de 8GB, une mémoire de 256GB et le système d'exploitation Windows 10.

## 3. Exigences fonctionnelles

Cette section présente les exigences fonctionnelles du logiciel *Fais-moi un dessin*, c'est-à-dire les fonctionnalités du client lourd et léger.

### Exigences fonctionnelles pour client léger et client lourd

#### 3.1 Clavardage

##### 3.1.1 Chatbox (Clavardage - Intégration)

- 3.1.1.1 [E] Le système doit permettre à l'utilisateur de clavarder dans un mode intégré.
- 3.1.1.2 [E] Le système doit permettre à l'utilisateur d'accéder au clavardage dès qu'il est connecté au serveur.
- 3.1.1.3 [E] Le système doit permettre l'usage du clavardage au cours d'une partie.

##### 3.1.2 Canaux de discussion

- 3.1.2.1 [E] Le système doit permettre à un utilisateur de créer un canal de discussion à partir du menu principal en entrant un nom unique de canal.
- 3.1.2.2 [E] Le système doit permettre à un utilisateur de joindre un canal créé en entrant son nom à partir du menu principal.
- 3.1.2.3 [E] Le système doit permettre à un utilisateur de joindre un canal existant en choisissant le canal dans une liste à partir du menu principal.
- 3.1.2.4 [E] Le système doit créer un canal temporaire de groupe pour tous les utilisateurs d'une même séance de jeu.
- 3.1.2.5 [E] Le système doit supprimer le canal temporaire pour les utilisateurs d'une séance de jeu lors de la fin du jeu.
- 3.1.2.6 [E] Le système doit permettre à un utilisateur de joindre plusieurs canaux différents en même temps.
- 3.1.2.7 [E] Le système doit permettre à un utilisateur de naviguer entre les différents canaux.

- 3.1.2.8 [E] Le système doit permettre à un utilisateur de seulement quitter les canaux créés par les utilisateurs..
- 3.1.2.9 [E] Le système doit permettre à l'utilisateur de consulter l'historique de clavardage des canaux de discussion rejoins.
- 3.1.2.10 [E] Le système doit permettre à tous les utilisateurs d'être connectés au canal principal en tout temps.

## **3.2 Profils**

### *3.2.1 Authentification*

- 3.2.1.1 [E] Le système doit permettre à l'utilisateur de choisir un pseudo lors de la création du compte.
- 3.2.1.2 [E] Le système doit permettre à l'utilisateur de choisir un mot de passe lors de la création du compte.
- 3.2.1.3 [E] Le système doit permettre à l'utilisateur de choisir un avatar entre 10 différents avatars lors de la création du compte.
- 3.2.1.4 [E] Le système doit permettre à l'utilisateur de choisir un prénom et un nom lors de la création du compte.
- 3.2.1.5 [E] Le système doit permettre à l'utilisateur de modifier le prénom et nom associé à son compte.
- 3.2.1.6 [E] Le système doit permettre à l'utilisateur de modifier l'avatar associé à son compte.
- 3.2.1.7 [E] Le système doit permettre à l'utilisateur d'avoir une partie privée et une partie publique sur son profil.
- 3.2.1.8 [E] Le système doit afficher le prénom et le nom d'un utilisateur dans la partie privée de son profil.
- 3.2.1.9 [E] Le système doit afficher le pseudo et l'avatar d'un utilisateur dans la partie publique de son profil.
- 3.2.1.10 [E] Le système doit permettre à un utilisateur de voir la partie publique du profil d'un autre utilisateur.
- 3.2.1.11 [E] Le système doit permettre à l'utilisateur de s'authentifier sur le serveur avec son pseudo et son mot de passe.
- 3.2.1.12 [E] Le système doit permettre seulement une connexion à un compte à la fois en tout temps.

