
Équipe 107

DrawHub
Spécifications des requis du système (SRS)

Version 3.0

Historique des révisions

Date	Version	Description	Auteurs
2020-09-19	1.0	Rédaction initiale complétée par l'équipe de DrawHub.	Bruno Curzi-Laliberté William Glazer-Cavanagh Thomas Houtart Antoine Martin Simon Robatto-Simard Vincent Tessier
2020-10-02	2.0	Correction du document selon les commentaires de l'évaluation du 20 septembre.	Bruno Curzi-Laliberté William Glazer-Cavanagh Thomas Houtart Antoine Martin Simon Robatto-Simard Vincent Tessier
2020-12-02	3.0	Révision finale du SRS.	Bruno Curzi-Laliberté William Glazer-Cavanagh Thomas Houtart Antoine Martin Simon Robatto-Simard Vincent Tessier

Table des matières

Introduction	5
But	5
Définitions, acronymes et abréviations	5
Vue d'ensemble du document	7
Description globale	7
Caractéristiques d'utilisateurs	7
Interfaces	7
Contraintes générales	8
Exigences fonctionnelles	9
Clavardage	9
Accès à un compte existant	11
Création d'un compte	11
Confirmation d'un nouveau compte	11
Menu principal	12
Profil utilisateur	12
Personnalisation du profil utilisateur	13
Création d'une partie	13
Lobby	14
Mode spectateur	14
Exigences générales des modes de jeu	15
Mêlée générale	16
Sprint solo	17
Sprint coopératif	18
Bataille royale	19
Création d'une paire mot-image	19
Utilisateurs virtuels	21
Effets visuels et sonores	22
Tutoriel	22
Recherche rapide et par calibre	23
Trophées	23
Tableau de classement	24
Système anti-triche	24
Interface de dessin	25
Magasin et système de monnaie	25
Interface serveur	25
Exigences non fonctionnelles	26
Utilisabilité	26
Fiabilité	26
Performance	26
Maintenabilité	27
Contraintes de conception	27

Spécifications des requis du système (SRS)

1. Introduction

1.1. But

Le but du SRS décrit le comportement externe du système. 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

Angular : Cadriciel logiciel de développement web utilisé pour développer le client lourd.

Azure App : Plateforme d'hébergement d'infonuagique de Microsoft pour les applications web.

Canal de discussion actif : Canal de discussion ouvert dans l'interface de discussion.

Client léger : Application exécutable compatible avec une tablette Galaxy Tab A 2019 et compatible avec Android 9.0 Pie.

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

CosmosDB : Base de données infonuagique hébergée par Microsoft fonctionnant avec MongoDB.

CRAP : Acronyme anglais, principe de design permettant une faible charge cognitive et une facilité de familiarisation avec l'interface par l'utilisateur. CRAP - Contrast Repetition Alignment Proximity.

Cross-platform : caractéristique d'un jeu qui permet à des utilisateurs sur différentes plateformes (client léger et client lourd) de jouer contre dans une même partie.

Dessinateur : Rôle d'un utilisateur dans une partie active qui doit faire un dessin.

Détective : Rôle d'un utilisateur dans une partie active qui doit deviner un dessin.

DrawScore : Nombre entier situé entre 0 et 1000 qui permet de déterminer les habiletés de l'utilisateur.

Electron : Cadriciel à base Chromium qui permet de développer des applications web sous format exécutable pour tout système d'opération supportant Chromium.

ESLint : Analyseur de code statique permettant d'avertir le programmeur lorsque des règles de codage ou formatage ne sont pas respectées.

Express.JS : Cadriciel permettant de développer des serveurs d'applications web.

Lobby : Une interface d'attente où un ou plusieurs utilisateurs sont réunis avant que la partie débute.

GoF : Patrons de conception orientée objet de « Gang of four ». Tirés du livre *Design Patterns: Elements of Reusable Object-Oriented Software*.

Historique d'un canal de discussion : Ensemble des messages envoyés à un canal de discussion depuis sa création.

HTTP : Protocole réseau permettant une communication unidirectionnelle ou seul le client peut « tirer » de l'information du serveur.

HTTPS : Protocole HTTP sécurisé.

Kotlin : Langage de programmation utilisé pour développer l'application Android.

Ktlint : Analyseur de code statique permettant d'avertir le programmeur lorsque des règles de codage ou formatage ne sont pas respectées.

MEAN (stack) : Solution adaptée au développement d'applications web spécifiant quatre cadres et environnements d'exécution pour l'élaboration d'une application soit: MongoDB (base de données), Express.JS (serveur), Angular (client) et Node.JS (serveur).

MongoDB : Base de données NoSQL permettant d'enregistrer des documents sous format s'apparentant à JSON.

Menu principal : Interface permettant à l'utilisateur d'accéder à plusieurs fonctionnalités de l'application en un seul clic.

Multi-touch : L'interaction entre l'utilisateur et l'écran tactile d'une tablette.

Mongoose : Librairie permettant la connexion en Node.JS et une base de données MongoDB,

MVVM(Model-View-ViewModel) : Découplage de l'affichage et de la logique dans le but de faciliter le développement.

Node.JS : Environnement d'exécution compilant du code JavaScript hors navigateur web.

Paire mot-image : Combinaison d'un mot à deviner associé à son dessin enregistré sur la base de données.

Partie : Affrontement entre plusieurs utilisateurs où l'un d'entre eux dessine et le reste de ceux-ci tente de deviner un mot. Composée de plusieurs tours similaires.

Partie classée : Partie dans laquelle le système attribue ou retire des points du DrawScore aux utilisateurs dépendamment de leur classement final.

Partie non classée : Partie dans laquelle le classement final d'utilisateurs n'affecte pas leur DrawScore.

Socket.IO : Librairie implémentée en plusieurs langages, dont Kotlin et JavaScript permettant une communication bidirectionnelle basée sur des événements.

Sprint solo/coopératif : Mode de jeu où un ou plusieurs utilisateurs doivent deviner le plus de mots possible en un temps donné.

SRP : le Single Responsibility Principle désigne le fait qu'une classe ne devrait pas avoir plus d'une responsabilité dans une conception de code orienté objet.

SVG (Scalable Vector Graphics) : Format d'enregistrement d'image qui a une résolution infinie. Basé sur XML (eXtensible Markup Language).

TCP/IP : Protocole de communication OSI qui permet le transfert de données.

Thème : Palette de couleurs utilisée dans l'interface graphique des clients.

Titre d'un utilisateur : Décoration textuelle qu'un utilisateur peut obtenir pour décorer son pseudonyme (E.g. : Tony the Relentless).

Tour: Phase d'une partie dont le fonctionnement est propre au mode de jeu. Chaque partie comporte plusieurs tours dont le fonctionnement est identique hormis les rôles (dessinateur-détective) et le mot à deviner qui change.

TypeScript: Langage de programmation typé qui étend les fonctionnalités de JavaScript.

Utilisateur : Personne qui utilise un des deux clients.

Utilisateur virtuel : Intelligence artificielle qui dessine dans une partie active.

Websocket : Librairie permettant une communication bidirectionnelle orientée événement entre le client et le serveur.

1.3. Vue d'ensemble du document

Ce document est organisé en trois sections, soit une description globale du système à concevoir et un recueil détaillé des exigences fonctionnelles et non fonctionnelles de celui-ci.

