

**Projet «Fais moi un Dessin»**  
**Spécifications des requis du système (SRS)**

**Version 1.1**

## Historique des révisions

Date	Version	Description	Auteur
2020-09-09	1.0	Écriture de l'introduction	Gloria Aubierge Sohou
2020-09-09	1.0	Rédaction initiale des exigences	Tous
2020-09-14	1.1	Consolidation de la typographie du document, ajout de spécifications.	Fabrice Ndui
2020-09-16	1.1	Révision et retouche	Tous
2020-09-16	1.1	Correction d'orthographe	Sophie Dorval
2020-09-17	1.1	Révision finale	Tous
2020-10-01	1.1	Révision des exigences fonctionnelles	Gloria Aubierge Sohou
2020-10-02	1.1	Révision des exigences non fonctionnelles et de la numérotation	Gloria Aubierge Sohou
2020-10-02	1.1	Révision finale	Tous

# Table des matières

<b>1. Introduction</b>	<b>5</b>
1.1. But	5
1.2. Définitions, acronymes et abréviations	5
1.3. Vue d'ensemble du document	5
<b>2. Description globale</b>	<b>5</b>
2.1. Caractéristiques des usagers	5
2.2. Interfaces	5
2.2.1. Interfaces usagers	5
2.2.2. Interfaces matérielles	6
2.2.3. Interfaces logicielles	6
2.2.4. Interfaces de communication	6
2.3. Contraintes générales	6
2.4. Hypothèses et dépendances	6
<b>3. Exigences fonctionnelles</b>	<b>7</b>
<b>3.1. Client lourd</b>	<b>7</b>
3.1.1. Clavardage - Intégration	7
3.1.2. Canaux de discussion	7
3.1.3. Profil de l'utilisateur et historique	8
3.1.4. Modes de jeu	8
3.1.5. Création d'une paire mot-image	11
3.1.6. Personnalité des joueurs virtuels	12
3.1.7. Effets visuels et sonores	13
3.1.8. Tutoriel	13
3.1.9. Leaderboard	13
3.1.10. Vote to kick	14
3.1.11. Internationalisation	14
3.1.12. Défier un utilisateur	14
3.1.13. Thème personnalisé	15
3.1.14. Inviter quelqu'un dans l'application	15
3.1.15. Ajouter un utilisateur comme ami	15
<b>3.2. Client léger</b>	<b>15</b>
3.2.1. Clavardage - Intégration	15
3.2.2. Canaux de discussion	16
3.2.3. Profil de l'utilisateur et historique	16

3.2.4. Modes de jeu	16
3.2.5. Effets visuels et sonores	16
3.2.6. Tutoriel	16
3.2.7. Leaderboard	16
3.2.8. Vote to kick	16
3.2.9. Internationalisation	17
3.2.10. Défier un utilisateur	17
3.2.11. Thème personnalisé	17
3.2.12. Inviter quelqu'un dans l'application	17
3.2.13. Ajouter un utilisateur comme ami	17
<b>4. Exigences non-fonctionnelles</b>	<b>17</b>
4.1. Utilisabilité	17
4.2. Fiabilité	17
4.3. Performance	18
4.4. Maintenabilité	18
4.5. Contraintes de conception	18
4.6. Sécurité	19
<b>ANNEXE A: Glossaire</b>	<b>19</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

L'annexe A à la fin de ce document définit tous les mots, acronymes et abréviations utilisés et nécessaires à la compréhension du SRS.

Les mots soulignés dans le document sont ceux décrits dans l'annexe A.

### 1.3. Vue d'ensemble du document

Ce document est organisé en trois grandes parties.

Tout d'abord, la section 2 présente une description globale du document. On y explique en quoi consiste le logiciel «Fais moi Dessin». Il s'agit en effet de définir les caractéristiques des potentiels usagers, les différentes interfaces usagers, les contraintes puis les hypothèses et dépendances du logiciel.

Ensuite, la section 3 porte sur les exigences fonctionnelles du logiciel c'est-à-dire les fonctionnalités du logiciel «Fais moi Dessin».

Enfin, la section 4 présente les exigences non fonctionnelles du logiciel «Fais moi Dessin». Il s'agit des qualités du logiciel dont l'utilisabilité et la performance.

Les exigences décrites dans ce document proviennent en partie du document de vision et du complément pédagogique fournis par l'autorité contractante.

## 2. Description globale

«Fais moi Dessin» est un logiciel de dessin avec plusieurs modes de jeux à savoir: le mode multijoueur compétitif, le mode solo et le mode multijoueur coopératif. Le logiciel est multi plateforme, utilisable sur un client lourd utilisant Windows 10 et également sur un client léger Android. Pour établir la communication entre ces différents clients, le système utilise un serveur NodeJS.

### 2.1. Caractéristiques des usagers

Les utilisateurs visés par cette application sont principalement des ingénieurs logiciels. Les ingénieurs logiciels ont une bonne connaissance de la technologie et sont familiers avec des applications similaires à la nôtre. Les ingénieurs logiciels ont des aptitudes élevées à manipuler les interfaces matérielles utilisées pour notre application et auront une facilité à apprendre à utiliser notre application.

Néanmoins, ces utilisateurs n'ont pas besoin d'avoir de formation ou des connaissances particulières pour utiliser le logiciel. Notre logiciel est un jeu, dont l'interface usager est simple pour que l'utilisateur n'ait pas d'opérations complexes à faire. Ceci rend le jeu accessible à tous et plaisant à utiliser.