### *3.2.2 Statistiques du jeu*

- 3.2.2.1 [E] Le système doit stocker le nombre de parties jouées de chaque utilisateur.
- 3.2.2.2 [E] Le système doit stocker le nombre de parties gagnées de chaque utilisateur.
- 3.2.2.3 [E] Le système doit stocker le temps de chaque partie de chaque utilisateur.
- 3.2.2.4 [E] Le système doit afficher le nombre de parties jouées dans la partie privée de son profil.
- 3.2.2.5 [E] Le système doit afficher le pourcentage de victoire dans la partie privée de son profil.
- 3.2.2.6 [E] Le système doit afficher le temps moyen des parties dans la partie privée de son profil.
- 3.2.2.7 [E] Le système doit afficher le temps total passé à jouer d'un utilisateur dans la partie privée de son profil.
- 3.2.2.8 [S] Le système doit stocker le meilleur score dans le mode Sprint Solo de chaque utilisateur.
- 3.2.2.9 [S] Le système doit afficher le meilleur score dans le mode Sprint Solo d'un utilisateur dans la partie privée du profil.
- 3.2.2.10 [S] Le système doit stocker un historique de toutes les connexions en forme de dates et heures.
- 3.2.2.11 [S] Le système doit stocker un historique de toutes les déconnexions en forme de dates et heures.

### *3.2.3 Historique des parties*

- 3.2.3.1 [S] Le système doit stocker la date du jeu des parties jouées de chaque utilisateur.
- 3.2.3.2 [S] Le système doit stocker les pseudos des joueurs des parties jouées de chaque utilisateur en mode classique.
- 3.2.3.3 [S] Le système doit stocker le score final des parties jouées de chaque utilisateur en mode classique.
- 3.2.3.4 [S] Le système doit stocker le mode de jeu des parties jouées de chaque utilisateur.
- 3.2.3.5 [S] Le système doit stocker le nombre de points des parties jouées de chaque utilisateur en mode sprint solo.
- 3.2.3.6 [S] Le système doit afficher le score final des parties jouées en mode classique de chaque utilisateur dans la partie privée de son profil.
- 3.2.3.7 [S] Le système doit afficher le mode de jeu des parties jouées de chaque utilisateur dans la partie privée de son profil.
- 3.2.3.8 [S] Le système doit afficher le nombre de points des parties jouées de chaque utilisateur en mode sprint solo dans la partie privée de son profil.

## **3.3 Modes de jeu**

### *3.3.1 Setup*



- 3.3.1.1 [E] Le système doit permettre à un joueur d'accéder au menu principal dès leur connexion.
- 3.3.1.2 [E] Le système doit permettre à un joueur de sélectionner le mode de jeu auquel ils souhaitent accéder.
- 3.3.1.3 [E] Le système doit permettre à un joueur de sélectionner la difficulté facile.
- 3.3.1.4 [E] Le système doit permettre à un joueur de sélectionner la difficulté intermédiaire.
- 3.3.1.5 [E] Le système doit permettre à un joueur de sélectionner la difficulté difficile.
- 3.3.1.6 [E] Le système doit permettre à un joueur de consulter la liste des groupes existants.
- 3.3.1.7 [E] Le système doit permettre à un joueur de rejoindre un groupe existant.
- 3.3.1.8 [E] Le système doit permettre à un joueur de créer un nouveau groupe en entrant un nom unique de groupe.
- 3.3.1.9 [E] Le système doit permettre à un joueur de voir la liste de tous les joueurs dans les groupes existants.
- 3.3.1.10 [E] Le système doit permettre aux joueurs d'enlever un joueur virtuel d'un groupe auquel ils appartiennent.
- 3.3.1.11 [E] Le système doit permettre aux joueurs d'ajouter un joueur virtuel dans jeu classique seulement si le nombre de joueurs dans le groupe est inférieur à 4.
- 3.3.1.12 [E] Le système doit permettre à un joueur de joindre un groupe seulement si le nombre de joueurs dans le groupe est inférieur à 4.

### 3.3.2 *Salle d'attente*

- 3.3.2.1 [E] Le système doit mettre les joueurs dans une salle d'attente d'une partie quand ils rejoignent un groupe.
- 3.3.2.2 [E] Le système doit créer un canal de discussion pour les joueurs dans chaque salle d'attente.

### 3.3.3 *Start Game:*

- 3.3.1.1 [E] Le système doit permettre à une partie de commencer seulement si le nombre minimal de joueurs est atteint.
- 3.3.1.2 [E] Le système doit permettre seulement des parties de 4 joueurs en mode classique.
- 3.3.1.3 [E] Le système doit permettre seulement des parties de 1 joueur en mode sprint solo.