2. Description globale

DrawHub est une continuité de Poly Dessin, un logiciel de dessin sur navigateur web. DrawHub est un jeu multijoueur qui permet aux utilisateurs de dessiner et deviner les dessins de ses adversaires et coéquipiers. Ce logiciel se joue en temps réel avec d'utilisateurs ainsi que d'utilisateurs virtuels sur client lourd et client léger.

2.1. Caractéristiques d'utilisateurs

Les utilisateurs de Drawhub sont des ingénieurs logiciels âgés entre 20 et 60 ans. Ils ont des connaissances techniques en logiciel vaste. Ils ont ainsi une connaissance des conventions d'applications mobiles et d'applications web modernes. Ils sont capables de manipuler les interfaces matérielles décrites en 2.2.2 avec aisance.

2.2. Interfaces

2.2.1. Interfaces utilisateurs

Les interfaces du client lourd et léger doivent être simples et intuitives et seront ainsi élaborées en suivant le principe de design CRAP. Elles seront basées sur des interfaces d'application similaire populaire comme Discord, FaceBook Messenger et iMessage pour la messagerie ou Discord et VisualStudio Code pour la barre de navigation de menu afin de respecter les conventions d'interfaces usager.

Les interfaces du client lourd seront quasi identiques à ceux du client léger pour que le jeu soit le plus

juste possible et ne donne en aucun temps un avantage aux joueurs sur un certain type de client. Le client lourd sera développé en HTML/CSS avec Angular 10. Le client léger quant à lui sera développé en XML et en Kotlin.

L'interface serveur sera hébergée sur le service Azure. Elle sera minimale d'avoir accès à certaines commandes à l'aide de boutons par exemple réinitialiser les bases de données. Elle aura aussi un affichage permettant de voir les événements sur le serveur comme les requêtes clients. Cette interface sera codée en HTML/JavaScript.

2.2.2. Interfaces matérielles

Les utilisateurs du client lourd utilisent le clavier et la souris. Les utilisateurs du client léger utilisent l'écran tactile de la tablette Samsung Galaxy Tab A 2019 avec support multi-touch. Les deux clients communiquent par haut-parleur avec l'utilisateur avec leur système de son.

2.2.3. Interfaces logicielles

Le client lourd fonctionne sous Windows 10 et est développé en TypeScript avec Angular 10 et utilise Electron pour déployer l'application. Le client léger fonctionne sous Android 9.0 Pie et est développé en Kotlin. Le serveur fonctionne avec le service Azure App Service hébergé sous Microsoft Azure et est développé avec Node.JS et Express.JS en TypeScript. Nous utiliserons une base de données MongoDB hébergée par CosmosDB sous Microsoft Azure qui communiquera avec le serveur à travers la librairie Mongoose.

Pour la conversion d'images en format bitmap et la conversion d'images de format .bmp vers le format SVG nous utiliserons PoTrace. Pour convertir les formats .jpg et .png vers .bpm nous utiliserons Jimp.

2.2.4. Interfaces de communication

Les clients communiquent avec le serveur par réseau avec la librairie Socket.IO qui est une solution utilisant plusieurs protocoles différents dont le principal est la librairie WebSocket pour toute communication bidirectionnelle basée sur des événements. Un API REST par HTTP/HTTPS est utilisé pour toutes les autres requêtes.

Le serveur communique avec la base de données à l'aide de la librairie JavaScript Mongoose qui repose sur le protocole TCP/IP.

2.3. Contraintes générales

Le serveur doit être en mesure de supporter au moins quatre parties simultanées avec au minimum 2 joueurs sans avoir plus de 500 ms de latence. Nous avons un nombre de parties concurrentes assez faible pour cette contrainte puisque nous sommes sur une version étudiante et gratuite d'Azure Web Apps, le budget de production sera géré en conséquence et permettra facilement de supporter plus

d'utilisateurs.

Le client lourd ne doit pas dépasser 1GB de mémoire vive et le client léger ne doit pas dépasser 500 Mb de mémoire vive.

2.4. Hypothèses et dépendances

Notre serveur étant hébergé sur les services Azure de Microsoft, nous faisons l'hypothèse que leur service CosmosDB et Azure Apps seront tous deux toujours disponible et fonctionnel pour la durée de vie de DrawHub.

Nous assumons que les utilisateurs sont familiers avec les conventions des applications web et mobiles modernes.

Nous assumons que les utilisateurs possèdent une bonne connexion réseau (au moins 5 Mb/s de téléchargement et 2Mb/s de téléversement sans coupures et une latence de moins de 200ms)

3. Exigences fonctionnelles

Les exigences sont classifiées par type (essentielle (E) ou souhaitable (S)) ainsi que par client, Mobile (M) et Desktop (D). Une omission de spécification (M) ou (D) indique implicitement une notation (MD) et celle-ci sera omise pour alléger la lecture.

Un exemple d'identifiant pour une exigence est : 3.X.X (ED) Le système doit <...>.

Ceci veut dire que l'exigence est essentielle et s'applique uniquement au client lourd.

3.1. Clavardage

- 3.1.1. (E) Le système doit ajouter chaque utilisateur au canal de discussion global lorsqu'il se connecte à DrawHub.
- 3.1.2. (E) Le système doit ajouter un utilisateur au canal de discussion d'une partie lorsqu'il rejoint le Lobby de celle-ci.
- 3.1.3. (E) Le système doit donner l'accès au canal de discussion d'une partie à un utilisateur de la partie.
- 3.1.4. (E) Le système doit cacher le mot qu'un utilisateur envoie dans le canal si ce mot s'avère être le mot à deviner dans un mode de jeu.
- 3.1.5. (E) Le système doit indiquer à un utilisateur lorsqu'il trouve le bon mot.
- 3.1.6. (E) Le système doit retirer un utilisateur d'un canal de discussion d'une partie lorsque l'utilisateur quitte la partie.
- 3.1.7. (E) Le système doit permettre à l'utilisateur de créer un canal de discussion en choisissant un nom de canal qui n'est pas actuellement utilisé.
- 3.1.8. (E) Le système doit permettre à l'utilisateur de supprimer un canal de discussion qu'il a créé.