### 2.2. Interfaces

#### 2.2.1. Interfaces usagers

L'interface du client lourd est développée en HTML avec Angular. Le client léger est une application mobile android codée en Kotlin.

Les interfaces des deux clients sont assez cohérentes afin que les utilisateurs ne soient pas perdus lorsqu'il changent de client.

Il y a aussi l'interface du serveur qui répond à des requêtes venant des deux clients soit le client lourd et léger ainsi que de permettre la communication entre ces derniers. Le serveur est développé en Node.js et utilisera comme langage de requêtes GraphQL.

### *2.2.2. Interfaces matérielles*

Les utilisateurs utilisent le clavier et la souris ou autre interface d'entrée tel qu'un écran tactile pour interagir avec l'application. Pour le client léger, les utilisateurs utilisent l'interface tactile multi-touch d'une tablette. Les deux clients utilisent le canal de son configuré par leur appareils. Donc, si l'utilisateur branche des écouteurs ou haut-parleurs externe, ceux-ci vont être configuré sur l'appareil et seront utilisé pour nos sons.

### *2.2.3. Interfaces logicielles*

Pour le client léger, le langage Kotlin sera utilisé avec Android Studio comme environnement de développement. La tablette intelligente android a une version minimum Android 9.0 et un sdk supérieur à 28.

Le client lourd sera développé en Typescript avec Angular sur le framework Electron. Pour finir le serveur sera développé également en Typescript sur le framework Apollo-server en utilisant GraphQL pour décorer les entités et les servir. Les données seront persistées avec PostgreSQL desservi directement sur Heroku. Visual Studio sera le principal environnement de développement pour le client Lourd et le serveur.

La librairie SVG sera utilisée pour la manipulation des formes vectorielles. Du côté du client-léger, l'API native Android de Canvas est utilisée pour les dessins. L'API QuickDraw sera utilisée pour accéder à la banque de dessin de Google.

L'outil Potrace permet de transformer des images bitmap en SVG.

### *2.2.4. Interfaces de communication*

Le système communique grâce à un serveur NodeJS utilisant le langage GraphQL afin de faire des requêtes cherchant seulement l'information nécessaire. Les souscriptions de GraphQL qui implémentent des Websockets, seront utilisées pour permettre une communication en temps réel. Une communication se fera également avec l'API de QuickDraw pour récupérer les images de dessins avec une requête par le serveur.

## **2.3. Contraintes générales**

Les utilisateurs doivent avoir assez d'espace pour le téléchargement de notre application. L'interface doit être visuellement attrayante et facile à utiliser.

Les clients (léger et lourd) doivent être consistants entre eux au niveau des fonctionnalités importantes et de l'interface.

Un dessin doit être identique pour tous les utilisateurs quelque soit le client afin d'assurer la cohérence.

Il y a huit joueurs maximum dans une partie, le nombre maximal de parties en cours est de 5, donc le maximum de connexions au total est de quarante à la fois.

## **2.4. Hypothèses et dépendances**

Nous supposons que les utilisateurs ont une bonne connexion internet qui est fluide (débit suffisant) et sans coupure. Nous supposons également que l'utilisateur accède à l'application, soit sur une tablette intelligente android, soit un ordinateur avec un système d'exploitation windows. Il faut aussi avoir assez d'espace pour le téléchargement de notre application.

Pour le client léger, nous supposons que la tablette intelligente android a une version minimum Android 9.0 et un sdk supérieur à 28.

### 3. Exigences fonctionnelles

Les exigences fonctionnelles décrivent les fonctionnalités, c'est-à-dire ce que fait le logiciel «Fais moi Dessin».

#### 3.1. Client lourd

##### 3.1.1. Clavardage - Intégration

###### Exigences essentielles

- 3.1.1. Le système doit offrir à l'utilisateur une boîte de clavardage intégrée à même l'application.
- 3.1.2. Le système doit permettre à l'utilisateur de clavarder avec les autres utilisateurs en ligne en dehors d'une partie.
- 3.1.3. Le système doit rendre la boîte de clavardage visible à l'utilisateur en mode dessin.

###### Exigences souhaitables

- 3.1.4. Le système doit permettre à l'utilisateur de clavarder dans un mode fenêtré.
- 3.1.5. Le système doit permettre à l'utilisateur de détacher la fenêtre de clavardage intégrée à l'application à l'aide d'un bouton.
- 3.1.6. Le système doit masquer la fenêtre de clavardage intégrée à l'application lorsque l'utilisateur appuie sur le bouton servant à le détacher de l'application.
- 3.1.7. Le système doit ouvrir une fenêtre externe lorsque l'utilisateur détache la fenêtre de clavardage intégrée à l'application.
- 3.1.8. Le système doit rendre disponible la boîte de clavardage dans la fenêtre externe en tout temps.
- 3.1.9. Le système doit synchroniser en temps réel la fenêtre externe de clavardage avec l'application principale.
- 3.1.10. Le système doit automatiquement intégrer la fenêtre externe de clavardage dans l'application principale si elle est fermée.