### 3.3.4 *Outils de dessin:*

- 3.3.4.1 [E] Le système doit avoir un outil Crayon.
- 3.3.4.2 [E] Le système doit permettre à l'utilisateur de sélectionner la couleur du crayon.
- 3.3.4.3 [E] Le système doit permettre à l'utilisateur de sélectionner l'opacité du crayon.
- 3.3.4.4 [E] Le système doit permettre à l'utilisateur de sélectionner la taille de la pointe du crayon.
- 3.3.4.5 [E] Le système doit permettre à l'utilisateur d'ajouter une grille sur le canvas.
- 3.3.4.6 [E] Le système doit permettre à l'utilisateur de retirer une grille sur le canvas.
- 3.3.4.7 [E] Le système doit permettre à l'utilisateur d'effacer des pixels avec l'outil Efface.
- 3.3.4.8 [E] Le système doit permettre à l'utilisateur de sélectionner la taille de la pointe de l'efface.
- 3.3.4.9 [E] Le système doit permettre à l'utilisateur d'annuler le dernier trait dessiné à plusieurs reprises.
- 3.3.4.10 [E] Le système doit permettre à l'utilisateur d'annuler le dernier trait effacé à plusieurs reprises.
- 3.3.4.11 [E] Le système doit permettre à l'utilisateur de refaire les actions annulées.
- 3.3.4.12 [E] Le système permet à l'utilisateur de sauvegarder tous les dessins réalisés durant la partie à la fin de celle-ci, dans son album de dessins.

### 3.3.5 *Mode de jeu Classique:*

- 3.3.5.1 [E] Le système permet à l'utilisateur de choisir le nombre de tours, entre 3, 5, 7 et 9, de la partie avant de commencer.
- 3.3.5.2 [E] Le système doit permettre à tous les utilisateurs de la séance de jeu de voir les dessins de l'équipe qui dessine.
- 3.3.5.3 [E] Pour le niveau de difficulté *Facile*, le nombre d'essais pour l'équipe qui dessine est de 10 et un temps limite de 70 secondes.
- 3.3.5.4 [E] Pour le niveau de difficulté *Intermédiaire*, le nombre d'essais pour l'équipe qui dessine est de 6 et un temps limite de 50 secondes.
- 3.3.5.5 [E] Pour le niveau de difficulté *Difficile*, le nombre d'essais pour l'équipe qui dessine est de 3 et un temps limite de 30 secondes.
- 3.3.5.6 [E] Le système doit alterner le rôle de dessinateur et de devineur entre 2 joueurs humains à chacun de leurs tours.
- 3.3.5.7 [E] Le système doit attribuer le rôle d'un dessinateur à un joueur virtuel lorsque ce dernier est présent.
- 3.3.5.8 [E] Le système doit vérifier si le mot deviné correspond au mot ou à l'expression du dessin.
- 3.3.5.9 [E] Le système doit incrémenter le score d'une équipe de 1 point lorsque cette dernière devine le mot à temps.

sans dépasser le nombre d'essais maximum.

3.3.5.10 [E] Le système doit terminer le tour de l'équipe active lorsque le joueur qui devine dépasse son nombre d'essais maximum.

3.3.5.11 [E] Le système doit terminer le tour de l'équipe active lorsque le joueur qui devine dépasse le temps alloué.

3.3.5.12 [E] Le système doit permettre un droit de réplique à l'équipe en attente quand l'équipe active échoue à deviner le mot.

3.3.5.13 [E] L'équipe possédant le droit de réplique possède un essai pour l'équipe et un temps limite de 10 secondes pour deviner.

3.3.5.14 [E] L'équipe possédant le droit de réplique incrémente son score de 1 point si elle réussit à deviner correctement.

### 3.3.6 Mode de jeu Sprint Solo:

3.3.6.1 [E] Le mode de jeu Sprint Solo permet à l'utilisateur de deviner le plus de dessins possible dans le temps alloué en ayant un nombre limite d'essais par dessin.