- 3.1.9. (E) Le système doit permettre à l'utilisateur de quitter un canal de discussion.
- 3.1.10. (E) Le système doit permettre à l'utilisateur de consulter la liste de canaux de discussion existants.
- 3.1.11. (E) Le système doit permettre à l'utilisateur de rejoindre un canal de discussion.
- 3.1.12. (E) Le système doit permettre à l'utilisateur de consulter la liste des canaux auxquels il appartient.
- 3.1.13. (E) Le système doit permettre à l'utilisateur de rejoindre un canal de discussion en cherchant le nom du canal dans une barre de recherche.
- 3.1.14. (E) Le système doit permettre à l'utilisateur de faire partie d'au plus 15 canaux en même temps, en plus du canal général.
- 3.1.15. (E) Le système doit permettre à l'utilisateur d'afficher n'importe quel canal de discussion auquel il appartient.
- 3.1.16. (E) Le système doit permettre à un seul canal de discussion à la fois d'être actif.
- 3.1.17. (E) Le système doit permettre à l'utilisateur de rendre actif n'importe quel canal de discussion auquel il appartient.
- 3.1.18. (E) Le système doit afficher les messages en ordre chronologique dans le canal de discussion actif.
- 3.1.19. (E) Le système doit avoir un historique vide par défaut dans le canal de discussion actif.
- 3.1.20. (E) Le système doit permettre à l'utilisateur d'afficher les messages antérieurs par intervalles de 10 messages jusqu'à ce qu'il atteigne le premier message dans le canal de discussion actif.
- 3.1.21. (E) Le système doit permettre d'envoyer un message de taille maximale de 140 caractères dans le canal de discussion actif.
- 3.1.22. (E) Le système doit supporter l'encodage UTF-8 pour les messages dans les canaux de discussion.
- 3.1.23. (E) Le système doit afficher un message envoyé par un autre utilisateur dans le canal actif.
- 3.1.24. (E) Le système doit notifier visuellement un utilisateur lors de la réception d'un nouveau message dans un canal.
- 3.1.25. (E) Le système doit notifier à l'aide d'un son un utilisateur lors de la réception d'un nouveau message dans un canal.
- 3.1.26. (E) Le système doit permettre à l'utilisateur d'activer la tonalité de réception d'un message.
- 3.1.27. (E) Le système doit permettre à l'utilisateur de désactiver la tonalité de réception d'un message.
- 3.1.28. (E) Le système doit fournir un indicateur visuel pour chaque canal ayant un message non lu.
- 3.1.29. (ED) Le système doit pouvoir afficher la fenêtre de clavardage dans une fenêtre séparée à l'application.
- 3.1.30. (ED) Le système doit pouvoir afficher la fenêtre de clavardage dans l'application.
- 3.1.31. (ED) Le système doit permettre à l'utilisateur d'alterner entre le mode fenêtré et le mode intégré.
- 3.1.32. (E) Le système doit permettre l'accès à un canal de discussion depuis le menu principal.
- 3.1.33. (E) Le système doit permettre l'accès à un canal de discussion depuis un Lobby.

3.1.34. (E) Le système doit permettre l'accès à un canal de discussion pendant une partie.

3.2. Accès à un compte existant

- 3.2.1. (E) Le système doit avoir un menu d'authentification comme premier menu affiché lors de l'ouverture du client.
- 3.2.2. (E) Le système doit authentifier un utilisateur avant de lui afficher le menu principal en 3.6.
- 3.2.3. (E) Le système doit authentifier l'utilisateur avec son pseudonyme et mot de passe.
- 3.2.4. (E) Le système doit valider les informations entrées par l'utilisateur.
- 3.2.5. (E) Le système doit informer l'utilisateur d'une erreur dans le cas où les informations d'authentification ne sont pas valides.
- 3.2.6. (E) Le système doit permettre à un seul utilisateur à la fois de se connecter sur un compte.
- 3.2.7. (S) Le système doit permettre à l'utilisateur de réinitialiser son mot de passe à partir de menu d'authentification.

3.3. Création d'un compte

- 3.3.1. (E) Le système doit permettre à un utilisateur de créer un nouveau compte à partir du menu d'authentification.
- 3.3.2. (E) Le système doit avoir un nom de 2 à 20 caractères pour créer un compte.
- 3.3.3. (E) Le système doit avoir un prénom de 2 à 20 caractères pour créer un compte.
- 3.3.4. (E) Le système doit avoir un courriel unique pour créer un compte.
- 3.3.5. (E) Le système doit avoir un mot de passe de 4 à 20 caractères pour créer un compte.
- 3.3.6. (E) Le système doit rediriger l'utilisateur à la page d'authentification après la création d'un compte.

3.4. Confirmation d'un nouveau compte

- 3.4.1. (S) Le système doit envoyer un courriel de confirmation à l'adresse d'un nouvel utilisateur après la création de son compte.
- 3.4.2. (S) Le système doit fournir un code de validation de 6 chiffres aléatoires dans son courriel de confirmation.
- 3.4.3. (S) Le système doit enregistrer un compte temporaire avant l'envoi du courriel.
- 3.4.4. (S) Le système doit supprimer un compte temporaire si après 15 minutes il n'a pas été validé.
- 3.4.5. (S) Le système doit rediriger l'utilisateur se connectant avec un compte temporaire à la page de validation où il doit entrer son code.
- 3.4.6. (S) Le système doit permettre à l'utilisateur de demander de recevoir un nouveau courriel avec un nouveau code.
- 3.4.7. (S) Le système doit associer un seul code de confirmation par compte temporaire.
- 3.4.8. (S) Le système doit permettre à l'utilisateur de quitter la page de validation.

3.5. Menu principal

- 3.5.1. (E) Le système doit permettre à l'utilisateur de choisir le mode de jeu.
- 3.5.2. (E) Le système doit permettre à l'utilisateur de choisir la difficulté de la partie.
- 3.5.3. (E) Le système doit montrer la liste de Lobbys associés aux paramètres choisis par l'utilisateur.
- 3.5.4. (E) Le système doit permettre à l'utilisateur de joindre une partie.
- 3.5.5. (E) Le système doit permettre à l'utilisateur de créer une nouvelle partie.
- 3.5.6. (ED) Le système doit permettre à l'utilisateur d'accéder à la création d'une paire mot-image.
- 3.5.7. (E) Le système doit permettre à l'utilisateur d'accéder à son profil.
- ~~3.5.8. (E) Le système doit permettre à l'utilisateur d'accéder au magasin.~~
- 3.5.9. (E) Le système doit permettre à l'utilisateur d'accéder au tableau de classement.
- 3.5.10. (E) Le système doit fournir une façon de revenir au menu principal en tout temps.

3.6. Profil utilisateur

- 3.6.1. (E) Le système doit permettre à l'utilisateur de consulter le nombre de parties qu'il a jouées.
- 3.6.2. (E) Le système doit permettre à l'utilisateur de consulter son pourcentage de victoire.
- 3.6.3. (E) Le système doit permettre à l'utilisateur de consulter la durée moyenne de ses parties.
- 3.6.4. (E) Le système doit permettre à l'utilisateur de consulter le temps total passé à jouer.
- 3.6.5. (E) Le système doit permettre à l'utilisateur de consulter son meilleur score pour chaque mode de jeu.
- 3.6.6. (E) Le système doit permettre à l'utilisateur de consulter la date et l'heure de toutes ses connexions.
- 3.6.7. (E) Le système doit permettre à l'utilisateur de consulter la date et l'heure de toutes ses déconnexions.
- 3.6.8. (E) Le système doit permettre à l'utilisateur de consulter la date et l'heure du début de toutes les parties qu'il a jouées.
- 3.6.9. (E) Le système doit permettre à l'utilisateur de consulter le mode de jeu de toutes les parties qu'il a jouées.
- 3.6.10. (E) Le système doit permettre à l'utilisateur de consulter la liste d'utilisateurs ayant participé à toutes les parties qu'il a jouées.
- 3.6.11. (E) Le système doit permettre à l'utilisateur de consulter les scores finaux de toutes les parties qu'il a jouées.