##### 3.1.2. Canaux de discussion

###### Exigences essentielles

- 3.1.2.1. Le système doit permettre à l'utilisateur de participer au canal principal de discussion avec la totalité des utilisateurs.
- 3.1.2.2. Le système doit permettre à l'utilisateur de créer un canal de discussion.
- 3.1.2.3. Le système doit permettre à l'utilisateur de supprimer tout canal de discussion autre que le canal principal.
- 3.1.2.4. Le système doit permettre à l'utilisateur de joindre un canal de discussion.
- 3.1.2.4. Le système doit permettre à l'utilisateur de participer à plus d'un canal de discussion en même temps.
- 3.1.2.5. Le système doit permettre à l'utilisateur de visualiser l'historique de clavardage d'un canal de discussion.
- 3.1.2.6. Le système doit permettre à un joueur de clavarder uniquement avec les autres joueurs lors d'une partie.

### 3.1.3. Profil de l'utilisateur et historique

#### Exigences essentielles

- 3.1.3.1. Le système doit permettre à l'utilisateur de créer un profil.
- 3.1.3.2. Le système doit permettre à l'utilisateur de s'authentifier à son profil.
- 3.1.3.3. Le système doit permettre à l'utilisateur ayant le bon mot de passe de s'authentifier à son profil.
- 3.1.3.4. Le système doit permettre qu'un seul utilisateur à la fois puisse être connecté à un compte.
- 3.1.3.5. Le système doit permettre à l'utilisateur **seul** de voir son nom sur son profil.
- 3.1.3.6. Le système doit permettre à l'utilisateur **seul** de voir son prénom sur son profil.
- 3.1.3.7. Le système doit permettre à l'utilisateur **seul** de voir ses statistiques sur son profil.
  - 3.1.3.7.1. Le système doit permettre à l'utilisateur de voir le nombre de parties jouées .
  - 3.1.3.7.2. Le système doit permettre à l'utilisateur de voir le pourcentage de victoire.
  - 3.1.3.7.3. Le système doit permettre à l'utilisateur de voir le temps moyen d'une partie.
  - 3.1.3.7.4. Le système doit permettre à un utilisateur de voir le temps total passé à jouer.
- 3.1.3.8. Le système doit permettre à l'utilisateur **seul** de voir son historique de connexion sur son profil .
  - 3.1.3.8.1. Le système doit permettre à un utilisateur de voir la date de chaque connexion à l'application.
  - 3.1.3.8.2. Le système doit permettre à un utilisateur de voir l'heure de chaque connexion à l'application.
  - 3.1.3.8.3. Le système doit permettre à un utilisateur de voir la date de chaque déconnexion.
  - 3.1.3.8.4. Le système doit permettre à un utilisateur de voir l'heure de chaque déconnexion.
- 3.1.3.9. Le système doit permettre à l'utilisateur **seul** de voir l'historique des parties que ce dernier a jouées sur son profil.
  - 3.1.3.9.1. Le système doit permettre à l'utilisateur de voir la date de chaque partie jouée.
  - 3.1.3.9.2. Le système doit permettre à l'utilisateur de voir l'heure de chaque partie jouée.
  - 3.1.3.9.3. Le système doit permettre à l'utilisateur de voir le nom des joueurs de chaque partie jouée.
  - 3.1.3.9.4. Le système doit permettre à l'utilisateur de voir le nom des gagnants de chaque partie jouée.
- 3.1.3.10. Le système doit permettre à un utilisateur de voir le pseudo de tout autre utilisateur sur le profil de ce dernier.
- 3.1.3.11. Le système doit permettre à un utilisateur de voir l'avatar de tout autre utilisateur sur le profil de ce dernier.

### 3.1.4. Modes de jeu

#### Exigences essentielles

- 3.1.4.1. Le système doit présenter à l'utilisateur un menu principal une fois qu'il est connecté.
- 3.1.4.2. Le système doit permettre à l'utilisateur de choisir un mode de jeu pour une partie.
- 3.1.4.3. Le système doit permettre à l'utilisateur de choisir la difficulté pour une partie.
- 3.1.4.4. Le système doit permettre à l'utilisateur de créer un groupe.
- 3.1.4.5. Le système doit permettre à l'utilisateur de rejoindre un groupe existant.
- 3.1.4.6. Le système doit automatiquement ajouter l'utilisateur au canal de discussion d'une partie quand il rejoint un groupe.



- 3.1.4.7. Le système doit automatiquement amener l'utilisateur dans une salle d'attente quand il rejoint un groupe.
- 3.1.4.8. Le système doit permettre à l'utilisateur de discuter avec les autres joueurs dans la salle d'attente.
- 3.1.4.9. Le système doit permettre à l'utilisateur de consulter la liste des joueurs humains dans un groupe
- 3.1.4.10. Le système doit permettre à l'utilisateur de consulter la liste des joueurs virtuels dans un groupe.
- 3.1.4.11. Le système doit permettre à l'utilisateur d'ajouter un joueur virtuel à un groupe.
- 3.1.4.12. Le système doit permettre à l'utilisateur d'enlever un joueur virtuel d'un groupe.
- 3.1.4.13. Le système doit définir un minimum de joueurs pour une partie.
- 3.1.4.14. Le système doit rendre disponible un bouton pour démarrer une partie quand le nombre de joueurs minimum est atteint.

