

---

**Équipe 111 (Stack Overwork)**

---

**Projet d'évolution logiciel  
Spécifications des requis du système (SRS)**

**Version 2.4**

# Historique des révisions

Date	Version	Description	Auteur
2021-01-20	1.0	ajout des caractéristiques usagers, des interfaces utilisateurs et des interfaces matérielles.	Raphaël Lasalle
2021-01-21	1.1	ajout des premières exigences fonctionnelles	Raphaël Lasalle
2021-01-24	1.2	- ajout de requis essentiels aux sections 3.1, 3.2, 3.3, 3.7 et 4.4 - changement de certains requis marqués comme souhaitables à essentiels	Jaafar Kaoussarani
2021-01-25	1.3	revue des caractéristiques usagers, des interfaces utilisateurs et des interfaces matérielles. Ajout des caractéristiques usagers, des interfaces utilisateurs et des interfaces matérielles propres au client Léger  ajout d'exigences fonctionnelles supplémentaires pour la catégorie Modes de jeu	David Fraval Raphaël Lasalle
2021-01-25	1.4	ajout des exigences non fonctionnelles aux sections 4.1, 4.2 et 4.3	Zouhair Derouich
2021-01-26	1.5	ajout des exigences fonctionnelles pour le mode de jeu "chacun pour soi" ajout d'une section pour l'interface de dessin	Raphaël Lasalle
2021-01-27	1.6	ajout d'exigences non-fonctionnelles aux sections 4.1, 4.3, 4.5 et 4.6	Jaafar Kaoussarani
2021-01-27	1.7	ajout d'exigences fonctionnelles pour la liste d'amis, les parties privées et le vote d'expulsion renommage de la catégorie "menu" à "menu et fonctionnalités générales" pour les exigences fonctionnelles	Raphaël Lasalle
2021-01-28	1.8	Ajout de critères de fiabilité (MTBF MTTR disponibilité) Ajout de critères d'utilisabilité (créer un canal, absence de saccade)	David Fraval
2021-01-30	1.9	ajout d'exigences fonctionnelles pour le mode de de jeu modification de certaines exigences non-fonctionnelles ajout d'exigences non-fonctionnelles aux sections 4.4, 4.5 et 4.6	Jaafar Kaoussarani
2021-01-30	1.10	ajout d'exigences fonctionnelles liées à l'option des "powers ups". renommage de la catégorie "modes de jeu" à "modes et options de jeu"	Raphaël Lasalle
2021-02-01	1.11	Relecture des exigences fonctionnelle	Raphaël Lasalle Turell Théo

2021-02-02	1.12	Ajout d'exigences non-fonctionnelles à 4.4 et 4.5	Jaafar Kaoussarani
2021-02-02	1.13	Révision finale en groupe du SRS, correction et harmonisation de chaque section	Raphaël Lasalle Turell Théo Jaafar Kaoussarani David Fraval Hugo Palisson Zouhair Derouich
2021-02-17	1.14	Corrections aux sections 1 et 2	Jaafar Kaoussarani
2021-02-17	1.15	Corrections à la section 4 et révision des sections 1 et 2	Jaafar Kaoussarani
2021-02-17	1.16	Révisions de la section 3 en fonction des critiques du corrigé	David Fraval
2021-02-18	1.17	Révisions de la section 3 en fonction des critiques du corrigé	David Fraval
2021-02-19	1.18	Révisions de la section 3 en fonction des critiques du corrigé	Théo Turell
2021-02-19	1.19	Révisions des sections 3 et 4	Théo Turell Jaafar Kaoussarani Raphaël Lasalle
2021-04-06	2.0	Révision des sections 1, 2, 3.1 selon le corrigé	Théo Turell
2021-04-10	2.1	Révision des sections 3.2 à 3.17 selon le corrigé	Théo Turell
2021-04-11	2.2	Révision des sections 3.18 à Annexes selon le corrigé	Théo Turell
2021-04-13	2.3	Ajustements aux sections 1, 2 et 3	Raphaël Lasalle
2021-04-18	2.4	Vérifications et corrections par rapport à la qualité de la langue et la formulation des phrases	David Fraval Jaafar Kaoussarani Raphaël Lasalle Théo Turell

# Table des matières

<b>1. Introduction</b>	<b>6</b>
1.1. But	6
1.2. Définitions, acronymes et abréviations	6
1.3. Vue d'ensemble du document	9
<b>2. Description globale</b>	<b>9</b>
2.1. Caractéristiques des usagers	9
2.2.1. Interfaces usagers	9
2.2.2. Interfaces matérielles	9
2.2.3. Interfaces logicielles	10
2.2.4. Interfaces de communication	10
2.3. Contraintes générales	10
2.4. Hypothèses et dépendances	10
2.4.1 Hypothèse	10
2.4.2 Dépendances	10
<b>3. Exigences fonctionnelles</b>	<b>11</b>
3.1 Gestion de compte	11
3.2 Clavardage	11
3.3 Clavardage client lourd	12
3.4 Clavardage client léger	12
3.5 Profil Privé de l'utilisateur	12
3.6 Profil Public de l'utilisateur	13
3.7 Historique des parties	13
3.8 Historique de connexion	14
3.9 Interface de dessin	14
3.10 Menu et fonctionnalités générales	15
3.11 Création d'une partie	15
3.12 Menu de sélection de salles d'attente	15
3.13 Salle d'attente d'une partie	16
3.14 Salle d'attente d'une partie "Classique"	16
3.15 Salle d'attente d'une partie "Chacun pour soi"	17
3.16 Déroulement d'une partie	17
3.17 Mode de jeu "Classique"	18
3.18 Mode de jeux "Chacun pour soi"	18
3.19 Création d'une paire mot-image	19
3.20 Joueurs virtuels	20
3.21 Effets visuels et sonores	20
3.22 Tutoriel	21

3.23 Tutoriel client lourd	21
3.24 Tutoriel client léger	21
3.25 Écran récapitulatif sur le client lourd	21
3.26 Écran de fin de partie sur le client léger	22
3.27 Système relationnel	22
3.28 Expulsion d'un joueur	23
<b>4. Exigences non-fonctionnelles</b>	<b>24</b>
4.1. Utilisabilité	24
4.2. Fiabilité	24
4.3. Performance	24
4.4. Maintenabilité	25
4.5. Contraintes de conception	25
4.6. Sécurité	25
4.7. Exigences de la documentation usager en ligne et du système d'assistance	26
<b>Annexes</b>	<b>26</b>

# Spécifications des requis du système (SRS)

## 1. Introduction

### 1.1. But

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

### 1.2. Définitions, acronymes et abréviations

#### 1.2.1 Android

Il s'agit d'un système d'exploitation pour appareils mobiles. Le client léger sera développé pour ce dernier.

#### 1.2.2 Angular

Il s'agit d'un *framework* facilitant le développement des applications web. Nous l'utiliserons pour le développement du client lourd.

#### 1.2.3 Application

Prototype du jeu *Fais-moi un dessin* commandé par la firme PolyApps. L'application est disponible sur ordinateur et sur tablette, et permet à plusieurs utilisateurs de jouer en réseau à différents modes de jeux qui nécessitent de deviner un mot dessiné par un joueur virtuel ou un autre joueur humain.

#### 1.2.4 Base de données

Collection organisée et structurée de données générées et utilisées par l'application.

#### 1.2.5 Camel case

Il s'agit d'une convention de notation où les mots sont distingués en mettant la première lettre de chaque mot en majuscule. Le premier mot n'a pas de majuscule (jeSuisCamelCase).

#### 1.2.6 Client léger

Ensemble de code installé sur la version tablette du client de l'application, responsable des fonctionnalités de dessin ainsi que de l'envoi de requêtes réseaux vers le serveur.

#### 1.2.7 Client lourd

Ensemble de code installé sur la version ordinateur du client de l'application, responsable des fonctionnalités de dessins ainsi que de l'envoi de requêtes réseaux vers le serveur.