3.7. Personnalisation du profil utilisateur

- 3.7.1. (E) Le système doit permettre à l'utilisateur de modifier son pseudonyme.
- 3.7.2. (E) Le système doit permettre à l'utilisateur de modifier son mot de passe.
- 3.7.3. (E) Le système doit permettre à l'utilisateur de modifier son avatar.
- 3.7.4. (S) Le système doit permettre à l'utilisateur de téléverser son avatar à partir d'une image locale carrée de type .png de 32x32 pixels.
- 3.7.5. (S) Le système doit permettre à l'utilisateur de changer le thème de son application.
- 3.7.6. (S) Le système doit permettre à l'utilisateur de modifier le son d'un début de partie.
- 3.7.7. (S) Le système doit permettre à l'utilisateur de modifier le son d'une victoire.
- 3.7.8. (S) Le système doit permettre à l'utilisateur de modifier le son d'une défaite.
- 3.7.9. (S) Le système doit permettre à l'utilisateur de modifier le son lorsqu'il devine un mot.
- 3.7.10. (S) Le système doit permettre à l'utilisateur de modifier le son lorsqu'un autre utilisateur devine un mot qu'il ne dessine pas.
- 3.7.11. (S) Le système doit permettre à l'utilisateur de modifier le son lorsqu'un autre utilisateur devine le mot qu'il dessine.
- 3.7.12. (S) Le système doit permettre à l'utilisateur de modifier le son d'une fin de tour.
- 3.7.13. (S) Le système doit permettre à l'utilisateur de choisir un titre parmi une liste.
- 3.7.14. (S) Le système doit afficher publiquement le titre d'un utilisateur.
- 3.7.15. (S) Le système doit permettre à l'utilisateur d'obtenir un titre en l'achetant du magasin.
- 3.7.16. (S) Le système doit permettre à l'utilisateur de modifier le titre actif.
- 3.7.17. (S) Le système doit afficher le titre actif d'un utilisateur avec son pseudonyme durant une partie.

3.8. Création d'une partie

- 3.8.1. (E) Le système doit permettre à l'utilisateur de créer une partie.
- 3.8.2. (E) Le système doit permettre à l'utilisateur de choisir le mode de jeu de la partie.
- 3.8.3. (E) Le système doit permettre à l'utilisateur de choisir la difficulté de la partie.
- 3.8.4. (E) Le système doit créer le Lobby d'une nouvelle partie.
- 3.8.5. (E) Le système doit rediriger l'utilisateur ayant créé une partie vers le Lobby correspondant.

3.9. Lobby

- 3.9.1. (E) Le système doit permettre à l'utilisateur de rejoindre Lobby.
- 3.9.2. (E) Le système doit assurer la présence d'au maximum 8 utilisateurs dans un Lobby.
- 3.9.3. (E) Le système doit afficher le nombre d'utilisateurs d'un Lobby.
- 3.9.4. (E) Le système doit permettre à l'utilisateur de quitter un Lobby.
- 3.9.5. (E) Le système doit permettre à l'utilisateur d'ajouter un utilisateur virtuel d'une partie lorsqu'il est dans le Lobby.
- 3.9.6. (E) Le système doit permettre à l'utilisateur d'enlever un utilisateur virtuel d'une partie lorsqu'il est dans le Lobby.
- 3.9.7. (E) Le système doit créer un canal de discussion pour chaque Lobby.
- 3.9.8. (E) Le système doit ajouter un nouvel utilisateur au canal de discussion dès son arrivée dans le Lobby.
- 3.9.9. (E) Le système doit effacer les messages du canal de discussion lors du départ de la partie.
- 3.9.10. (E) Le système doit enlever un utilisateur quittant le Lobby du canal de discussion.
- 3.9.11. (E) Le système doit permettre à un utilisateur de lancer une partie lorsque le nombre d'utilisateurs minimal a été atteint.
- 3.9.12. (E) Le système doit supprimer un Lobby lorsque l'utilisateur ayant créé la partie quitte le Lobby.

3.10. Mode spectateur

- 3.10.1. (E) Le système doit permettre à l'utilisateur de rejoindre une partie en cours en mode spectateur.
- 3.10.2. (E) Le système doit permettre à un utilisateur de joindre un lobby en mode spectateur.
- 3.10.3. (E) Le système doit permettre à l'utilisateur en mode spectateur de voir le dessin en cours.
- 3.10.4. (E) Le système doit permettre à l'utilisateur en mode spectateur de voir le canal de discussion.
- 3.10.5. (E) Le système doit inclure le spectateur dans la liste d'utilisateurs visible lors d'une partie.
- 3.10.6. (E) Le système doit fournir un moyen de différencier le spectateur d'un utilisateur non-spectateur.
- 3.10.7. (E) Le système doit permettre à l'utilisateur en mode spectateur de quitter une partie.

3.11. Exigences générales des modes de jeu

- 3.11.1. (E) Le système doit notifier un utilisateur du début d'une partie.
- 3.11.2. (E) Le système doit notifier un utilisateur du début d'un tour.
- 3.11.3. (E) Le système doit notifier un utilisateur de la fin d'un tour.
- 3.11.4. (E) Le système doit notifier un utilisateur de la fin d'une partie.
- 3.11.5. (E) Le système doit assigner le rôle de dessinateur à un utilisateur au début de chaque tour.
- 3.11.6. (E) Le système doit assigner le rôle de détective aux utilisateurs qui ne sont pas le dessinateur au début de chaque tour.
- 3.11.7. (E) Le système doit assigner un mot au dessinateur au début de chaque tour.
- 3.11.8. (E) Le système doit commencer un chronomètre au début de chaque tour.
- 3.11.9. (S) Le système doit toujours afficher le temps restant sur le chronomètre.
- 3.11.10. (E) Le système doit passer au tour suivant à la fin du chronomètre
- 3.11.11. (E) Le système doit passer au tour suivant lorsque tous les détectives ont trouvé le mot à deviner.
- 3.11.12. (E) Le système doit attribuer des points à un utilisateur ayant écrit le mot à deviner dans le canal de discussion de la partie.
- 3.11.13. (E) Le système doit offrir la difficulté facile.
- 3.11.14. (E) Le système doit offrir la difficulté normale.
- 3.11.15. (E) Le système doit offrir la difficulté difficile.
- 3.11.16. (E) Le système doit, à la fin d'une partie, afficher le classement de chaque utilisateur.
- 3.11.17. (E) Le système doit, à la fin d'une partie, offrir un moyen de retourner au menu principal.
- 3.11.18. (E) Le système doit permettre à l'utilisateur de quitter une partie à n'importe quel moment.

3.12. Mêlée générale

- 3.12.1. (E) Le système doit assurer un minimum de 2 utilisateurs non virtuels dans une partie.
- 3.12.2. (E) Le système doit comptabiliser le nombre d'utilisateurs virtuels dans le nombre total d'utilisateurs.
- 3.12.3. (E) Le système doit lancer une partie lorsqu'il y a au moins 3 utilisateurs.
- 3.12.4. (E) Le système doit choisir le prochain dessinateur dans le même ordre que celui d'arrivée dans le Lobby.
- 3.12.5. (E) Le système doit, lorsque tous les utilisateurs ont été dessinateurs, recommencer le cycle d'attribution du rôle de dessinateur à partir de l'utilisateur choisi au début de la partie.
- 3.12.6. (E) Le système doit afficher le score à un utilisateur à la fin de la partie.
- 3.12.7. (E) Le système doit attribuer le pointage à un utilisateur qui devine le bon mot, selon la formule :

$$100 + \max(0, 10t - 50 * n/N)$$

Où :

t = nombre de secondes restantes ;

n = nombre d'utilisateurs ayant déjà deviné le mot ;

N = nombre d'utilisateurs.

- 3.12.8. (E) Le système doit attribuer le pointage à un utilisateur qui dessine, selon la formule :

$$\sum_{i=1}^n (\frac{P_i}{2} + 25Ti) + E$$

Où :

n = nombre d'utilisateurs ayant deviné le mot ;

P = nombre de points reçus par un utilisateur *i* ayant deviné le mot ;

E = 250 si tous les utilisateurs ont deviné le mot, sinon 0 ;

Ti = nombre de secondes restantes après qu'un utilisateur *i* ait deviné le mot.