### ***Mêlée générale***

#### **Exigences essentielles**

- 3.1.4.15. Le système doit permettre à l'utilisateur de sélectionner le mode mêlée générale à partir du menu principal.
- En mode mêlée générale:
- 3.1.4.16. Le système doit définir le minimum de joueurs pour une partie à deux.
  - 3.1.4.17. Le système doit assigner le rôle de dessinateur à chaque joueur à tour de rôle.
  - 3.1.4.18. Le système doit assigner un mot ou une expression au joueur ayant le rôle de dessinateur.
  - 3.1.4.19. Le système doit permettre à un joueur n'ayant pas le rôle de dessinateur de saisir un mot.
  - 3.1.4.20. Le système doit assigner un certain nombre de points au joueur ayant entré la bonne réponse.
  - 3.1.4.21. Le système doit passer le tour à un autre joueur lorsque le temps maximum alloué pour un dessin est écoulé.
  - 3.1.4.22. Le système doit passer le tour à un autre joueur lorsque tous les joueurs ont deviné le mot.
  - 3.1.4.23. Le système doit assigner au joueur qui dessine, un nombre de points dépendant du nombre de joueurs ayant deviné avec succès.
  - 3.1.4.24. Le système doit permettre à un joueur virtuel d'avoir uniquement le rôle de dessinateur.
  - 3.1.4.25. Le système doit permettre à un joueur de voir en temps réel ce que dessine tout autre joueur.
  - 3.1.4.26. Le système doit permettre à l'utilisateur d'accéder à l'interface de dessin lorsqu'il a le rôle de dessinateur.
  - 3.1.4.27. Le système doit permettre à l'utilisateur d'accéder aux outils de dessin lorsqu'il a le rôle de dessinateur.
    - 3.1.4.27.1. Le système doit permettre à l'utilisateur d'accéder à l'outil crayon
    - 3.1.4.27.2. Le système doit permettre à l'utilisateur d'accéder à l'outil grille.
    - 3.1.4.27.3. Le système doit permettre à l'utilisateur d'accéder à l'outil effacé.
  - 3.1.4.28. Le système doit permettre à l'utilisateur d'annuler un dessin.
  - 3.1.4.29. Le système doit permettre à l'utilisateur de refaire un dessin précédemment annulé.

## **Exigences souhaitables**

### ***Sprint solo***

3.1.4.30. Le système doit permettre à l'utilisateur de sélectionner le mode sprint solo à partir du menu principal.

#### **En mode sprint solo:**

3.1.4.31. Le système doit définir un nombre maximum d'essais pour deviner un mot pour chaque utilisateur..

3.1.4.32. Le système doit permettre à l'utilisateur de voir à l'écran le temps restant à tout moment dans une partie.

3.1.4.33. Le système doit permettre à l'utilisateur de voir à l'écran son nombre d'essais restants pour deviner un mot.

3.1.4.34. Le système doit permettre à l'utilisateur de voir à l'écran son score actuel à tout moment dans la partie.

3.1.4.35. Le système doit ajouter un temps bonus au compte à rebours du joueur pour chaque bonne réponse.

3.1.4.36. Le système doit ajouter un point au score de l'utilisateur pour chaque bonne réponse.

3.1.4.37. Le système doit afficher l'image suivante si le joueur ne réussit pas à deviner le mot dans la limite d'essais allouée.

### ***Sprint coopératif***

3.1.4.38. Le système doit permettre à l'utilisateur de sélectionner le mode sprint coopératif à partir du menu principal.

#### **Si l'utilisateur sélectionne le mode sprint coopératif :**

3.1.4.39. Le système doit permettre à l'utilisateur de se joindre à trois autres joueurs humains à partir du menu principal.

3.1.4.40. Le système doit permettre au groupe un maximum collectif d'essais en fonction de la difficulté définie à la création du jeu.

3.1.4.41. Le système doit afficher à l'écran une image à chaque tour.

3.1.4.42. Le système doit permettre à l'utilisateur de saisir un mot ou une expression.

3.1.4.43. Le système doit permettre à l'utilisateur de voir à l'écran le temps restant à tout moment dans la partie.

3.1.4.44. Le système doit permettre à l'utilisateur de voir à l'écran le nombre d'essais restants à tout moment dans la partie.

3.1.4.45. Le système doit permettre à l'utilisateur de voir à l'écran le score actuel à tout moment dans la partie.

3.1.4.46. Le système doit accorder du temps bonus dès qu'un joueur du groupe entre la bonne réponse.

3.1.4.47. Le système doit accorder un point dès qu'un joueur du groupe entre la bonne réponse.

3.1.4.48. Le système doit passer à l'image suivante si aucun joueur du groupe ne réussit à deviner le mot ou l'expression.