#### 1.2.8 Compte

Instance protégée avec un mot de passe et un nom d'utilisateur qui permet à un utilisateur de se connecter à "Fais-moi un dessin".

#### 1.2.9 CSS

Il s'agit des *Cascading Style Sheets*. Ce langage permet de définir l'apparence des éléments définis dans le code HTML5. Nous l'utiliserons pour le développement du client lourd.

#### 1.2.10 Electron

Il s'agit d'un environnement de développement permettant de développer et d'exécuter des applications web comme des applications de bureau. Nous l'utiliserons pour le développement du client lourd.

#### 1.2.11 Express

Il s'agit d'un *framework* facilitant le développement de serveurs. Nous l'utiliserons pour le développement du serveur.

#### **1.2.12 Firestore**

Un système de base de données NoSQL offert par le service Firebase de Google.

#### **1.2.13 Framework**

Une collection d'outils et d'interfaces de programmation.

#### **1.2.14 Heroku**

Un service qui permet d'héberger des applications web dans des environnements d'exécution indépendants similaires aux conteneurs Docker.

#### **1.2.15 HTML5**

Il s'agit de la 5ème version du *HyperText Markup Language*. Ce langage permet de définir la structure et l'emplacement des éléments visuels des applications web. Nous l'utiliserons pour le développement du client lourd.

#### **1.2.16 Interface**

Limite commune à deux systèmes, permettant des échanges entre ceux-ci.

(Définition tiré du site suivant : <https://www.larousse.fr/dictionnaires/francais/interface/43685>)

#### **1.2.17 Jasmine**

Il s'agit d'un cadriciel permettant de définir des tests unitaires pour des applications web. Nous l'utiliserons pour le développement des tests du client lourd.

#### **1.2.18 Joueur**

Ensemble des entités pouvant participer à une partie. Ils peuvent être humains (1.2.19) ou virtuels (1.2.20).

#### **1.2.19 Joueur Humain**

Utilisateur humain interagissant avec l'application lors d'une partie.

#### **1.2.20 Joueur Virtuel**

Ensemble de code permettant de simuler un utilisateur humain pendant une partie.

#### **1.2.21 Kotlin**

Il s'agit d'un langage de programmation permettant, entre autres, de développer des applications mobiles pour Android. Nous l'utiliserons pour le développement du client léger

#### **1.2.22 Git**

Il s'agit du système de gestion des versions qui sera utilisé par l'équipe de développement afin de pouvoir travailler sur plusieurs branches en parallèle.

#### **1.2.23 Jira**

Il s'agit du système de gestion de projet qui sera utilisé par l'équipe de développement pour gérer les tâches et les bugs.

#### **1.2.24 Manche**

Un manche est une étape du jeu uniquement présente pour mieux segmenter le déroulement de la partie. Une manche se conclut une fois que tous les joueurs ont fini leur tour. Une partie est composée de plusieurs manches en fonction du mode de jeu.

#### **1.2.25 Modes de Jeux**

Manières de jouer à "Fais-moi un dessin" avec des modalités différentes. L'objectif des deux modes est de remporter le plus de points pour la fin de la partie. Il y a deux modes de jeux: "classique" et "chacun pour soi".

### **1.2.26 NoSQL**

Il s'agit d'un système de gestion des bases de données qui n'utilise pas nécessairement des tableaux relationnels. Dans notre contexte, on parle de données organisées en collections et en documents, où une collection est un regroupement de documents et un document représente une entité dans la base de données.

### **1.2.27 Mode de jeu "classique"**

Se réfère à un mode de jeu durant lequel 4 joueurs forment deux équipes de deux. Pendant qu'un membre de l'équipe dessine, son partenaire doit déterminer le mot. Si l'équipe comprenant l'utilisateur qui dessine ne parvient pas à trouver le mot, l'équipe adverse a une fenêtre de temps pour deviner le mot venant d'être dessiné. La partie s'achève au bout de 4 manches.

### **1.2.28 Mode de jeu "chacun pour soi"**

Se réfère à un mode de jeu durant lequel entre 3 et 8 joueurs s'affrontent. Lorsqu'un joueur dessine, les autres joueurs doivent tenter de deviner le mot dessiné. La partie s'achève au bout de 3 manches.

### **1.2.29 Pascal case**

Il s'agit d'une convention de notation où les mots sont distingués en mettant la première lettre de chaque mot en majuscule (JeSuisPascalCase).

### **1.2.30 Profils**

Un profil est une instance liée à un utilisateur dans laquelle sont stockés: le prénom de l'utilisateur, le nom de l'utilisateur, son image de profil, ses historiques de partie et ses historiques de connexion et déconnexion.

### **1.2.31 Screaming snake case**

Il s'agit d'une convention de notation où les mots sont distingués en utilisant des lettres majuscules séparées par des traits de soulignement (SCREAMING\_SNAKE\_CASE)

### **1.2.32 Serveur**

Ensemble de code qui répond aux requêtes réseaux des clients pour leur fournir différentes fonctionnalités.

### **1.2.33 Socket.io**

Il s'agit d'une collection de méthodes permettant d'établir des connexions entre un serveur et des clients à distance et d'échanger des données entre ces derniers. Nous l'utiliserons pour le développement du serveur.

### **1.2.34 SVG**

Scalable Vector Graphics ou graphique vectoriel adaptable est un format de données pour décrire des objets vectoriels.

### **1.2.35 TCP/IP**

Un protocole de transport des données dans un réseau.

### **1.2.36 Test unitaire**

Un test dont le but est de vérifier le comportement d'une seule fonction ou méthode.

### **1.2.37 Tour**

Un tour est associé à un joueur. Lorsque c'est au tour de l'utilisateur, il pourra se mettre à dessiner pour essayer de faire deviner son dessin à d'autres personnes. Si le mot est deviné, le joueur ayant dessiné et le(s) joueur(s) ayant trouvé(s) le mot gagneront des points.

### **1.2.38 TypeScript**

Il s'agit d'un langage de programmation permettant de décrire la logique des applications web. Nous l'utiliserons pour le développement du client lourd.

### **1.2.39 Utilisateur**



Client hypothétique de la firme PolyApps qui utilise l'application depuis son ordinateur ou sa tablette pour jouer à *Fais-moi un dessin*.

#### 1.2.40 Dessinateur

Joueur humain qui est en train de dessiner un dessin car c'est à son tour de le faire.

#### 1.2.41 Devineur

Joueur humain qui est en train de deviner un mot d'un devineur car le mode de jeu courant le lui exige pour gagner des points.

### 1.3. Vue d'ensemble du document

Ce document présente d'abord une description globale de l'application, c'est-à-dire une vue d'ensemble de son intention, suivi par les caractéristiques de ses utilisateurs et les différentes interfaces permettant d'interagir avec les systèmes internes, ainsi que les diverses contraintes et hypothèses que l'on impose. Ensuite, le document énumère l'ensemble des exigences fonctionnelles, soit les comportements attendus des divers systèmes et fonctionnalités. Par la suite, on passe aux exigences non-fonctionnelles, soit les qualités et contraintes que le logiciel doit respecter.

## 2. Description globale

L'application *Fais-moi un dessin* est une application permettant de jouer à des jeux de dessins collaboratifs. Ces jeux se jouent en équipe ou individuellement, et ont comme but de faire deviner des mots ou expressions à partir de dessins créés par les joueurs en temps réel. Il est également possible d'ajouter des joueurs virtuels à une partie pour remplacer un ou plusieurs joueurs manquant. L'application possède aussi des fonctionnalités sociales permettant de se connecter, de gérer un profil, de voir des statistiques de jeu et d'interagir avec d'autres utilisateurs. Elle est disponible sur ordinateur et sur tablette Android, et se joue aussi bien avec clavier et souris qu'avec écran tactile.