3.13. Sprint solo

- 3.13.1. (S) Le système doit attribuer le rôle de dessinateur à un utilisateur virtuel.
- 3.13.2. (S) Le système doit permettre 10 essais pour deviner le mot d'un dessin en difficulté facile.
- 3.13.3. (S) Le système doit toujours afficher le nombre d'essais restants.
- 3.13.4. (S) Le système doit, lorsque le nombre d'essais est écoulé, passer au prochain mot à deviner.
- 3.13.5. (S) Le système doit toujours afficher le pointage cumulé de l'utilisateur.
- 3.13.6. (S) Le système doit, lorsqu'en difficulté facile, allouer 10 essais à un utilisateur pour deviner le mot.
- 3.13.7. (S) Le système doit, lorsqu'en difficulté facile, accorder 50 points à un utilisateur ayant trouvé le bon mot.
- 3.13.8. (S) Le système doit, lorsqu'en difficulté facile, accorder 5 secondes supplémentaires à un utilisateur ayant trouvé le bon mot.
- 3.13.9. (S) Le système doit, lorsqu'en difficulté normale, allouer 7 essais à un utilisateur pour deviner le mot.
- 3.13.10. (S) Le système doit, lorsqu'en difficulté normale, accorder 100 points à un utilisateur ayant trouvé le bon mot.
- 3.13.11. (S) Le système doit, lorsqu'en difficulté normale, accorder 4 secondes supplémentaires à un utilisateur ayant trouvé le bon mot.
- 3.13.12. (S) Le système doit, lorsqu'en difficulté difficile, allouer 5 essais à un utilisateur pour deviner le mot.
- 3.13.13. (S) Le système doit, lorsqu'en difficulté difficile, accorder 150 points à un utilisateur ayant trouvé le bon mot.
- 3.13.14. (S) Le système doit, lorsqu'en difficulté difficile, accorder 3 secondes supplémentaires à un utilisateur ayant trouvé le bon mot.

3.14. Sprint coopératif

- 3.14.1. (S) Le système doit attribuer le rôle de dessinateur à un utilisateur virtuel.
- 3.14.2. (S) Le système doit assurer un minimum de 2 utilisateurs non-virtuels dans une partie.
- 3.14.3. (S) Le système doit assurer un maximum de 4 utilisateurs non-virtuels dans une partie.
- 3.14.4. (S) Le système doit assigner un chronomètre partagé par tous les utilisateurs.
- 3.14.5. (S) Le système doit toujours afficher le chronomètre restant.
- 3.14.6. (S) Le système doit, lorsque le chronomètre est écoulé, terminer la partie
- 3.14.7. (S) Le système doit allouer un nombre maximal d'essais pour deviner le mot par dessin.
- 3.14.8. (S) Le système doit toujours afficher le nombre d'essais restants.
- 3.14.9. (S) Le système doit, lorsque le nombre d'essais est écoulé, passer au prochain mot à deviner.
- 3.14.10. (S) Le système doit toujours afficher le pointage accumulé par chaque utilisateur.
- 3.14.11. (S) Le système doit, lorsqu'en difficulté facile, allouer 10 essais à un utilisateur pour deviner le mot.
- 3.14.12. (S) Le système doit, lorsqu'en difficulté facile, accorder 50 points à un utilisateur ayant trouvé le bon mot.
- 3.14.13. (S) Le système doit, lorsqu'en difficulté facile et qu'un utilisateur trouve le bon mot, allouer 5 secondes de temps supplémentaire à tous les utilisateurs.
- 3.14.14. (S) Le système doit, lorsqu'en difficulté facile et qu'un utilisateur trouve le bon mot, accorder le nombre de points suivants à tous les utilisateurs exceptés celui qui a trouvé le mot :
$$25 \text{ points} / (\text{nombre d'utilisateurs} - 1)$$
- 3.14.15. (S) Le système doit, lorsqu'en difficulté normale, allouer 7 essais à un utilisateur pour deviner le mot d'un dessin.
- 3.14.16. (S) Le système doit, lorsqu'en difficulté normale, accorder 100 points à un utilisateur ayant trouvé le bon mot.
- 3.14.17. (S) Le système doit, lorsqu'en difficulté normale et qu'un utilisateur trouve le bon mot, allouer 4 secondes de temps supplémentaire à tous les utilisateurs.
- 3.14.18. (S) Le système doit, lorsqu'en difficulté normale et qu'un utilisateur trouve le bon mot, accorder le nombre de points suivants à tous les utilisateurs exceptés celui qui a trouvé le mot :
$$50 \text{ points} / (\text{nombre d'utilisateurs} - 1)$$
- 3.14.19. (S) Le système doit, lorsqu'en difficulté difficile, allouer 5 essais à un utilisateur pour deviner le mot.
- 3.14.20. (S) Le système doit, lorsqu'en difficulté difficile, accorder 50 points à un utilisateur ayant trouvé le bon mot.
- 3.14.21. (S) Le système doit, lorsqu'en difficulté difficile et qu'un utilisateur trouve le bon mot, allouer 5 secondes de temps supplémentaire à tous les utilisateurs.
- 3.14.22. (S) Le système doit, lorsqu'en difficulté difficile et qu'un utilisateur trouve le bon mot, accorder le nombre de points suivants à tous les utilisateurs exceptés celui qui a trouvé le mot :

3.15. Bataille royale

- 3.15.1. (S) Le système doit attendre un minimum de 4 utilisateurs avant de débiter la partie.
- 3.15.2. (S) Le système doit assurer un maximum de 10 utilisateurs dans une partie.
- 3.15.3. (S) Le système doit empêcher l'ajout de joueurs virtuels dans la partie.
- 3.15.4. (S) Le système doit éliminer au hasard un utilisateur qui n'a pas deviné le mot à la fin du chronomètre.
- 3.15.5. (S) Le système doit mettre un utilisateur éliminé en mode spectateur.
- 3.15.6. (S) Le système doit éliminer le dessinateur si aucun détective n'a trouvé le mot.
- 3.15.7. (S) Le système doit, lorsqu'il reste deux utilisateurs non éliminés, désigner un utilisateur virtuel comme dessinateur.
- 3.15.8. (S) Le système doit, lorsqu'il reste deux utilisateurs non éliminés, désigner le premier utilisateur devinant le mot comme gagnant.
- 3.15.9. (S) Le système doit, lorsqu'il reste deux utilisateurs non éliminés et qu'ils ne trouvent pas le mot à la fin du chronomètre, passer au prochain mot.