## **3.1.5. Création d'une paire mot-image**

### **Exigences essentielles**

3.1.5.1. Le système doit offrir à l'utilisateur l'option de créer une paire mot image à partir du menu principal.

#### ***Manuelle I***

3.1.5.2. Le système doit offrir un interface permettant à l'utilisateur d'entrer son choix de mot ou d'expression.

- 3.1.5.3. Le système doit offrir une interface permettant à l'utilisateur de fournir au minimum un indice.
- 3.1.5.4. Le système doit permettre à l'utilisateur de sélectionner un niveau de difficulté associé à son image.
- 3.1.5.5. Le système doit permettre à l'utilisateur de dessiner le mot ou l'expression choisie dans une zone de dessin.
- 3.1.5.6. Le système doit permettre à l'utilisateur de prendre le temps nécessaire pour dessiner son mot ou expression.
- 3.1.5.7. Le système doit offrir à l'utilisateur un interface de dessin ayant les mêmes fonctionnalités que lors d'une partie.
- 3.1.5.8. Le système doit pouvoir dessiner l'image comme s'il s'agissait du joueur virtuel en accéléré pour donner un aperçu à l'utilisateur.
- 3.1.5.9. Le système doit dessiner chaque trait d'un dessin dans le même ordre que lors de sa création.
- 3.1.5.10. Le système doit définir la vitesse de dessin d'un joueur virtuel en fonction du niveau de difficulté associé à l'image.
- 3.1.5.11. Le système doit définir le temps alloué à un joueur pour deviner en fonction du niveau de difficulté associé à l'image.
- 3.1.5.12. Le système doit définir le nombre de tentatives alloué à un joueur pour deviner en fonction du niveau de difficulté associé à l'image.

## **Exigences souhaitables**

### ***Manuelle II***

- 3.1.5.13. Le système doit offrir une interface permettant à l'utilisateur de sélectionner un mode de dessin des joueurs virtuels.
- 3.1.5.14. Le système doit permettre à l'utilisateur de sélectionner le mode de dessin classique.
- 3.1.5.15. Le système doit dessiner chaque trait du dessin dans le même ordre que lors de la création du dessin dans le mode classique.
- 3.1.5.17. Le système doit permettre à l'utilisateur de sélectionner le mode de dessin aléatoire.
- 3.1.5.18. Le système doit afficher chaque élément individuel dans un ordre aléatoire dans le mode aléatoire.
- 3.1.5.19. Le système doit afficher les traits en suivant deux différentes séquences lors de deux différentes parties avec la même image dans le mode aléatoire.
- 3.1.5.20. Le système doit permettre à l'utilisateur de sélectionner le mode de dessin panoramique.
- 3.1.5.21. Le système doit rendre disponible la direction de gauche à droite en mode panoramique.
- 3.1.5.22. Le système doit rendre disponible la direction haut à bas en mode panoramique.
- 3.1.5.23. Le système doit rendre disponible la direction bas à haut en mode panoramique.
- 3.1.5.24. Le système doit rendre disponible la direction droite à gauche en mode panoramique.
- 3.1.5.25. Le système doit afficher les traits dans l'ordre de leur position sur les axes cartésiens dans le mode panoramique.
- 3.1.5.26. Le système doit permettre à l'utilisateur de sélectionner le mode de dessin centré.

3.1.5.27. Le système doit afficher les traits dans l'ordre de leur distance au centre de l'image dans le mode centré.

3.1.5.28. Le système doit rendre disponible le sens intérieur vers extérieur en mode centré.

3.1.5.29. Le système doit rendre disponible le sens extérieur vers l'intérieur en mode centré.

#### ***Assistée I***

3.1.5.30. Le système doit permettre à l'utilisateur d'importer une image de son ordinateur.

3.1.5.31. Le système doit faire passer l'image importée par un engin de conversion.

3.1.5.32. Le système doit offrir à l'utilisateur un interface permettant de spécifier manuellement les paramètres de l'engin de conversion.

3.1.5.33. Le système doit permettre à l'utilisateur de sélectionner l'ordre de séquence d'apparition des zones tel que définie dans la section *manuelle II*.

3.1.5.34. Le système doit rendre indisponible le mode de dessin classique.

#### ***Assistée II***

3.1.5.35. Le système doit proposer à l'utilisateur des mots et dessins à partir de la banque de dessins Quick Draw» de Google.

3.1.5.36. Le système doit offrir à l'utilisateur un bouton permettant de changer le mot et le dessin jusqu'à ce que ce dernier en trouve un le satisfaisant.

3.1.5.37. Le système doit permettre à l'utilisateur de fournir le mode de dessin.

3.1.5.38. Le système doit permettre à l'utilisateur de fournir un indice.

#### ***Assistée III***

3.1.5.39. Le système doit permettre à l'utilisateur de rechercher une image sur internet avec le mot ou l'expression choisie.

3.1.5.40. Le système doit proposer à l'utilisateur une série d'images suite à une recherche.

3.1.5.41. Le système doit permettre à l'utilisateur de choisir une image à partir des résultats d'une recherche.

3.1.5.42. Le système doit permettre à l'utilisateur de demander plus de résultats si aucune image des résultats de la recherche n'est satisfaisante.

3.1.5.43. Le système doit passer directement de l'image à l'engin de conversion décrit dans *Assistée I*, une fois que l'image désirée est trouvée.

### **3.1.6. Personnalité des joueurs virtuels**

#### **Exigences essentielles**

3.1.6.1. Le système doit publier au nom de chaque joueur virtuel, un message au début de la partie de chaque tour dans le canal de discussion de la partie actuelle .

3.1.6.2. Le système doit publier un message au nom de chaque joueur virtuel, à la fin de chaque tour dans le canal de discussion de la partie actuelle.

3.1.6.3. Le système doit publier au nom d'un joueur virtuel un indice définie à la création de la paire mot-image lorsqu'un membre du groupe de ce dernier demande un indice,.

3.1.6.4. Le système doit publier au nom de chaque joueur virtuel, un message générique pour féliciter le gagnant d'une partie.