### 2.1. Caractéristiques des usagers

Les utilisateurs de l'application sont les clients potentiels de PolyApps. Ces derniers sont principalement des ingénieurs logiciels. Cependant, l'application sera aussi utilisable par des clients dont les connaissances en informatique sont moins approfondies que celles d'un ingénieur. Dans tous les cas, l'application sera tout aussi bien utilisable par les utilisateurs qui préfèrent leur ordinateur que ceux qui préfèrent leur tablette.

#### 2.2.1. Interfaces usagers

L'interface du client lourd doit être une fenêtre Electron roulant sur le bureau de l'ordinateur de l'utilisateur. Cette fenêtre est minimisable et sa taille peut être ajustée. Pour créer les interfaces du client lourd, nous utiliserons un ancien projet intégrateur d'un membre du groupe fonctionnant avec Angular.

L'interface du client léger est une application Android en plein écran. L'utilisateur peut y interagir tant et aussi longtemps qu'elle est visible sur l'écran. Pour créer les interfaces, nous utiliserons les cadriciels fournis par Google et une bibliothèque nommée Glide.

L'interface du serveur doit être au minimum composée d'une ligne de commande qui permet de le démarrer. L'utilisateur n'aura jamais à voir cette ligne de commande et la connexion au serveur se fait automatiquement.

#### 2.2.2. Interfaces matérielles

L'application est destinée à l'utilisation sur ordinateur et sur tablette. Il doit être possible d'utiliser chaque fonctionnalité avec un clavier, une souris et un écran (client lourd), ainsi qu'avec un écran tactile (client léger). Des effets sonores devront être diffusés grâce aux haut-parleurs des systèmes.

Le serveur roule sur un environnement d'exécution hébergé par Heroku. Il suffit d'un écran et d'un clavier pour interagir avec la ligne de commande permettant de le déployer. L'utilisateur a besoin d'un ordinateur connecté à Internet afin de pouvoir y accéder.

### 2.2.3. Interfaces logicielles

L'interface du client lourd sera développée en Angular avec le cadriciel Electron.

L'interface du client léger sera développée en Kotlin. Elle doit être fonctionnelle sur Android.

L'interface du serveur sera développée en TypeScript en utilisant Express et Socket.io pour la communication client-serveur.

L'interface de la base de données utilise l'API Firestore. Il s'agit d'une base de données NoSQL.

### 2.2.4. Interfaces de communication

Les trois composantes du système que sont le client lourd, le serveur et le client léger vont tous communiquer avec une connexion internet. Le client lourd et le serveur pourront communiquer par wifi ou LAN avec leurs routeurs respectifs et le client léger communiquera forcément par wifi ou par données mobiles.

La communication client-serveur se fait à partir du protocole TCP/IP, qui est celui utilisé par l'API Socket.io. La connexion est établie directement entre le client et le routeur du réseau où se situe le serveur. Ce même routeur s'assure de transmettre les données de l'application vers le serveur.

## 2.3. Contraintes générales

L'application doit fournir une interface utilisateur en français.

Le code doit être rédigé en anglais.

L'application doit pouvoir rouler sur Windows 10 pour le client lourd.

L'application doit pouvoir rouler sur Android pour le client léger.

## 2.4. Hypothèses et dépendances

### 2.4.1 Hypothèse

On suppose que les utilisateurs disposent soit d'un ordinateur qui opère avec Windows 10, soit d'une tablette qui opère avec Android.

On suppose également que tous les ordinateurs des utilisateurs sont équipés de claviers et de souris ou de pavés tactiles ("touchpad").

On suppose que toutes les tablettes des utilisateurs sont le modèle Samsung Galaxy Tab A 2019.

On suppose que les utilisateurs possèdent un niveau de français écrit leur permettant de comprendre l'interface de l'application ainsi que les mots à deviner.

On suppose que les utilisateurs ont une connexion internet fiable.

On suppose que les utilisateurs utiliseront l'application uniquement lorsque le serveur est en marche.

### 2.4.2 Dépendances

Les fonctionnalités de dessin dépendent en partie d'une base de code réutilisée qui provient du projet *Poly-Dessin*.

Dans le code, les communications entre le client lourd et le serveur seront faites avec la bibliothèque Socket.io.

Pour la création d'une paire mot-image, la conversion d'un fichier téléversé en dessin se fait grâce à l'outil Potrace.

### 3. Exigences fonctionnelles

#### 3.1 Gestion de compte

- 3.1.1 Le système doit permettre à l'utilisateur de créer un compte depuis son client. (essentiel)
- 3.1.2 Le système doit exiger la saisie d'un mot de passe pour créer un compte utilisateur (essentiel)
- 3.1.3 Le système doit exiger la saisie d'un pseudonyme pour créer un compte utilisateur (essentiel)
- 3.1.4 Le système doit exiger la saisie du prénom de l'utilisateur pour créer un compte utilisateur (essentiel)
- 3.1.5 Le système doit exiger la saisie la saisie du nom de l'utilisateur pour créer un compte utilisateur (essentiel)
- 3.1.6 Le système doit s'assurer que l'utilisateur a un mot de passe d'au plus 16 caractères. (essentiel)
- 3.1.7 Le système doit s'assurer que l'utilisateur a un pseudonyme d'au plus 16 caractères. (essentiel)
- 3.1.8 Le système doit s'assurer que l'utilisateur a un prénom d'au plus 32 caractères. (essentiel)
- 3.1.9 Le système doit s'assurer que l'utilisateur a un nom de famille d'au plus 32 caractères. (essentiel)
- 3.1.10 Le système doit exécuter la connexion de l'utilisateur à partir d'un menu de connexion. (essentiel)
- 3.1.11 Le système doit connecter l'utilisateur lorsque le nom d'utilisateur et le mot de passe saisies sont corrects. (essentiel)
- 3.1.12 Le système doit vérifier qu'une seule instance d'un client (lourd ou léger) peut être connectée à un compte à la fois (essentiel).
- 3.1.13 Le système doit refuser la connexion à un utilisateur si le compte auquel il veut se connecter est déjà utilisé. (essentiel)
- 3.1.14 Le système doit demander à l'utilisateur de se connecter à son compte avant de pouvoir accéder au menu principal. (essentiel)
- 3.1.15 La page d'authentification doit en tout temps cacher les caractères entrés dans les champs de saisie de mot de passe du menu de connexion. (essentiel)
- 3.1.16 Le service d'authentification doit refuser l'accès à un compte utilisateur lorsque les identifiants sont erronés

#### 3.2 Clavardage

- 3.2.1 Le système doit permettre à l'utilisateur de créer un canal de discussion en saisissant le nom du canal. (essentiel)
- 3.2.2 Le système doit permettre à l'utilisateur de rejoindre un canal de discussion. (essentiel)
- 3.2.3 Le système doit permettre à l'utilisateur d'être dans plusieurs canaux de communication à la fois. (essentiel)
- 3.2.4 Le système doit permettre à l'utilisateur de quitter un canal de communication. (essentiel)
- 3.2.5 Le système doit supprimer un canal quand tous les utilisateurs l'ont quitté. (essentiel)
- 3.2.6 Le système doit permettre à l'utilisateur de consulter l'historique de clavardage d'un canal auquel il se joint.