3.16. Création d'une paire mot-image

- 3.16.1. (ED) Manuelle I :
 - 3.16.1.1. (ED) *Le système doit permettre à l'utilisateur d'ajouter une paire mot-image à la banque de jeux.*
 - 3.16.1.2. (ED) *Le système doit obliger l'utilisateur d'entrer son choix de mot ou d'expression associé à l'image.*
 - 3.16.1.3. (ED) *Le système doit obliger l'utilisateur de spécifier un indice pour le pair ajouté.*
 - 3.16.1.4. (ED) *Le système doit obliger l'utilisateur de dessiner le mot ou l'expression choisie lors de la création de la paire.*
 - 3.16.1.5. (ED) *Le système doit fournir les mêmes fonctionnalités que lors d'une partie dans la zone de dessin.*
 - 3.16.1.6. (ED) *Le système doit permettre à l'utilisateur de choisir la difficulté associée à son dessin.*
 - 3.16.1.7. (ED) *Le système doit fournir un aperçu du traçage final à l'utilisateur.*
 - 3.16.1.8. (ED) *Le système doit pouvoir montrer la liste des paires mot-image existantes.*
- 3.16.2. (SD) Manuelle II :
 - ~~3.16.2.1. (SD) Le système doit permettre à l'utilisateur de choisir le mode de formation du dessin par un utilisateur virtuel.~~
 - ~~3.16.2.2. (SD) Mode classique : Le système doit afficher chaque trait dans le même ordre que lors de la création du dessin par l'utilisateur.~~
 - ~~3.16.2.3. (SD) Mode aléatoire : Le système doit afficher chaque élément individuel du dessin dans~~

~~un ordre aléatoire.~~

~~3.16.2.4. (SD) Mode panoramique : Le système doit afficher les traits dans l'ordre de leur position sur les axes cartésiens, soit vertical ou horizontal.~~

~~3.16.2.5. (SD) Mode centré : Le système doit afficher les traits dans l'ordre de leur distance en pixel au centre de l'image avec comme maximum la taille du dessin.~~

~~3.16.3. (SD) Assistée I :~~

~~3.16.3.1. (SD) Le système doit permettre à l'utilisateur de pouvoir créer une paire mot-image à partir d'un fichier d'image locale.~~

~~3.16.3.2. (SD) Le système doit pouvoir traiter une image de type BMP.~~

~~3.16.3.3. (SD) Le système doit pouvoir traiter une image de type JPG.~~

~~3.16.3.4. (SD) Le système doit pouvoir traiter une image de type png.~~

~~3.16.3.5. (SD) Le système doit pouvoir convertir l'image choisie en un format utilisable par l'application.~~

~~3.16.3.6. (SD) Le système doit permettre à l'utilisateur de sélectionner différents paramètres de l'engin de conversion.~~

~~3.16.3.7. (SD) Le système doit omettre le mode de dessin classique des choix du mode de dessin.~~

~~3.16.4. (SD) Assistée II :~~

~~3.16.4.1. (SD) Le système doit permettre à l'utilisateur de choisir la paire mot-image à partir de la banque Quick Draw de Google.~~

~~3.16.4.2. (SD) Le système doit permettre à l'utilisateur de spécifier un indice.~~

~~3.16.4.3. (SD) Le système doit permettre à l'utilisateur de choisir le mode de dessin.~~

~~3.16.4.4. (SD) Le système doit permettre à l'utilisateur de choisir la difficulté du dessin.~~

~~3.16.5. (SD) Assistée III :~~

~~3.16.5.1. (SD) Le système doit permettre à l'utilisateur de chercher l'image voulue sur le web.~~

~~3.16.5.2. (SD) Le système doit permettre à l'utilisateur d'entrer des mots-clés pour exécuter la recherche.~~

~~3.16.5.3. (SD) Le système doit présenter 3 choix d'image reliée au mot ou à l'expression clé.~~

~~3.16.5.4. (SD) Le système doit permettre à l'utilisateur de faire la sélection d'une image parmi celles présentées.~~

~~3.16.5.5. (SD) Le système doit permettre à l'utilisateur de changer la sélection d'images présentées par une nouvelle sélection.~~

~~3.16.5.6. (SD) Le système doit être en mesure de convertir l'image sélectionnée en un format utilisable par l'application.~~

3.17. Utilisateurs virtuels

- 3.17.1. (E) Le système doit permettre à l'utilisateur virtuel de communiquer à partir du canal de communication d'une partie.
- 3.17.2. (E) Le système doit permettre à l'utilisateur de demander un indice à un utilisateur virtuel.
- 3.17.3. (E) Le système doit permettre, lorsqu'un utilisateur demande un indice, à l'utilisateur virtuel de communiquer un indice.
- 3.17.4. (E) Le système doit permettre à l'utilisateur virtuel de féliciter l'utilisateur lors d'une bonne réponse dans une partie.
- 3.17.5. (E) Le système doit permettre à l'utilisateur virtuel de publier un message au début de la partie.
- 3.17.6. (E) Le système doit permettre à l'utilisateur virtuel de publier un message à la fin de chaque tour.
- 3.17.7. (S) Le système doit permettre à l'utilisateur virtuel d'avoir une personnalité unique lors d'une partie.
- 3.17.8. (S) Le système doit permettre à l'utilisateur virtuel de référencer des parties antérieures d'une des manières suivantes :
 - 3.17.8.1. (S) *Le système doit, lorsqu'un utilisateur est proche de battre le pointage de la partie précédente, permettre à l'utilisateur virtuel d'envoyer un message.*
 - 3.17.8.2. (S) *Le système doit, lorsqu'un utilisateur est proche de battre son meilleur pointage enregistré dans le mode de jeu courant, permettre à l'utilisateur virtuel d'envoyer un message.*
 - 3.17.8.3. (S) *Le système doit permettre à l'utilisateur virtuel d'envoyer un message pour féliciter un utilisateur qui a dépassé son pointage de la partie précédente*
 - 3.17.8.4. (S) *Le système doit permettre à un utilisateur virtuel d'envoyer un message pour féliciter un utilisateur qui a dépassé son meilleur pointage à vie dans le mode de jeu courant.*
 - 3.17.8.5. (S) *Le système doit permettre à l'utilisateur virtuel de saluer un utilisateur avec lequel il a déjà joué.*
- 3.17.9. (S) Le système doit permettre à l'utilisateur virtuel d'imiter une personnalité connue:
 - 3.17.9.1. (S) *Le système doit permettre à l'utilisateur virtuel de posséder un nom de personnalité connue.*
 - 3.17.9.2. (S) *Le système doit permettre à l'utilisateur virtuel de formuler un message avec une phrase iconique de la personnalité connue qu'il imite.*
- 3.17.10. (S) Le système doit, lors d'une partie, permettre à l'utilisateur virtuel de dessiner les traits dans l'ordre spécifié lors de la création de la paire mot-image.

3.18. Effets visuels et sonores

- 3.18.1. (E) Le système doit pouvoir produire un effet sonore lors de la réception d'un message.
- 3.18.2. (S) Le système doit pouvoir produire un effet sonore lors d'un début de partie.
- 3.18.3. (S) Le système doit pouvoir produire un effet sonore lors d'une fin de partie.
- 3.18.4. (S) Le système doit pouvoir produire un effet sonore lors d'un début d'un tour.
- 3.18.5. (S) Le système doit pouvoir produire un effet sonore lors d'une fin de tour.
- 3.18.6. (S) Le système doit pouvoir produire un effet sonore lors de l'entrée du mot recherché lors d'une partie.
- 3.18.7. (S) Le système doit pouvoir produire un effet sonore lors de la sortie d'un utilisateur dans une partie ou un Lobby.
- 3.18.8. (E) Le système doit pouvoir produire un effet visuel lors de la réception d'un message.
- 3.18.9. (S) Le système doit pouvoir produire un effet visuel lors d'un début de partie.
- 3.18.10. (S) Le système doit pouvoir produire un effet visuel lors d'une fin de partie.
- 3.18.11. (S) Le système doit pouvoir produire un effet visuel lors d'un début d'un tour.
- 3.18.12. (S) Le système doit pouvoir produire un effet visuel lors d'une fin de tour.
- 3.18.13. (S) Le système doit pouvoir produire un effet visuel lors d'une victoire.
- 3.18.14. (S) Le système doit pouvoir produire un effet visuel lors d'une défaite.