3.3.6.2 [E] Le système affiche le temps restant durant chaque partie.

3.3.6.3 [E] Le système affiche le nombre d'essais restant par dessin.

3.3.6.4 [E] Le système affiche le score actuel du participant.

3.3.6.5 [E] L'utilisateur gagne 1 point lorsqu'il devine le mot en respectant le nombre d'essais.

3.3.6.6 [E] Le système doit allouer 10 essais par dessin pour le niveau de difficulté *Facile*.

3.3.6.7 [E] Le système doit allouer un temps limite de 120 secondes par dessin pour le niveau de difficulté *Facile*.

3.3.6.8 [E] Le système doit allouer un temps bonus de 10 secondes par dessin pour le niveau de difficulté *Facile*.

3.3.6.9 [E] Le système doit allouer 6 essais par dessin pour le niveau de difficulté *Intermédiaire*.

3.3.6.10 [E] Le système doit allouer un temps limite de 120 secondes par dessin pour le niveau de difficulté *Intermédiaire*.

3.3.6.11 [E] Le système doit allouer un temps bonus de 5 secondes par dessin pour le niveau de difficulté *Intermédiaire*.

3.3.6.12 [E] Le système doit allouer 3 essais par dessin pour le niveau de difficulté *Difficile*.

3.3.6.13 [E] Le système doit allouer un temps limite de 120 secondes par dessin pour le niveau de difficulté *Difficile*.

3.3.6.14 [E] Le système doit allouer un temps bonus de 3 secondes par dessin pour le niveau de difficulté *Difficile*.

3.3.6.15 [E] Le système doit passer à la prochaine image sans accorder de temps bonus lorsque l'utilisateur dépasse le nombre d'essais maximum.

3.3.6.16 [E] Le système doit passer à la prochaine image sans accorder de point lorsque l'utilisateur dépasse le nombre d'essais maximum.

## 3.4 Paire mot-image

### 3.4.1 Banque de mot-image

3.4.1.1 [E] Le système permet aux utilisateurs d'ajouter un mot ou une expression à la banque de mot-image à partir du menu principal.

3.4.1.2 [E] Le système vérifie que le mot est unique avant de l'ajouter à la banque de mot-image.

3.4.1.3 [E] Le système permet à un utilisateur d'ajouter des indices lors de la création d'une paire mot-image.

3.4.1.4 [E] Le système vérifie qu'au moins un indice est ajouté à la paire mot-image.

3.4.1.5 [E] Le système permet à l'utilisateur de dessiner son mot dans une zone de dessin à l'aide des outils disponibles. Voir section 3.3.4 *Outils de dessin*.

3.4.1.6 [E] Le système permet à l'utilisateur de sélectionner le niveau de difficulté *Facile* pour sa paire mot-image.

3.4.1.7 [E] Le système permet à l'utilisateur de sélectionner le niveau de difficulté *Intermédiaire* pour sa paire mot-image.

3.4.1.8 [E] Le système permet à l'utilisateur de sélectionner le niveau de difficulté *Difficile* pour sa paire mot-image.

### 3.4.2 Manuelle II

3.4.2.1 [E] Le système permet à l'utilisateur de sélectionner un mode de dessin lors de la création du dessin, indiquant la manière de dessiner les images par le joueur virtuel.

3.4.2.2 [E] Le système doit dessiner les traits dans le même ordre que lors de la création du dessin pour le mode conventionnel.

3.4.2.3 [E] Le système doit dessiner les traits dans un ordre aléatoire différent d'une partie à une autre pour le mode aléatoire.

- 3.4.2.4 [E] Le système doit permettre de dessiner les traits dans l'ordre de leur position sur les axes cartésiens de droite à gauche pour le mode panoramique.
- 3.4.2.5 [E] Le système doit permettre de dessiner les traits dans l'ordre de leur position sur les axes cartésiens de gauche à droite pour le mode panoramique.
- 3.4.2.6 [E] Le système doit permettre de dessiner les traits dans l'ordre de leur position sur les axes cartésiens de haut à bas pour le mode panoramique.
- 3.4.2.7 [E] Le système doit permettre de dessiner les traits dans l'ordre de leur position sur les axes cartésiens de bas à haut pour le mode panoramique.
- 3.4.2.8 [E] Le système doit permettre de dessiner les traits dans l'ordre de leur distance au centre de l'image de l'intérieur vers l'extérieur pour le mode centré.
- 3.4.2.9 [E] Le système doit permettre de dessiner les traits dans l'ordre de leur distance au centre de l'image de l'extérieur vers l'intérieur pour le mode centré.