#### **Exigences souhaitables**

3.1.6.5. Le système doit au nom de chaque joueur virtuel, garder en mémoire les anciennes parties jouées.

3.1.6.6. Le système doit envoyer un message personnalisé au nom de chaque joueur virtuel à un autre joueur selon les statistiques de ce dernier.

3.1.6.7. Le système doit envoyer un message personnalisé en fonction de sa personnalité au nom de chaque joueur virtuel.

### **3.1.7. Effets visuels et sonores**

#### **Exigences essentielles**

3.1.7.1. Le système doit afficher à l'écran des confettis à la fin d'une partie.

3.1.7.1. Le système doit afficher à l'écran le nom du gagnant à la fin d'une partie.

3.1.7.2. Le système doit émettre un bruit sonore lorsqu'un utilisateur trouve la bonne réponse.

3.1.7.3. Le système doit émettre un bruit sonore lorsqu'un utilisateur entre une mauvaise réponse.

3.1.7.4. Le système doit émettre un bruit sonore stressant lorsqu'il ne reste plus beaucoup de temps à la partie.

### **3.1.8. Tutoriel**

#### **Exigences souhaitables**

3.1.8.1. Le système doit afficher automatiquement un tutoriel lorsque l'utilisateur entre pour la première fois dans l'application.

3.1.8.2. Le système doit permettre à l'utilisateur de réaliser une action précise pour progresser à travers le tutoriel.

3.1.8.3. Le système doit offrir à l'utilisateur un bouton lui permettant de visionner le tutoriel à tout moment.

### **3.1.9. Leaderboard**

#### **Exigences souhaitables**

3.1.9.1. Le système doit afficher à l'écran les résultats à la fin de chaque match .

3.1.9.2. Le système doit proposer à l'utilisateur l'option d'afficher le leaderboard à la fin de chaque match.

3.1.9.3. Le système doit permettre à l'utilisateur de voir dans le leaderboard, les joueurs avec les plus grands scores dans chaque mode.

3.1.9.3.1. Le système doit permettre à l'utilisateur de voir dans le leaderboard, les joueurs avec les plus grands scores cumulatifs dans chaque mode.

3.1.9.3.2. Le système doit permettre à l'utilisateur de voir les joueurs avec le plus grand nombre de mêlées générales gagnées.

3.1.9.4. Le système doit permettre à l'utilisateur d'accéder en tout temps au leaderboard à l'aide d'un bouton.

### 3.1.10. Vote to kick

#### Exigences souhaitables

- 3.1.10.1. Le système doit permettre à un joueur d'initier un vote afin de retirer un autre joueur dans une partie de plus de deux joueurs.
- 3.1.10.2. Le système doit permettre à un joueur de voter afin de retirer un joueur du match.
- 3.1.10.3. Le système doit offrir lors d'un vote, l'option « pour le retrait d'un joueur » à chaque votant.
- 3.1.10.4. Le système doit offrir lors d'un vote, l'option « contre le retrait d'un joueur » à chaque votant.
- 3.1.10.5. Le système doit retirer un joueur si plus de la moitié des votants votent pour son retrait.
- 3.1.10.6. Le système doit permettre à un joueur de continuer la partie si plus de la moitié des votants votent contre son retrait.
- 3.1.10.7. Le système doit afficher un message au joueur retiré lui indiquant qu'il a été retiré de la partie.
- 3.1.10.8. Le système doit permettre à un joueur d'initier plus d'un vote dans une partie.
- 3.1.10.9. Le système doit permettre à un joueur de participer à tout vote initié dans une partie .

### 3.1.11. Internationalisation

#### Exigences souhaitables

- 3.1.11.1. Le système doit permettre à un utilisateur de sélectionner la langue de son choix à sa première ouverture de l'application.
- 3.1.11.2. Le système doit permettre à un utilisateur de modifier la langue de son choix en tout temps dans son profil.
- 3.1.11.3. Le système doit associer la langue au profil de chaque utilisateur.
- 3.1.11.4. Le système doit choisir la langue du système par défaut.
- 3.1.11.5. Si un utilisateur choisit une autre langue, le système doit choisir cette langue au détriment de la langue du système à chaque ouverture de l'application avec ce profil.
- 3.1.11.6. Le système doit offrir une interface permettant à l'utilisateur de choisir une langue lors de la création d'une partie.
- 3.1.11.7. Le système doit permettre à un joueur de se joindre à n'importe quelle partie quelle que soit la langue de son profil.

### 3.1.12. Défier un utilisateur

#### Exigences souhaitables

- 3.1.12.1. Le système doit permettre à un joueur d'envoyer une invitation à un défi à tout autre utilisateur.
- 3.1.12.2. Le système doit permettre à un utilisateur d'envoyer une invitation à un défi à un autre utilisateur en cliquant sur son nom ou son avatar dans le canal de discussion principal.
- 3.1.12.3. Le système doit informer un joueur invité à un défi de l'invitation.
- 3.1.12.4. Le système doit offrir à un joueur ayant reçu une invitation à un défi l'option d'accepter l'invitation.
- 3.1.12.5. Le système doit offrir à un joueur ayant reçu une invitation à un défi l'option de décliner l'invitation.

- 3.1.12.6. Le système doit automatiquement ajouter un joueur invité au lobby du jeu s'il accepte l'invitation avant que la partie ne commence.
- 3.1.12.7. Le système doit notifier au joueur qu'il ne pourra plus faire ce défi, s'il accepte le défi après que la partie ait commencé.