(essentiel)

**3.2.7** Le système doit connecter automatiquement l'utilisateur au canal global par défaut suite à sa première connexion. (essentiel)

**3.2.8** Le système doit se souvenir des canaux auxquels un utilisateur participe suite à sa reconnexion. (essentiel)

**3.2.9** Le système doit permettre à l'utilisateur connecté de pouvoir clavier dans tout canal dont il est membre à tout moment. (essentiel)

**3.2.10** Le système doit créer un canal pour chaque partie par défaut accessible uniquement par les utilisateurs qui participent à la partie. (essentiel)

**3.2.11** Le système doit permettre à l'utilisateur de pouvoir clavier avec les autres utilisateurs présents dans une salle d'attente en attendant que son groupe de jeu soit complet. (essentiel)

**3.2.12** Le système doit limiter les messages de l'utilisateur à 200 caractères maximum. (essentiel)

### **3.3 Clavardage client lourd**

**3.3.1** Le système doit permettre à l'utilisateur de clavier dans un mode fenêtré. (essentiel)

**3.3.2** Le système doit permettre à l'utilisateur de clavier dans un mode intégré. (essentiel)

**3.3.3** Le système doit permettre à l'utilisateur d'alterner entre le mode fenêtré et le mode intégré. (essentiel)

**3.3.4** Le système doit faire réapparaître la fenêtre de clavardage sur l'interface en mode intégré si elle est fermée. (essentiel)

**3.3.5** Le système doit montrer un seul mode d'affichage à la fois pour le clavardage. (essentiel)

### **3.4 Clavardage client léger**

**3.4.1** Le système doit permettre à l'utilisateur d'accéder à l'interface de clavardage depuis n'importe quelle vue de l'application. (essentiel)

**3.4.2** Le système doit afficher un indicateur visuel à l'utilisateur lorsqu'il reçoit un nouveau message. (essentiel)

**3.4.3** Le système doit jouer un indicateur sonore à l'utilisateur lorsqu'il reçoit un nouveau message. (essentiel)

**3.4.4** Le système doit faire disparaître l'indicateur visuel pour les messages lus lorsque tous les canaux contenant des nouveaux messages ont été ouverts. (essentiel)

### **3.5 Profil Privé de l'utilisateur**

**3.5.1** Le système doit permettre à chaque profil de posséder une partie privée qui est invisible aux autres utilisateurs. (essentiel)

**3.5.2** Le système doit afficher le nom de l'utilisateur dans la partie privée du profil. (essentiel)

**3.5.3** Le système doit afficher le prénom de l'utilisateur dans la partie privée du profil. (essentiel)

**3.5.4** Le système doit afficher le mot de passe de l'utilisateur dans la partie privée du profil. (essentiel)

**3.5.5** Le système doit laisser voir la partie public du compte de l'utilisateur depuis la partie privée. (essentiel)

**3.5.6** Le système doit permettre à l'utilisateur d'accéder à son profil quand il est hors d'une partie. (essentiel)

**3.5.7** Le système doit permettre à l'utilisateur de modifier son avatar (essentiel)

**3.5.8** Le système doit proposer à l'utilisateur plusieurs images pour la sélection de l'avatar dans le client lourd. (souhaitable)

**3.5.9** Le système doit proposer à l'utilisateur de dessiner son propre avatar grâce à l'interface de dessin lorsqu'il choisi son avatar dans le client lourd (voir exigences de la section [3.9](#)) (souhaitable)

**3.5.10** Le système doit montrer les statistiques utilisateur dans la partie privée du profil utilisateur (essentiel)

**3.5.11** Le système doit afficher un historique des partie dans la partie privée du profil utilisateur (voir exigences de la section [3.7](#)) (essentiel)

**3.5.12** Le système doit afficher un historique de connexion dans la partie privée du profil utilisateur (voir exigences de la section [3.8](#)) (essentiel)

### **3.6 Profil Public de l'utilisateur**

**3.6.1** Le système doit permettre à chaque profil de posséder une partie publique qui est visible par les autres utilisateurs. (essentiel)

**3.6.2** Le système doit afficher le pseudonyme de l'utilisateur dans la partie publique du profil. (essentiel)

**3.6.3** Le système doit afficher l'avatar de l'utilisateur dans la partie publique du profil (essentiel)

### **3.7 Historique des parties**

**3.7.1** Le système doit afficher chaque partie figurant dans l'historique d'un profil utilisateur avec la date et l'heure à laquelle elle a commencé. (essentiel)

**3.7.2** Le système doit afficher chaque partie figurant dans l'historique d'un profil utilisateur avec la date et l'heure à laquelle elle a terminé. (essentiel)

**3.7.3** Le système doit afficher chaque partie figurant dans l'historique d'un profil utilisateur avec le nom de chaque joueur y ayant participé. (essentiel)

**3.7.4** Le système doit afficher chaque partie figurant dans l'historique d'un profil utilisateur avec un indicateur disant si la partie a été gagnée ou perdue. (essentiel)

**3.7.5** Le système doit afficher chaque partie figurant dans l'historique d'un profil utilisateur avec le mode de jeu de la partie. (essentiel)

**3.7.6** Le système doit afficher chaque partie figurant dans l'historique d'un profil utilisateur avec le score final. (essentiel)

**3.7.7** Le système doit afficher chaque partie figurant dans l'historique d'un profil utilisateur avec l'hôte de la partie. (essentiel)

### 3.8 Historique de connexion

3.8.1 Le système doit afficher les dates de connexion dans l'historique de connexion (essentiel)

3.8.2 Le système doit afficher les dates de déconnexion dans l'historique de connexion (essentiel)

### 3.9 Interface de dessin

3.9.1 Le système doit montrer une interface de dessin à l'utilisateur lorsqu'il participe à une partie. (essentiel)

3.9.2 Le système doit rendre l'interface de dessin utilisable par l'utilisateur uniquement lorsque c'est à son tour de dessiner. (essentiel)

3.9.3 Le système doit afficher en temps réel les traits du joueur qui dessine à tous les autres joueurs dans la partie. (essentiel)

3.9.4 Le système doit présenter une interface de dessin à l'utilisateur qui comporte un outil crayon qui permet de tracer des traits sur la surface de dessin. (essentiel)

3.9.5 Le système doit présenter une interface de dessin à l'utilisateur qui lui permet de modifier l'épaisseur du trait du crayon. (essentiel)

3.9.6 Le système doit présenter une interface de dessin à l'utilisateur qui lui permet de modifier la couleur du trait du crayon. (essentiel).

3.9.7 Le système doit présenter une interface de dessin à l'utilisateur qui lui permet de modifier la transparence du trait du crayon. (essentiel).

3.9.8 Le système doit présenter une interface de dessin à l'utilisateur qui comporte un outil efface. (essentiel)

3.9.9 Le système doit présenter une interface de dessin à l'utilisateur qui lui permet de modifier l'épaisseur de l'efface sur le client lourd. (essentiel)

3.9.10 Le système doit présenter une interface de dessin à l'utilisateur qui permet d'afficher une grille quadrillée sur la surface de dessin. (essentiel)

3.9.11 Le système doit présenter une interface de dessin à l'utilisateur qui lui permet de modifier la taille des carreaux de la grille sur le client lourd. (essentiel)

3.9.12 Le système doit présenter une interface de dessin à l'utilisateur qui lui permet de modifier la transparence des carreaux de la grille sur le client lourd. (essentiel)

3.9.13 Le système doit présenter une interface de dessin à l'utilisateur qui comporte une fonctionnalité "annuler" lui permettant d'annuler les dernières modifications dans un ordre chronologique inverse. (essentiel)

3.9.14 Le système doit présenter une interface de dessin à l'utilisateur qui comporte une fonctionnalité "refaire" lui permettant de réafficher les modifications annulées dans un ordre chronologique inverse. (essentiel)

3.9.15 Le système doit présenter une interface de dessin à l'utilisateur qui comporte une fonctionnalité lui permettant de changer la couleur de fond de la surface de dessin. (essentiel)

3.9.16 Le système doit présenter une interface de dessin à l'utilisateur qui comporte une fonctionnalité lui permettant de sélectionner les 10 dernières couleurs qu'il a utilisé. (essentiel)

3.9.17 Le système doit présenter une interface de dessin à l'utilisateur qui lui permet de spécifier des valeurs RGB

lors de la sélection d'une couleur sur le client lourd. (essentiel)

**3.9.18** Le système doit présenter une interface de dessin à l'utilisateur qui lui permet de spécifier une valeur hexadécimale lors de la sélection d'une couleur sur le client lourd. (essentiel)