### *3.4.3 Assisté I*

- 3.4.3.1 [E] Le système doit produire un dessin à partir d'une image fournie par un utilisateur.
- 3.4.3.2 [E] Le système doit permettre à un utilisateur de fournir une image en forme de fichier bmp de son ordinateur.
- 3.4.3.3 [E] Le système doit permettre à un utilisateur de fournir une image en forme de fichier jpg de son ordinateur.
- 3.4.3.4 [E] Le système doit permettre à un utilisateur de fournir une image en forme de fichier png de son ordinateur.

### *3.4.4 Assisté II*

- 3.4.4.1 [S] Le système doit permettre l'utilisation de la banque de dessin "Quick Draw" de Google afin de proposer des mots et dessins à l'utilisateur.
- 3.4.4.2 [S] Le système doit permettre à un utilisateur de visionner une paire de mot-image une après l'autre jusqu'à ce qu'il trouve un mot et un dessin satisfaisants.

## **3.5 Personnalité des joueurs virtuels**

- 3.5.1 [E] Le système doit permettre à un joueur virtuel d'envoyer des messages de salutations prédéterminés dans le canal de discussion au début de la partie.
- 3.5.2 [E] Le système doit permettre à un joueur virtuel d'envoyer des messages d'encouragement prédéterminés dans le canal de discussion à la fin de la partie.
- 3.5.3 [E] Le système doit permettre à un joueur virtuel de fournir un indice à la demande d'indice d'un utilisateur de son équipe.
- 3.5.4 [E] Le système doit permettre à chaque joueur virtuel d'envoyer des messages personnalisés et prédéterminés selon sa personnalité.
- 3.5.5 [E] Le système doit inclure 5 joueurs virtuels avec les personnalités suivantes: gentil, arrogant, intellectuel, comique et fâché.
- 3.5.6 [S] Le système doit permettre aux joueurs virtuels d'envoyer des messages d'encouragement personnalisés selon l'historique des jeux avec son partenaire humain à la fin de la partie.
- 3.5.7 [S] Le système doit permettre aux joueurs virtuels d'envoyer des messages d'encouragement personnalisés contenant des références aux statistiques de son partenaire humain à la fin de la partie.

## **3.6 Effets visuels et sonores**

- 3.6.1 [E] Le système doit présenter une rétroaction sonore unique après une tentative de résolution réussie.
- 3.6.2 [E] Le système doit présenter une rétroaction sonore unique après une tentative de résolution manquée.
- 3.6.3 [E] Le système doit présenter un effet sonore vers l'approche de la fin de chaque tour.
- 3.6.4 [E] Le système doit présenter un effet de particules lors d'une victoire.
- 3.6.5 [E] Le système doit présenter un effet de transition lors du début d'une partie.

## **3.7 Tutoriel**

- 3.7.1 [E] Le système doit offrir un tutoriel interactif pour chaque mode de jeu.
- 3.7.2 [E] Le système doit permettre à l'utilisateur de comprendre les fonctionnalités de l'application à l'aide de bulles

d'information indiquant l'action à faire.

3.7.3 [E] Le système doit attendre que l'utilisateur exécute l'action demandée avant de passer à la prochaine étape du tutoriel.

3.7.4 [E] Le système doit indiquer à l'utilisateur sa progression dans le tutoriel à l'aide d'une barre de progression.

3.7.5 [E] Le système doit permettre à l'utilisateur de saisir une option pour consulter le tutoriel à partir du menu principal.

3.7.6 [E] Le système doit afficher le tutoriel automatiquement lorsque l'utilisateur joue à un mode de jeu pour la première fois.

3.7.7 [E] Le système doit afficher un message de confirmation lorsque le tutoriel est terminé.