### 3.1.13. Thème personnalisé

#### Exigences souhaitables

- 3.1.13.1. Le système doit permettre à l'utilisateur de sélectionner un thème à la première ouverture de l'application.
- 3.1.13.2. Le système doit permettre à l'utilisateur de modifier en tout temps le thème dans les paramètres de l'application.
- 3.1.13.3. Le système doit proposer cinq thèmes différents à l'utilisateur.
- 3.1.13.4. Le système doit choisir par défaut le Dark mode (couleur sombre) .
- 3.1.13.5. Le système doit associer le thème au profil de chaque utilisateur.

### 3.1.14. Inviter quelqu'un dans l'application

- 3.1.14.1. Le système doit permettre à un utilisateur d'inviter une personne extérieure dans l'application.
- 3.1.14.2. Le système doit permettre à un utilisateur d'envoyer un email d'invitation à n'importe qui à partir de son profil.
- 3.1.14.3. Le système doit indiquer dans un email d'invitation un lien de téléchargement sur le serveur de l'application.

### 3.1.15. Ajouter un utilisateur comme ami

- 3.1.15.1. Le système doit permettre à un utilisateur d'avoir une liste d'amis.
- 3.1.15.2. Le système doit permettre à un utilisateur d'ajouter un autre utilisateur dans sa liste d'amis.
- 3.1.15.3. Le système doit permettre à un utilisateur de retirer un autre utilisateur dans sa liste d'amis.
- 3.1.15.4. Le système doit afficher en premier dans la liste des joueurs en ligne, le nom d'un ami .
- 3.1.15.5. Le système doit afficher le nom d'un ami d'un utilisateur dans une couleur différente.

## 3.2. Client léger

### 3.2.1. Clavardage - Intégration

#### Exigences essentielles

- 3.2.1.1. Le système doit offrir à l'utilisateur une fenêtre de clavardage intégrée à même l'application.
- 3.2.1.2. Le système doit permettre à l'utilisateur de clavarder en dehors d'une partie avec les autres utilisateurs en ligne.
- 3.2.1.3. Le système doit envoyer une notification lors de la réception d'un nouveau message.
- 3.2.1.4. Le système doit afficher un indicateur visuel indiquant la présence de messages non lus lorsque le

clavardage n'est pas ouvert.

3.2.1.6. Si un seul canal de discussion existe, l'indicateur doit disparaître dès l'ouverture du clavardage.

3.2.1.7. Le système doit jouer un effet sonore lors de la réception d'un nouveau message.

### **3.2.2. Canaux de discussion**

Les exigences sont les mêmes que celles présentées à la section 3.2 du client lourd.

### **3.2.3. Profil de l'utilisateur et historique**

Les exigences sont les mêmes que celles présentées à la section 3.3 du client lourd.

### **3.2.4. Modes de jeu**

Les exigences sont les mêmes que celles présentées à la section 3.4 du client lourd.

### **3.2.5. Effets visuels et sonores**

#### **Exigences essentielles**

Les exigences sont les mêmes que celles présentées à la section 3.7 du client lourd.

### **3.2.6. Tutoriel**

#### **Exigences essentielles**

3.2.6.1. Le système doit afficher à l'utilisation un tutoriel lors de sa première utilisation de l'application

3.2.6.2. Le système doit offrir à l'utilisateur un bouton lui permettant de visionner le tutoriel à tout moment.

3.2.6.3. Le tutoriel doit uniquement demander à l'utilisateur de cliquer sur un bouton « Suivant » pour progresser à travers le tutoriel.

#### **Exigences souhaitables**

Les exigences sont les mêmes que celles présentées à la section 3.8 du client lourd.

### **3.2.7. Leaderboard**

#### **Exigences souhaitables**

Les exigences sont les mêmes que celles présentées à la section 3.9 du client lourd.

### **3.2.8. Vote to kick**

#### **Exigences souhaitables**

Les exigences sont les mêmes que celles présentées à la section 3.x du client lourd.



### 3.2.9. Internationalisation

#### Exigences souhaitables

Les exigences sont les mêmes que celles présentées à la section 3.x du client lourd.

### 3.2.10. Défier un utilisateur

#### Exigences souhaitables

Les exigences sont les mêmes que celles présentées à la section 3.x du client lourd.

### 3.2.11. Thème personnalisé

#### Exigences souhaitables

Les exigences sont les mêmes que celles présentées à la section 3.x du client lourd.

### 3.2.12. Inviter quelqu'un dans l'application

#### Exigences souhaitables

Les exigences sont les mêmes que celles présentées à la section 3.x du client lourd.

### 3.2.13. Ajouter un utilisateur comme ami

#### Exigences souhaitables

Les exigences sont les mêmes que celles présentées à la section 3.x du client lourd.

## 4. Exigences non-fonctionnelles

Dans cette section, on décrit les qualités du logiciel «Fais moi un Dessin»

### 4.1. Utilisabilité

- 4.1.1. Le système doit permettre à l'utilisateur de créer un profil en deux minutes.
- 4.1.2. Le système doit permettre à l'utilisateur de se connecter à son profil en moins de 20 secondes.
- 4.1.3 Le système doit permettre à l'utilisateur de configurer une partie en moins de cinq minutes.
- 4.1.4. Le système doit permettre à l'utilisateur de joindre une partie en moins de 20 secondes.
- 4.1.5. Le système doit proposer à l'utilisateur des thèmes cohérents.