**3.9.19** Le système doit faire en sorte que chaque fonctionnalité de dessin disponible sur le client léger doit également être disponible sur le client lourd et vice-versa. (essentiel)

**3.9.20** Le système doit rendre disponible l'interface de dessin lors de la création de l'avatar sur le client lourd. (souhaitable)

### **3.10 Menu et fonctionnalités générales**

**3.10.1** Le système doit donner l'accès au menu principal à l'utilisateur après qu'il se soit connecté à son profil. (essentiel)

**3.10.2** Le système doit permettre à l'utilisateur d'accéder à l'interface de création de partie depuis le menu principal (essentiel)

**3.10.3** Le système doit permettre à l'utilisateur d'accéder à l'interface de sélection de salles d'attente depuis le menu principal (essentiel)

**3.10.4** Le système doit permettre à l'utilisateur d'accéder à l'interface de dessin pour créer une paire mot-image depuis le menu principal (essentiel)

**3.10.5** Le système doit permettre à l'utilisateur d'accéder à son menu de profil à partir du menu principal. (essentiel)

**3.10.6** Le système doit permettre à l'utilisateur d'accéder à l'interface sociale à partir du menu principal. (souhaitable)

### **3.11 Création d'une partie**

**3.11.1** Le système doit permettre à l'utilisateur de sélectionner le mode de jeu auquel il veut jouer depuis le menu de création de partie. (essentiel)

**3.11.2** Le système doit permettre à l'utilisateur de sélectionner la difficulté des paires mot-images depuis le menu de création de partie. (essentiel)

**3.11.3** Le système doit permettre à l'utilisateur de créer une partie privée qui est seulement accessible par mot de passe depuis le menu de création de partie. (souhaitable)

**3.11.4** Le système doit décrire la difficulté des paires mot-image qui seront utilisées dans une partie comme étant "facile", "normale" ou "difficile"(essentiel)

**3.11.5** Le système doit pré-sélectionner une difficulté facile et le mode de jeu "classique" lors de l'ouverture de menu de la création de partie. (essentiel)

### **3.12 Menu de sélection de salles d'attente**

**3.12.1** Le système doit montrer à l'utilisateur toutes les salles d'attente disponibles lorsqu'il arrive sur le menu de sélection de salles d'attente. (essentiel)

**3.12.2** Le système doit donner l'option à l'utilisateur de filtrer les salles d'attente en fonction de leurs difficultés. (essentiel)

**3.12.3** Le système doit donner l'option à l'utilisateur de filtrer les salles d'attente en fonction de leurs modes de jeu. (essentiel)

**3.12.4** Le système doit donner l'option à l'utilisateur de filtrer les salles d'attente en fonction de la présence de ses amis. (souhaitable)

**3.12.5** Le système doit donner l'option à l'utilisateur de filtrer les salles d'attente en fonction de si leur capacité maximale est atteinte. (souhaitable)

**3.12.6** Le système doit donner l'option à l'utilisateur de filtrer les salles d'attente en fonction du nom de leur hôte. (souhaitable)

**3.12.7** Le système doit donner l'option à l'utilisateur de filtrer les salles d'attente en fonction de si elles sont publiques. (souhaitable)

**3.12.8** Le système doit montrer à l'utilisateur toutes les parties en attente qui correspondent à ses critères de recherche. (essentiel)

### **3.13 Salle d'attente d'une partie**

**3.13.1** Le système doit assigner le rôle d'hôte à l'utilisateur ayant créé la partie.

**3.13.2** Le système doit permettre à l'utilisateur d'être placé dans une salle d'attente de la partie qu'il rejoint. (essentiel)

**3.13.3** Le système doit permettre au joueur hôte d'ajouter un joueur virtuel à une salle d'attente de partie en mode "classique". (essentiel)

**3.13.4** Le système doit permettre au joueur hôte d'ajouter autant de joueurs virtuels qu'il y a d'emplacements disponibles (essentiel)

**3.13.5** Le système doit retirer les utilisateurs de la salle d'attente si l'hôte quitte la salle d'attente sans aller dans une partie. (essentiel)

**3.13.6** Le système doit permettre à l'utilisateur de démarrer une partie qu'il a créé si le nombre de joueurs minimum est atteint (voir sections **3.14** et **3.15** pour plus de détails). (essentiel)

### **3.14 Salle d'attente d'une partie "Classique"**

**3.14.1** Le système doit laisser l'utilisateur hôte démarrer une partie en mode "classique" uniquement si chaque équipe comprend deux joueurs dont au moins un est humain. (essentiel)

**3.14.2** Le système doit séparer les joueurs en deux équipes pour une partie en mode "classique". (essentiel)

**3.14.3** Le système doit laisser l'utilisateur hôte d'une partie en mode "classique" changer les joueurs d'équipe. (essentiel)

**3.14.4** Le système doit laisser l'utilisateur hôte d'une partie en mode "classique" ajouter un joueur virtuel à une équipe qui comporte moins de deux joueurs. (essentiel)

**3.14.5** Le système doit laisser l'utilisateur hôte d'une partie en mode "classique" ajouter un joueur virtuel à une équipe seulement si cette équipe est dépourvue de joueur virtuel. (essentiel)



**3.14.6** Le système doit proposer à l'utilisateur hôte d'une partie en mode "classique" des joueurs virtuels à ajouter dont le nom est différent du joueur virtuel déjà présent dans la partie. (essentiel)

**3.14.7** L'application doit s'assurer qu'au plus 4 joueurs peuvent participer à une partie en mode "classique". (essentiel)

### **3.15 Salle d'attente d'une partie "Chacun pour soi"**

**3.15.1** Le système doit permettre à une partie en mode "chacun pour soi" de comporter un minimum de 3 joueurs humains. (souhaitable)

**3.15.2** Le système doit permettre à une partie en mode "chacun pour soi" de comporter un maximum de 8 joueurs humains. (souhaitable)

**3.15.3** Le système doit s'assurer qu'une partie en mode "chacun pour soi" ne comporte que des joueurs humains. (souhaitable)

### **3.16 Déroulement d'une partie**

**3.16.1** Le système doit s'assurer que tous les clients des joueurs humains ont envoyé une requête disant qu'ils sont prêts à commencer la partie. (essentiel)

**3.16.2** Le système doit permettre à l'utilisateur d'essayer de deviner un mot sous forme de saisie texte lorsque c'est à son tour de deviner. (essentiel)

**3.16.3** Le système doit accorder un nombre de points défini par le mode de jeu (voir sections **3.17** et **3.18**) à un utilisateur qui tente de deviner un mot si sa réponse correspond au mot choisi par l'utilisateur qui dessine. (souhaitable)

**3.16.4** Le système doit accorder à l'utilisateur 78 secondes pour tenter de deviner un mot ou une expression dans une partie avec une difficulté "facile". (essentiel)

**3.16.5** Le système doit accorder à l'utilisateur 60 secondes pour tenter de deviner un mot ou une expression dans une partie avec une difficulté "normale". (essentiel)

**3.16.6** Le système doit accorder à l'utilisateur 42 secondes pour tenter de deviner un mot ou une expression dans une partie avec une difficulté "difficile". (essentiel)

**3.16.7** Le système doit donner à l'utilisateur des mots d'une paire mot-image en fonction de la difficulté de la partie. (souhaitable)

**3.16.8** Le système doit donner à l'utilisateur 3 suggestions de mots à dessiner au début de son tour. (souhaitable)

**3.16.9** Le système doit donner 10 secondes à l'utilisateur pour choisir un des trois mots proposés. (souhaitable)

**3.16.10** Le système doit sélectionner un mot pour l'utilisateur en l'absence de choix de mot par l'utilisateur au bout de l'intervalle donné du **3.16.8**. (souhaitable)

**3.16.11** Le système doit afficher une liste de tous les joueurs présents dans la partie en cours. (souhaitable)

**3.16.12** Le système doit permettre à l'utilisateur d'émettre des réactions sous la forme d'émoticônes durant une partie. (souhaitable)