3.19. Tutoriel

- 3.19.1. (E) Le système doit posséder un tutoriel expliquant les modes de jeu.
- 3.19.2. (E) Le système doit, lors de la première connexion d'un utilisateur, afficher le tutoriel.
- 3.19.3. (E) Le système doit permettre à l'utilisateur de quitter le tutoriel à n'importe quel moment.
- 3.19.4. (E) Le système doit posséder un tutoriel expliquant comment créer une partie.
- 3.19.5. (E) Le système doit posséder un tutoriel expliquant comment joindre une partie.
- 3.19.6. (E) Le système doit posséder un tutoriel expliquant comment créer un canal de discussion.
- 3.19.7. (E) Le système doit posséder un tutoriel expliquant comment joindre un canal de discussion.
- 3.19.8. (ED) Le système doit posséder un tutoriel expliquant comment créer une paire mot-image.

3.20. Recherche rapide et par calibre

- 3.20.1. (S) Le système doit permettre à l'utilisateur de trouver une partie sans consulter la liste de lobby à l'aide d'une recherche rapide.
- 3.20.2. (S) Le système doit permettre à l'utilisateur de décider si une partie est classée.
- 3.20.3. (S) Le système doit, si aucune partie existante ne contient les paramètres choisis, créer une nouvelle partie respectant les paramètres de recherche choisis par l'utilisateur.
- 3.20.4. (S) Le système doit permettre à l'utilisateur de posséder un pointage nommé *DrawScore*.
- 3.20.5. (S) Le système doit attribuer 500 points de *DrawScore* à l'utilisateur à la création de son compte.
- 3.20.6. (S) Le système doit ajouter des points au *DrawScore* de l'utilisateur lorsque celui-ci gagne une partie de type *classée*.
- 3.20.7. (S) Le système doit enlever des points au *DrawScore* de l'utilisateur lorsque celui-ci perd une partie de type *classée*.
- 3.20.8. (S) Le système doit afficher le *DrawScore* de l'utilisateur dans son profil.
- 3.20.9. (S) Le système doit, lors d'une recherche automatique, pouvoir placer les utilisateurs possédant un *DrawScore* similaire dans une même partie.
- 3.20.10. (S) Le système doit, dans le cas où aucun utilisateur possédant un *DrawScore* similaire est disponible, proposer à l'utilisateur de rejoindre une partie avec des utilisateurs de pointage différent.
- 3.20.11. (S) Le système doit permettre à l'utilisateur courant de visionner le *DrawScore* d'un autre utilisateur en accédant à son profil.

3.21. Trophées

- 3.21.1. (S) Le système doit, lorsque l'utilisateur gagne un nombre préétabli de parties, permettre à celui-ci d'obtenir un trophée virtuel.
- 3.21.2. (S) Le système doit, lorsque l'utilisateur se connecte n jours de suite, permettre à celui-ci d'obtenir un trophée virtuel.
- 3.21.3. (S) Le système doit, lorsque l'utilisateur devine le bon mot en moins de n secondes, permettre à celui-ci d'obtenir un trophée virtuel.
- 3.21.4. (S) Le système doit sauvegarder un trophée obtenu par un utilisateur.
- 3.21.5. (S) Le système doit permettre à l'utilisateur d'accéder à sa liste de trophées à partir de son profil.
- 3.21.6. (S) Le système doit permettre à l'utilisateur d'accéder à la galerie de trophées d'un autre utilisateur à partir du profil de ce dernier.
- 3.21.7. (S) Le système doit permettre à l'utilisateur de voir la liste des trophées qu'il n'a pas encore obtenus.
- 3.21.8. (S) Le système doit afficher les requis pour obtenir un trophée.

3.22. Tableau de classement

- 3.22.1. (S) Le système doit avoir un tableau de classement selon le nombre de parties gagnées.
- 3.22.2. (S) Le système doit avoir un tableau de classement selon le DrawScore.
- 3.22.3. (S) Le système doit avoir un tableau de classement selon le nombre de PaintDrops.
- 3.22.4. (S) Le système doit pouvoir afficher un tableau de classement avec les données de la journée.
- 3.22.5. (S) Le système doit pouvoir afficher un tableau de classement avec les données de la semaine.
- 3.22.6. (S) Le système doit pouvoir afficher un tableau de classement avec toutes les données.
- 3.22.7. (S) Le système doit pouvoir afficher un tableau de classement en ordre décroissant.
- 3.22.8. (S) Le système doit afficher le classement de l'utilisateur dans l'interface du tableau de classement.

3.23. Système anti-triche

- 3.23.1. (S) Le système doit donner une offense à un utilisateur qui entre plus de 10 mots en moins de 5 secondes dans un utilisateur de clavardage.
- 3.23.2. (S) Le système doit, lorsque l'utilisateur commet une offense, avertir celui-ci dans le canal.
- 3.23.3. (S) Le système doit, à la première offense, bloquer le clavardage de l'utilisateur pour 20 secondes.
- 3.23.4. (S) Le système doit, à la deuxième offense, éliminer l'utilisateur du tour en cours.
- 3.23.5. (S) Le système doit, à la troisième offense, retirer l'utilisateur de la partie.
- 3.23.6. (S) Le système doit, à la suite d'un retrait, informer les autres utilisateurs du retrait.
- 3.23.7. (S) Le système doit, à la suite d'un retrait, informer les autres utilisateurs de la raison du retrait.
- 3.23.8. (S) Le système doit, à la suite d'un retrait, enregistrer une offense de type « *kick* » dans le profil de l'utilisateur.
- 3.23.9. (S) Le système doit, à la suite d'un « *kick* », être inutilisable par l'utilisateur pour un nombre de minutes égal au nombre de fois que l'utilisateur a été « *kick* » au carré.
- 3.23.10. (S) Le système doit bannir un utilisateur ayant reçu 5 « *kicks* » ou plus.
- 3.23.11. (S) Le système doit interdire la connexion d'un utilisateur banni.
- 3.23.12. (S) Le système doit, lors d'une partie, permettre à l'utilisateur d'initialiser un vote pour retirer un utilisateur d'une partie avec la commande « `/votekick <username>` ».
- 3.23.13. (S) Le système doit allouer 30 secondes à un utilisateur pour voter.
- 3.23.14. (S) Le système doit invalider le vote après 30 secondes.
- 3.23.15. (S) Le système doit informer les utilisateurs du vote en cours, sauf l'utilisateur ciblé.
- 3.23.16. (S) Le système doit, durant un vote, permettre à un utilisateur de voter en entrant la commande « `/votekick yes` ».
- 3.23.17. (S) Le système doit, lorsque 70% d'utilisateurs concernés ont voté en faveur, accepter un vote.
- 3.23.18. (S) Le système doit, lorsque le vote est accepté, retirer l'utilisateur ciblé par un vote.
- 3.23.19. (S) Le système doit catégoriser le retrait d'un utilisateur comme une offense de type « *kick* ».

3.23.20. (S) Le système doit désactiver la fonction « votekick » pour une partie ayant moins de 4 utilisateurs non virtuels.