3.7.8 [E] Le système doit permettre à l'utilisateur de fermer la fenêtre du message de confirmation, désactivant ainsi le tutoriel.

3.7.9 [E] Le système doit permettre à l'utilisateur de désactiver le tutoriel en tout temps à l'aide d'un bouton clairement identifié.

3.7.10 [E] Le système doit permettre à l'utilisateur d'être redirigé à la page où il se trouvait avant de commencer le tutoriel après avoir désactivé le tutoriel.

### **3.8 Langues**

3.8.1 [E] Le système doit offrir l'option de changer la langue de logiciel en français.

3.8.2 [E] Le système doit offrir l'option de changer la langue de logiciel en anglais.

3.8.3 [S] Le système doit offrir l'option de changer la langue de logiciel en allemand.

### **3.9 Couleurs/Thèmes**

3.9.1 [E] Le système doit permettre à l'utilisateur de changer le thème du logiciel entre "Light/Dark" à l'aide d'un bouton.

3.9.2 [S] Le système doit permettre à l'utilisateur de changer le thème du logiciel entre plusieurs thèmes prédéterminés (Halloween, Noël, St-Valentin)

### **3.10 Leaderboard**

3.10.1 [E] Le système doit permettre à un utilisateur de voir les classements de tous les joueurs selon leur(s) points d'expérience/niveau.

3.10.2 [E] Le système doit classer les 15 premiers utilisateurs en ordre décroissant selon leurs points d'expérience.

3.10.3 [E] Le système doit afficher le classement de l'utilisateur présentement connecté, peu importe son classement.

### **3.11 Système de points/niveau**

3.11.1 [E] Le système doit stocker le nombre de points d'expérience obtenus de chaque utilisateur.

3.11.2 [E] Le système doit afficher les points d'expérience d'un utilisateur dans la partie publique de son profil.

3.11.3 [E] Le système doit augmenter le nombre de points d'expérience total des gagnants de 50 points d'expérience pour un jeu facile, 75 points d'expérience pour un jeu moyen et 100 points d'expérience pour un jeu difficile.

3.11.4 [S] Le système doit afficher le niveau d'un utilisateur dans la partie publique de son profil.

3.11.5 [S] Le système doit afficher une barre de progression de points d'expérience d'un utilisateur dans la partie publique de son profil.

3.11.6 [S] Le système attribue à chaque utilisateur le niveau 1 à sa première connexion.

3.11.7 [S] L'utilisateur a besoin de 100 points d'expérience pour atteindre le niveau 2, de 200 points d'expérience pour le niveau 3, de 400 points d'expérience pour le niveau 5, de 1000 points d'expérience pour le niveau 6, de 2000 points d'expérience pour le niveau 7, de 3000 points d'expérience pour le niveau 8, de 4000 points d'expérience pour le niveau 9 et de 10 000 points d'expérience pour le niveau 10.

3.11.8 [S] Le système doit permettre à un utilisateur de débloquent un titre à mettre devant leur pseudo à chaque niveau.

3.11.9 [S] Le système doit permettre à un utilisateur de débloquent un choix de bordures à ajouter à leur avatar à chaque niveau.

### **3.12 Album de dessins**

3.12.1 [E] Le système doit permettre à un utilisateur de consulter son album de dessins dans la partie publique de son profil.

- 3.12.2 [E] Le système doit permettre à un album de dessin de contenir 10 images par page
- 3.12.3 [E] Le système doit permettre à un utilisateur de naviguer un album de dessin en alternant les pages
- 3.12.4 [E] Le système doit permettre à un album de dessin de contenir un nombre maximal de 5 pages
- 3.12.5 [E] Le système doit permettre à un utilisateur de supprimer les images d'un album de dessin
- 3.12.6 [E] L'utilisateur peut consulter l'album de dessins des autres utilisateurs dans la partie publique de leur profil.

### **3.13 Exigences fonctionnelles pour client léger seulement:**

#### *3.13.1 Notifications*

- 3.13.1.1 [E] Le système doit jouer un effet sonore lors de la réception de chaque nouveau message.
- 3.13.1.2 [E] Le système doit afficher un indicateur visuel lors de la réception de chaque nouveau message.
- 3.13.1.3 [E] Le système doit supprimer l'indicateur visuel lorsque tous les canaux avec de nouveaux messages ont été ouverts.