**3.16.13** Le système doit faire apparaître les réactions émises par un utilisateur durant une partie vis-à-vis de son nom

dans la liste des utilisateurs présents. (souhaitable)

**3.16.14** Le système doit éviter de répéter les mots d'une paire mot-image ayant déjà été sélectionnés durant la partie. (souhaitable)

**3.16.15** Le système doit réinitialiser la liste des mots déjà utilisé d'une parties si la liste de mots des paires mot-image globale est entièrement utilisée (souhaitable)

**3.16.16** Le système doit permettre à l'utilisateur de faire une demande d'indice lorsque c'est à son tour de deviner. (essentiel)

**3.16.17** Le système doit afficher un nouvel indice associé dessin en cours de réalisation à chaque fois que l'utilisateur en fait la demande. (essentiel)

**3.16.18** Le système doit octroyer à un utilisateur autant de demandes d'indice qu'il y a d'indices associés à l'image de la paire mot-image en train d'être dessinée. (essentiel)

### **3.17 Mode de jeu "Classique"**

**3.17.1** Le système doit faire jouer les tours d'une manche dans un ordre déterminé en fonction du nombre de joueurs virtuels (voir [Tableau 1 à Tableau 3](#)). (essentiel)

**3.17.2** Le système doit terminer une partie à la fin de la quatrième manche en mode "classique". (essentiel)

**3.17.3** Le système doit laisser un utilisateur tenter de deviner le mot du dessin fait par un autre joueur dans une partie en mode "classique" uniquement s'ils appartiennent à la même équipe. (essentiel)

**3.17.4** Le système doit permettre à chaque joueur virtuel dans une partie en mode "classique" de dessiner 2 images durant le tour de son équipe. (essentiel)

**3.17.5** Le système doit donner 1 point à l'équipe d'un utilisateur lorsqu'il réussit à deviner le mot dessiné par son coéquipier dans une partie en mode "classique". (essentiel)

**3.17.6** Le système doit donner l'opportunité aux utilisateurs dans une partie en mode "classique" de tenter de deviner le mot de l'équipe adverse si cette dernière est incapable de le deviner dans le temps imparti. (essentiel)

**3.17.7** Le système doit offrir à un utilisateur qui tente de deviner le dessin de l'autre équipe un temps imparti égal à la moitié de celui de l'autre équipe dans une partie en mode "classique". (essentiel)

**3.17.8** Le système doit offrir un point à l'équipe d'un utilisateur qui réussit à deviner un mot de l'équipe adverse. (essentiel)

### **3.18 Mode de jeux "Chacun pour soi"**

**3.18.1** Le système doit mettre fin à une partie en mode "chacun pour soi" après 3 manches. (souhaitable)

**3.18.2** Le système doit laisser un utilisateur dessiner exactement une fois lors d'une manche dans une partie en mode "chacun pour soi". (souhaitable)

**3.18.3** Le système doit laisser tous les utilisateurs autres que le dessinateur d'être des devineurs dans une partie en mode "chacun pour soi". (souhaitable)

**3.18.4** Le système doit accorder au dessinateur un nombre de points entre 1 et 5 qui est proportionnel au nombre de devineurs qui ont deviné le mot associé à son dessin dans une partie en mode "chacun pour soi". (souhaitable)

**3.18.5** Le système doit accorder à un utilisateur qui devine correctement un mot ou expression dans une partie en

mode “chacun pour soi” un nombre de points entre 1 et 5 qui est proportionnel au temps qu’il reste sur l’horloge du tour.

### **3.19 Création d’une paire mot-image**

**3.19.1** Le système doit rendre accessible la fonctionnalité de création d’une paire mot-image uniquement depuis le client lourd. (essentiel)

**3.19.2** Le système doit permettre à l’utilisateur de saisir le mot ou l’expression associés à une paire mot-image lors de sa création. (essentiel)

**3.19.3** Le système doit permettre à l’utilisateur d’ajouter autant d’indices associés à une paire mot-image qu’il le souhaite lors de sa création. (essentiel)

**3.19.4** Le système doit donner l’option à l’utilisateur de dessiner l’image d’une paire mot-image lors de sa création avec l’interface de dessin. (essentiel)

**3.19.5** Le système doit donner l’option à l’utilisateur de fournir un fichier bmp, png ou jpg lors de la création d’une paire mot-image. (essentiel)

**3.19.6** Le système doit s’assurer que quand l’utilisateur soumet un fichier lors de la création d’une paire mot-image, l’image sera convertie en svg avec l’outil Potrace. (essentiel)

**3.18.7** Le système doit permettre à l’utilisateur de pouvoir modifier la taille des mouchetures lors de la conversion du fichier en image svg. (essentiel)

**3.19.8** Le système doit permettre à l’utilisateur de pouvoir modifier l’optimisation de la courbe lors de la conversion du fichier en image svg. (essentiel)

**3.19.9** Le système doit permettre à l’utilisateur de pouvoir modifier le seuil de tolérance colorimétrique lors de la conversion du fichier en image svg. (essentiel)

**3.19.10** Le système doit permettre à un utilisateur de dessiner avec l’interface de dessin habituelle sur un dessin qui a été généré à partir d’un fichier lors de la création d’une paire mot-image. (souhaitable)

**3.19.11** Le système doit permettre à l’utilisateur de définir un niveau de difficulté associé à une paire mot-image lors de sa création. (essentiel)

**3.19.12** Le système doit permettre à l’utilisateur de définir un mode de dessin associé à une paire mot-image lors de sa création. (essentiel)

**3.19.13** Le système doit permettre à l’utilisateur de pouvoir choisir le mode conventionnel pour une paire mot-image lors de sa création. (essentiel)

**3.19.14** Le système doit permettre à l’utilisateur de pouvoir choisir le mode aléatoire pour une paire mot-image lors de sa création. (essentiel)

**3.19.15** Le système doit permettre à l’utilisateur de pouvoir choisir le mode panoramique pour une paire mot-image lors de sa création. (essentiel)

**3.19.16** Le système doit permettre à l’utilisateur de pouvoir choisir la direction droite à gauche pour une paire mot-image en mode panoramique. (essentiel)

**3.19.17** Le système doit permettre à l’utilisateur de pouvoir choisir la direction gauche à droite pour une paire mot-image en mode panoramique. (essentiel)

**3.19.18** Le système doit permettre à l'utilisateur de pouvoir choisir la direction haut à bas pour une paire mot-image en mode panoramique. (essentiel)

**3.19.19** Le système doit permettre à l'utilisateur de pouvoir choisir la direction bas à haut pour une paire mot-image en mode panoramique. (essentiel)

**3.19.20** Le système doit permettre à l'utilisateur de pouvoir choisir le mode centré pour une paire mot-image lors de sa création. (essentiel)

**3.19.21** Le système doit permettre à l'utilisateur de pouvoir choisir la direction intérieur vers extérieur pour une paire mot-image en mode centré. (essentiel)

**3.19.22** Le système doit permettre à l'utilisateur de pouvoir choisir la direction extérieur vers intérieur pour une paire mot-image en mode centré. (essentiel)

### **3.20 Joueurs virtuels**

**3.20.1** Le système doit permettre aux joueurs virtuels d'envoyer des messages par clavardage. (essentiel)

**3.20.2** Le système doit envoyer des messages par clavardage de la part des joueurs virtuels au début d'une partie. (essentiel)

**3.20.3** Le système doit envoyer des messages par clavardage de la part des joueurs virtuels à la fin de chaque tour. (essentiel)

**3.20.4** Le système doit donner une personnalité distincte à chaque joueur virtuel qu'il garde tout au long d'une partie. (essentiel)