3.24. Interface de dessin

- 3.24.1. (E) Le système doit permettre à l'utilisateur de dessiner avec le crayon.
- 3.24.2. (E) Le système doit permettre à l'utilisateur d'effacer un trait de crayon avec l'efface.
- 3.24.3. (E) Le système doit permettre à l'utilisateur de changer la taille du tracé du crayon.
- 3.24.4. (E) Le système doit permettre à l'utilisateur de changer l'opacité du tracé du crayon.
- 3.24.5. (E) Le système doit permettre à l'utilisateur d'alterner entre le crayon et l'efface.
- 3.24.6. (E) Le système doit permettre à l'utilisateur de défaire la dernière action qui affecte le dessin.
- 3.24.7. (E) Le système doit permettre à l'utilisateur de refaire la dernière action qui affecte le dessin.
- 3.24.8. (E) Le système doit fournir un moyen de changer la couleur du tracé.
- 3.24.9. (E) Le système doit montrer le temps restant au tour.
- 3.24.10. (E) Le système doit donner à l'utilisateur accès à la grille de dessin.
- 3.24.11. (E) Le système doit permettre à l'utilisateur de changer la taille des carrés de la grille.
- 3.24.12. (E) Le système doit permettre à l'utilisateur de changer l'opacité de la grille.

~~3.25. Magasin et système de monnaie~~

- ~~3.25.1. (S) Le système doit permettre à l'utilisateur de gagner des *PaintDrops* en fonction des points obtenus à la fin d'une partie.~~
- ~~3.25.2. (S) Le système doit permettre à l'utilisateur de gagner des *PaintDrops* après avoir obtenu un trophée.~~
- ~~3.25.3. (S) Le système doit permettre à l'utilisateur de dépenser ses *PaintDrops* dans une interface prévue à cet effet.~~
- ~~3.25.4. (S) Le système doit bloquer certaines options de personnalisation de profil.~~
- ~~3.25.5. (S) Le système doit d'acheter avec les *PaintDrops* les options de personnalisation via un magasin.~~

~~3.26. Interface serveur~~

- ~~3.26.1. (S) Le système doit avoir une interface serveur à l'adresse <https://drawhub.azurewebsites.net>.~~
- ~~3.26.2. (S) Le système doit pouvoir effacer les données sur une collection dans la base de données à l'aide d'un bouton.~~

4. Exigences non fonctionnelles

4.1. Utilisabilité

- 4.1.1. L'interface doit suivre les principes de CRAP (contraste, répétition, alignement et proximité).
 - 4.1.1.1. *La couleur de chaque bouton doit être similaire.*
 - 4.1.1.2. *La taille des boutons doit être similaire.*
 - 4.1.1.3. *L'intention des boutons doit être clairement indiquée.*
 - 4.1.1.4. *Les éléments interactifs doivent être similaires.*
 - 4.1.1.5. *Les éléments non interactifs doivent être similaires.*
 - 4.1.1.6. *Les éléments interactifs doivent être différents des éléments non interactifs.*
 - 4.1.1.7. *Les éléments partagés par différentes pages doivent être placés aux mêmes endroits.*
 - 4.1.1.8. *Les éléments partagés par différentes pages doivent être identiques.*
 - 4.1.1.9. *Les éléments d'une même fonctionnalité sur une page doivent être placés à proximité.*
 - 4.1.1.10. *Les éléments d'une fonctionnalité différente sur une page ne doivent pas être placés à proximité.*
 - 4.1.1.11. *Les éléments d'une même fonctionnalité sur une page doivent être alignés.*
 - 4.1.1.12. *Les éléments d'une fonctionnalité différente sur une page ne doivent pas être alignés.*
 - 4.1.1.13. *Le contraste entre les éléments sur une page doit faciliter la lecture.*
 - 4.1.1.14. *Le contraste entre les éléments sur une page doit mettre en évidence les actions possibles.*
 - 4.1.1.15. *Le contraste entre les éléments sur une page doit permettre de balayer visuellement la page en 15 secondes.*
- 4.1.2. L'interface doit suivre les heuristiques de Nielsen [\[ref\]](#)
- 4.1.3. Un nouvel utilisateur doit pouvoir créer un lobby en une minute.
- 4.1.4. Un nouvel utilisateur doit pouvoir créer un compte en une minute.
- 4.1.5. Un nouvel utilisateur doit pouvoir compléter le tutoriel en une minute.
- 4.1.6. Un nouvel utilisateur doit pouvoir créer un canal de texte en une minute.
- 4.1.7. Un utilisateur doit se sentir heureux en utilisant le client.

4.2. Fiabilité

- 4.2.1. Le système doit être disponible 99% du temps dans une année.
- 4.2.2. Le système doit avoir un temps moyen pour réparation inférieure à 1 heure.
- 4.2.3. Le système doit avoir un temps moyen avant échec supérieur à un mois.

4.3. Performance

- 4.3.1. Le système doit afficher les changements d'interface utilisateur en moins de 700 ms.
- 4.3.2. Le serveur doit traiter une requête en moins de 250 ms.
- 4.3.3. Le serveur doit envoyer une réponse en moins de 250 ms.

4.3.4. Le serveur doit supporter l'activité d'au moins 10 canaux de communications simultanés.

4.3.5. Le serveur doit supporter l'activité de 4 parties simultanées.

4.4. Maintenabilité

4.4.1. Le code source doit être écrit en anglais.

4.4.2. Le code source écrit en TypeScript doit respecter les règles de programmation par défaut d'ESLint.

4.4.3. Le code source écrit en Kotlin doit respecter les règles de programmation par défaut d'Android Lint.

4.4.4. Le code source doit comprendre au maximum une déclaration par ligne.

4.4.5. Le code source doit être testé avec une couverture d'au minimum 80% des lignes.

4.4.6. Le code source doit être testé avec une couverture d'au minimum 80% des branches.

4.4.7. Le code source doit être testé avec une couverture de 95% des fonctions.

4.4.8. Le code source doit être constitué d'au moins 80% de fonctions de moins de 10 lignes.

4.4.9. Le code source doit être constitué d'au moins 80% de méthodes de moins de 10 lignes.

4.4.10. Le code source du client lourd en TypeScript doit suivre les recommandations de hiérarchie de fichiers d'Angular.

4.4.11. Le code source en Kotlin doit suivre les recommandations de hiérarchie de fichiers d'Android Studio.

4.4.12. Le code source de chaque classe doit suivre le « Single Responsibility Principle », SRP.

4.4.13. Le code source de chaque fonction doit avoir un seul but.

4.5. Contraintes de conception

4.5.1. Le système du client léger doit être développé avec Android Studio et Kotlin.

4.5.2. Le système du client lourd doit être développé avec le MEAN stack.

4.5.3. Le système du client lourd doit être déployé avec Electron.

4.5.4. Le client lourd doit communiquer avec le serveur par HTTPS ou Socket.IO.

4.5.5. Le client léger doit communiquer avec le serveur par HTTPS ou Socket.IO.

4.6. Sécurité

4.6.1. Le client léger doit envoyer le mot de passe d'un utilisateur au serveur en le passant par au moins une fonction de hachage.

4.6.2. Le client lourd doit envoyer le mot de passe d'un utilisateur au serveur en le passant par au moins une fonction de hachage.

4.6.3. La base de données doit sauvegarder le mot de passe crypté d'un utilisateur.

4.6.4. Le système doit uniquement transmettre le mot de passe une fois crypté.

4.6.5. Le système doit utiliser des protocoles de communication sécurisés.