## **4. Exigences non-fonctionnelles**

### **4.1. Utilisabilité**

- 4.1.1. L'interface doit permettre à un utilisateur qui vient de finir le guide puisse se retrouver dans l'application sans rencontrer de problème.
- 4.1.2. L'utilisateur doit pouvoir compléter le tutoriel en moins de cinq minutes.
- 4.1.3. L'utilisateur doit pouvoir se créer un profil en moins de 2 minutes avec le minimum d'information nécessaire à remplir.
- 4.1.4. L'utilisateur doit pouvoir changer la langue par défaut de l'application en moins de 10 secondes.
- 4.1.5. Les boutons de l'interface doivent être visuels et réactifs.

### **4.2. Fiabilité**

- 4.2.1. Le système doit être disponible 99% du temps.
- 4.2.2. La base de données doit être disponible 99.9% du temps.

### **4.3. Performance**

- 4.3.1. L'interface utilisateur doit afficher à tous les participants ce que l'utilisateur dessine sur le canevas dans un délai de 1000 ms.
- 4.3.2. L'interface utilisateur doit afficher à tous les participants ce que l'utilisateur envoie dans le canal de discussion dans un délai de 100ms.
- 4.3.3. Le serveur doit pouvoir supporter des parties de 4 joueurs.
- 4.3.4. Le serveur doit pouvoir supporter simultanément la connexion de 10 joueurs.
- 4.3.5. La taille de l'application doit être au maximum 1Go pour le client lourd
- 4.3.6. La taille de l'application doit être au maximum 500 Mo pour le client léger.

### **4.4. Maintenabilité**

- 4.4.1. Le code final doit contenir 0 erreur ou avertissement de codage TSLINT.
- 4.4.2. Le code doit être rédigé exclusivement en anglais
- 4.4.3 Le nom des constantes doit être nommé en MAJUSCULES.
- 4.4.4 Le nom des fonctions doit être nommé en camelCase.
- 4.4.5 Le nom des classes doit être nommé en PascalCase.
- 4.4.6 Le nom des énumérations doit être nommé en PascalCase.
- 4.4.7 Le nom des interfaces doit être nommé en PascalCase.
- 4.4.8 Le nom des interfaces doit débuter par un 'I'.
- 4.4.9 Le nom des balises des composants Angular doit être nommé en kebab case.
- 4.4.10. Les fonctions doivent avoir une seule responsabilité.
- 4.4.11 Les administrateurs doivent pouvoir manipuler les données dans la base de données.

### **4.5. Contraintes de conception**

- 4.5.1. Le client lourd doit être développé avec le cadriciel Angular 8.
- 4.5.2. Le client lourd doit être développé avec le langage de programmation Typescript.
- 4.5.3. Le client lourd doit être développé sur l'éditeur de code Visual Studio Code.
- 4.5.4. Le client léger doit être développé avec le langage de programmation Kotlin.
- 4.5.5. Le client léger doit être développé sur l'environnement de développement Android Studio.
- 4.5.6. Le serveur doit être développé sur l'environnement Node.js.
- 4.5.7. Le serveur doit être développé avec le cadriciel Express.
- 4.5.8. Le serveur doit être développé avec le langage de programmation Typescript.
- 4.5.9. Le serveur doit être développé sur l'éditeur de code Visual Studio.
- 4.5.10. Le serveur doit stocker les dessins à l'aide de MongoDB.

#### **4.6. Sécurité**

4.6.1. Le serveur doit être protégé contre les injections HTML, JavaScript et SQL.

4.6.2. La base de données est seulement accessible par les administrateurs pour préserver les données des utilisateurs.

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

4.7.1 Le tutoriel doit toujours être disponible.

4.7.2 Une page FAQ doit être accessible pour les utilisateurs à partir du menu principal.

4.7.3 L'utilisateur doit être notifié par une modale lorsqu'il sauvegarde un dessin indiquant le succès ou l'échec de ce dernier.