**3.20.5** Le système doit permettre à des joueurs virtuels d'avoir une personnalité "neutre". (essentiel)

**3.20.6** Le système doit permettre à des joueurs virtuels d'avoir une personnalité "méchante". (essentiel)

**3.20.7** Le système doit permettre à des joueurs virtuels d'avoir une personnalité "gentille". (essentiel)

**3.20.8** Le système doit dessiner des images appartenant aux paires mot-images à travers les joueurs virtuels. (essentiel)

**3.20.9** Le système doit dessiner trait par trait les dessins tel que défini par l'ordre de traçage de la paire mot-image. (essentiel)

**3.20.10** Le système doit ajuster la vitesse de traçage des images dessiné par les joueurs virtuels pour qu'elle soit inversement proportionnelle à la difficulté de l'image. (essentiel)

### **3.21 Effets visuels et sonores**

**3.21.1** Le système doit jouer un effet sonore distinct après chaque tentative pour deviner un mot réussie par l'utilisateur. (essentiel)

**3.21.2** Le système doit jouer un effet sonore distinct après chaque tentative pour deviner un mot échouée par l'utilisateur. (essentiel)

**3.21.3** Le système doit jouer un effet sonore distinct à chaque seconde dans les dix secondes qui précèdent la fin du temps alloué à un joueur pour deviner un mot. (essentiel)

**3.21.4** Le système doit jouer un effet sonore distinct lorsque le temps alloué à un joueur pour deviner un mot est

épuisé. (essentiel)

**3.21.5** Le système doit jouer un effet sonore distinct lorsque l'utilisateur ou son équipe gagne une partie. (souhaitable)

**3.21.6** Le système doit jouer un effet sonore distinct lorsque l'utilisateur termine le tutoriel. (essentiel)

**3.21.7** Le système doit jouer un effet sonore pour une durée maximale de 5s. (essentiel)

**3.21.8** Le système doit montrer un effet visuel distinct lorsque l'utilisateur ou son équipe gagne une partie. (souhaitable)

### **3.22 Tutoriel**

**3.22.1** Le système doit proposer un tutoriel à l'utilisateur lorsqu'il se connecte à l'application pour la première fois. (essentiel)

**3.22.2** Le système doit proposer à l'utilisateur un tutoriel qui comporte une section expliquant le fonctionnement de l'outil crayon (incluant les couleurs). (essentiel)

**3.22.3** Le système doit proposer à l'utilisateur un tutoriel qui comporte une section expliquant le fonctionnement de l'outil efface. (essentiel)

**3.22.4** Le système doit proposer à l'utilisateur un tutoriel qui comporte une section expliquant le fonctionnement de l'outil annuler-refaire. (essentiel)

**3.22.5** Le système doit proposer à l'utilisateur un tutoriel qui comporte une section expliquant le fonctionnement de la grille. (essentiel)

**3.22.6** Le système doit proposer à l'utilisateur un tutoriel qui comporte une section expliquant le déroulement d'une partie. (essentiel)

**3.22.7** Le système doit permettre en tout temps à l'utilisateur de consulter le tutoriel depuis le menu principal de l'application. (essentiel)

**3.22.8** Le système doit permettre en tout temps à l'utilisateur de quitter le tutoriel. (essentiel)

### **3.23 Tutoriel client lourd**

**3.23.1** Le système doit proposer un tutoriel sur le client lourd qui nécessite une interaction de l'utilisateur avec l'interface de jeu pour accéder à l'étape suivante. (essentiel)

### **3.24 Tutoriel client léger**

**3.24.1** Le système doit proposer un tutoriel sur le client léger qui nécessite que l'utilisateur appuie sur un bouton "suivant" pour accéder à l'étape suivante. (essentiel)

**3.24.2** Le système doit offrir un bouton «retour» qui permet de retourner à l'étape précédente du tutoriel. (essentiel)

### **3.25 Écran récapitulatif sur le client lourd**

**3.25.1** Le système doit permettre à l'utilisateur d'avoir accès à l'écran récapitulatif à la fin d'une partie. (souhaitable)

**3.25.2** Le système doit afficher tous les utilisateurs ayant participé à la partie sur l'écran récapitulatif. (souhaitable)

**3.25.3** Le système doit créer un écran récapitulatif avec un nombre de lignes correspondant au nombre de manches jouées. (souhaitable)

**3.25.4** Le système doit créer un écran récapitulatif avec un nombre de colonnes correspondant aux nombres d'utilisateurs ayant participé la partie. (souhaitable)

**3.25.5** Le système doit remplir chaque case du tableau avec l'image dessinée par le joueur durant la manche correspondant à la ligne. (souhaitable)

**3.25.6** Le système doit remplir chaque case du tableau avec le mot donné aux utilisateurs durant la manche correspondant à la ligne. (souhaitable)

**3.25.7** Le système doit afficher la liste des utilisateurs ayant participé à la partie à côté du tableau. (souhaitable)

**3.25.8** Le système doit permettre à l'utilisateur de sélectionner un utilisateur dans la liste pour mettre en évidence les images que cet utilisateur a dessiné. (souhaitable)

**3.25.9** Le système doit permettre à l'utilisateur de sélectionner un utilisateur dans la liste pour mettre en évidence les images que cet utilisateur a deviné. (souhaitable)

### **3.26 Écran de fin de partie sur le client léger**

**3.26.1** Le système doit indiquer à l'utilisateur s'il a gagné ou perdu une partie à la fin de cette partie.

### **3.27 Système relationnel**

**3.27.1** Le système doit pouvoir recevoir une demande d'ami d'un utilisateur en tout temps. (souhaitable)

**3.27.2** Le système doit pouvoir envoyer une demande d'ami d'un utilisateur à un autre en tout temps. (souhaitable)

**3.27.3** Le système doit avoir une interface pour le système relationnel (souhaitable)

**3.27.4** Le système doit permettre à un utilisateur de voir la partie publique du profil des autres utilisateurs dans sa liste d'amis. (souhaitable)

**3.27.5** Le système doit permettre à un utilisateur d'envoyer une demande d'ami à l'aide d'un pseudonyme. (souhaitable)

**3.27.6** Le système doit permettre à la liste d'amis d'afficher le statut "En ligne" des amis qui sont connectés et actifs. (souhaitable)

**3.27.7** Le système doit permettre à la liste d'amis d'afficher le statut "En partie" des amis qui participent à une partie. (souhaitable)

**3.27.8** Le système doit permettre à la liste d'amis d'afficher le statut "En attente" des amis qui sont dans une salle d'attente. (souhaitable)

**3.27.9** Le système doit permettre à la liste d'amis d'afficher le statut "Inactif" des amis connectés à un client lourd qui ne bougent pas leur souris depuis 2 minutes. (souhaitable)

**3.27.10** Le système doit permettre à la liste d'amis d'afficher le statut "Hors Ligne" des amis qui ne sont pas connectés à l'application. (souhaitable)

**3.27.11** Le système doit permettre à la liste d'amis d'afficher le statut "Ne pas déranger" des amis qui ont sélectionné l'option correspondante. (souhaitable)

**3.27.12** Le système doit permettre à l'utilisateur de retirer des amis de sa liste d'amis. (souhaitable)

**3.27.13** Le système doit permettre à l'utilisateur d'avoir accès à une interface du système relationnel dédiée à la gestion des demandes d'amis. (souhaitable)

**3.27.14** Le système doit permettre à l'utilisateur de pouvoir accepter une demande d'amis qui lui a été envoyée. (souhaitable)

**3.27.15** Le système doit permettre à l'utilisateur de refuser une demande d'amis qui lui a été envoyée. (souhaitable)

**3.27.16** Le système doit faire apparaître dans la liste d'amis de l'utilisateur les autres profils utilisateurs ayant accepté une invitation d'amis par un utilisateur. (souhaitable)

**3.27.17** Le système doit permettre à l'utilisateur de recevoir les demandes d'amis qui lui ont été envoyées pendant qu'il était hors ligne lors de sa connexion. (souhaitable)

### **3.28 Expulsion d'une joueur**

**3.28.1** Le système doit permettre à un utilisateur d'initier un vote pour expulser un utilisateur qui se trouve dans la même partie que lui.

**3.28.2** Le système doit permettre à l'utilisateur de pouvoir voter "pour" sur un vote pour expulser un autre utilisateur. (souhaitable)

**3.28.3** Le système doit permettre à l'utilisateur de pouvoir voter "contre" sur un vote pour expulser un autre utilisateur. (souhaitable)

**3.28.4** Le système doit permettre à l'utilisateur de pouvoir voter uniquement sur des votes ciblant un autre utilisateur que lui-même. (souhaitable)

**3.28.5** Le système doit permettre à l'utilisateur d'initier un vote d'expulsion uniquement sur un utilisateur autre que lui-même. (souhaitable)

**3.28.7** Le système doit expulser l'utilisateur ciblé par un vote d'expulsion si ce dernier a recueilli une majorité de votes "pour". (souhaitable)

**3.28.8** Le système doit garder l'utilisateur ciblé par un vote d'expulsion dans sa partie ou salle d'attente si le vote a recueilli une majorité de "contre". (souhaitable)

**3.28.8** Le système doit garder un utilisateur dans la partie s'il y a égalité entre le nombre de votes pour son expulsion et le nombre de votes contre son expulsion. (souhaitable)

**3.28.9** Le système doit permettre à l'utilisateur d'initier un vote pour expulser un autre utilisateur si et seulement si la partie en cours est dépourvue de joueurs virtuels. (souhaitable)

**3.28.10** Le système doit permettre à l'utilisateur d'initier un vote pour expulser un autre utilisateur seulement si après cette expulsion le nombre de joueurs sera supérieur au nombre minimum de joueurs pour le mode de jeu actuel. (souhaitable)

## 4. Exigences non-fonctionnelles

L'application doit s'assurer qu'un utilisateur ne peut pas accéder aux fonctionnalités s'il n'est pas connecté à son compte

L'application doit se limiter à 5 parties simultanées.

L'application doit nécessiter moins de 2 Gigaoctets (Go) de mémoire vive pour pouvoir fonctionner.

L'application doit fonctionner à une résolution minimale de 1920 par 1200 pour le client léger.

L'application doit fonctionner à une résolution de 1920 par 1080 pour le client lourd.

### 4.1. Utilisabilité

4.1.1 Un utilisateur novice doit terminer le tutoriel en moins de 2 minutes

4.1.2 L'interface de dessin doit être séparée visuellement des autres catégories de l'interface.

4.1.3 L'utilisateur novice doit pouvoir créer une partie en moins de 1 minute depuis le menu principal.

4.1.4 L'utilisateur expérimenté doit pouvoir créer une partie en moins de 30 secondes depuis le menu principal.

4.1.5 L'utilisateur novice doit pouvoir créer un canal de discussion en moins de 30 secondes depuis le menu principal.

4.1.6 L'utilisateur expérimenté doit pouvoir créer un canal de discussion en moins de 10 secondes depuis le menu principal.

4.1.7 La zone de clavardage doit avoir une largeur maximale de 30% de la taille réelle ("viewport") de la fenêtre.

### 4.2. Fiabilité

4.2.1 Le temps entre les pannes du serveur (*downtime*) doit être de moins de 365 heures par année.

4.2.2 Le temps moyen jusqu'au relancement du serveur doit être de moins de 1 heure.

4.2.3 L'application doit être en moyenne accessible à l'utilisateur au moins 23 heures par jour.

### 4.3. Performance

4.3.1 Le service d'authentification doit mettre au plus 500 ms pour connecter un client au serveur lors de l'authentification.

4.3.2 Le système doit envoyer un indice avec un délai d'au plus 500 ms après que l'utilisateur en ait fait la demande.

4.3.3 Le statut de l'utilisateur doit se mettre à jour avec un délai d'au plus 5s.

4.3.4 Une demande d'amis doit apparaître dans la liste d'amis de l'utilisateur au plus 500 ms après avoir été émise.

4.3.5 Si l'utilisateur accepte une demande d'amis d'un autre utilisateur, il doit apparaître dans la liste d'amis de ce dernier au plus 500 ms après qu'il ait accepté la demande.

4.3.6 Un message doit être visible à l'utilisateur dans un canal de clavardage avec un délai d'au plus 100 ms après que son auteur l'ait envoyé.



**4.3.7** L'historique de connexion dans le profil de l'utilisateur doit être mis à jour avec un délai d'au plus 100 ms après une connexion / déconnexion.

**4.3.8** Les statistiques dans le profil de l'utilisateur doivent être mises à jour avec un délai d'au plus 1 seconde après la fin d'une partie.

**4.3.9** Si l'utilisateur choisit de téléverser un fichier pour la création d'une paire mot-image, le service de conversion doit prendre au plus 5 secondes pour convertir son fichier en dessin.

**4.3.10** La base de données doit être mise à jour avec un délai d'au plus 5s après la création d'une paire mot-image.

#### **4.4. Maintenabilité**

**4.4.1** Les variables doivent être nommés en suivant la convention lower camel case (camelCase)

**4.4.2** Les attributs de classe publiques doivent être nommés en suivant la convention lower camel case (camelCase)

**4.4.3** Les fonctions doivent être nommés en suivant la convention lower camel case (camelCase)

**4.4.4** Les attributs de classe privés doivent être nommés en suivant la convention lower camel case en commençant par un trait de soulignement (\_camelCase)

**4.4.5** Les constantes doivent être nommés en suivant la convention screaming snake case (SCREAMING\_SNAKE\_CASE)

**4.4.8** Les classes doivent être nommés en utilisant la convention pascal case (CamelCase)

**4.4.9** La gestion des changements doit se faire avec Git

**4.4.10** Les commits Git doivent suivre le format suivant: «[type de changement] (id de la tâche sur Jira) description»

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

**4.5.1** Le langage de programmation utilisé pour la logique du client lourd doit être TypeScript.

**4.5.2** Les langages utilisés pour l'aspect visuel du client lourd doivent être HTML5 et CSS.

**4.5.3** Le client lourd doit utiliser le framework Angular.

**4.5.4** Le client lourd doit fonctionner sur Electron.

**4.5.5** Le langage de programmation utilisé pour le client léger doit être Kotlin.

**4.5.6** Le langage de programmation utilisé pour le serveur doit être TypeScript.

**4.5.7** Le serveur doit utiliser Express.

**4.5.8** Le serveur doit utiliser l'API socket.io.

**4.5.9** Le cadriciel de test doit être Jasmine.

#### 4.6. Sécurité

4.6.1 Les fichiers du code source doivent demeurer inaccessibles par l'utilisateur.

4.6.2 La console de développement doit demeurer inaccessible par l'utilisateur à partir de l'application.

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

4.7.1 Le tutoriel doit contenir toute l'information nécessaire afin de permettre à un utilisateur novice de pouvoir dessiner et deviner lors d'une partie.

### Annexes

Ordre	Equipe	Numéro du joueur dans l'équipe	Type
1	1	1	Joueur Humain
2	1	2	Joueur Humain
3	2	1	Joueur Humain
4	2	2	Joueur Humain

Tableau 1: Ordre des tours d'une manche d'une partie "Classique" si il n'y aucun joueur virtuel

Ordre	Equipe	Numéro du joueur dans l'équipe	Type
1	1	1	Joueur Humain
2	1	2	Joueur Humain
3	2	2	Joueur Virtuel

Tableau 2: Ordre des tours d'un manche d'un partie "Classique" avec une équipe avec un joueur virtuel (Equipe 2)

Ordre	Equipe	Numéro du joueur dans l'équipe	Type
1	1	2	Joueur Virtuel
2	2	2	Joueur Virtuel

Tableau 3: Ordre des tours d'un manche d'un partie "Classique" avec une équipe avec deux joueurs virtuels (Equipe 2)