### 4.2. Fiabilité

- 4.2.1. Le système doit être accessible 99% du temps.
- 4.2.2. Le système doit être défaillant au plus cinq fois par an.
- 4.2.3. Le système doit être réparé en un temps moyen de douze heures lors d'une panne.

### 4.3. Performance

- 4.3.1. Le système doit fonctionner de façon fluide sur le réseau local de Polytechnique Montréal.
- 4.3.2. Le système doit permettre à l'utilisateur d'accéder au lobby d'une partie en 5 secondes au plus après avoir rejoint la partie.
- 4.3.3. Le système doit permettre à l'utilisateur de recevoir les messages des autres joueurs avec un temps de latence inférieur à 5 secondes.
- 4.3.4. Le système doit permettre à l'utilisateur de voir à l'écran les traits d'un dessin avec un temps de latence inférieur à 1 seconde.
- 4.3.5. Le système doit permettre à l'utilisateur de recevoir une notification pour une invitation à un défi en moins de 3 secondes après son envoi.
- 4.3.6. Le système doit permettre à l'utilisateur de changer d'outil de dessin avec un temps de latence inférieur à 250 ms.
- 4.3.7. Le système doit émettre un effet visuel lorsque l'utilisateur gagne une partie en moins de 250 ms.
- 4.3.9. Le système doit afficher l'interface d'une partie au plus une seconde après sa création.
- 4.3.10. Le système doit permettre à l'utilisateur de saisir un indice ayant une taille inférieure à 20 caractères.
- 4.3.11. Le système doit émettre un son lorsqu'une personne devine correctement le dessin en moins de 250 ms.
- 4.3.12. Le système doit changer la couleur du chronomètre lorsqu'il reste 10 secondes pour un tour en moins de 250 ms.
- 4.3.13. Le serveur doit pouvoir exécuter une requête en un temps moyen inférieur à une seconde.
- 4.3.14. Le serveur doit effectuer une recherche dans la base de données en moins de deux secondes.
- 4.3.15. Le serveur doit effectuer une recherche d'image à partir de Quick Draw en cinq secondes au plus.
- 4.3.16. Le serveur doit effectuer une recherche d'image à partir de Google Images en cinq secondes au plus.
- 4.3.17. Le serveur doit être capable de supporter cinq parties simultanément.
- 4.3.18. Le serveur doit répondre aux requêtes en moins d'une seconde lorsque l'utilisateur tente de deviner un dessin.
- 4.3.19. Le serveur doit valider le nom d'utilisateur et le mot de passe d'un utilisateur voulant se connecter à son profil en une seconde au plus.
- 4.3.20. Les images envoyées au serveur doivent avoir une taille inférieure à 25 Mo.
- 4.3.21. Le taux de rafraîchissement doit être de 30 images par seconde pour le client lourd.
- 4.3.22. Le taux de rafraîchissement doit être de 20 images par seconde pour le client léger.
- 4.3.23. La taille de l'historique des messages au niveau du serveur doit être inférieure à 4 Go.

### 4.4. Maintenabilité

- 4.4.1. Le code doit être documenté.
- 4.4.2. L'anglais doit être utilisé pour les commentaires.

- 4.4.3. L'anglais doit être utilisé pour le nom d'un commit.
- 4.4.4. L'anglais doit être utilisé pour le code.
- 4.4.5. Le code du client lourd doit respecter les normes et standards de programmation du document "C# Coding Standards for .NET".
- 4.4.6. Le code du client léger doit respecter les normes et standards de programmation du document 'Java Coding Standards and Guidelines for CS4233'.
- 4.4.7. Le code du serveur doit respecter les normes et standards de programmation du site web '<https://www.typescriptlang.org/docs/handbook/declaration-files/do-s-and-don-ts.html>'.
- 4.4.8. Les algorithmes doivent être rédigés de façon qu'il y ait le moins possible de duplication de code.

## 4.5. Contraintes de conception

- 4.5.1. Le langage de programmation utilisé pour réaliser le client lourd doit être Typescript.
- 4.5.2. Le langage de programmation utilisé pour réaliser le client léger doit être Kotlin.
- 4.5.3. Le serveur doit être développé avec Node.JS.
- 4.5.4. GraphQL doit être utilisé dans le serveur.
- 4.5.5. Apollo Client doit être utilisé dans les deux clients (lourd et léger) pour gérer la communication avec le serveur
- 4.5.6. Le client lourd doit être basé sur le cadre logiciel Electron
- 4.5.7. L'API de Quick Draw doit être utilisée.
- 4.5.8. L'exécutable de Potrace doit être utilisé pour convertir l'image en type SVG.
- 4.5.9. GraphQL subscription doit être utilisé pour gérer les communications en temps réel à travers les Websocket.
- 4.5.10. La base de données doit être Postgresql.
- 4.5.11. Le serveur doit être hébergé sur Heroku.
- 4.5.12. Le code du client lourd doit être rédigé sur Visual Studio 2017 ou 2019.
- 4.5.13. Le code du client léger doit être rédigé sur Android Studio.
- 4.5.14. L'API Espresso doit être utilisé pour tester l'interface utilisateur du client léger.
- 4.5.15. La version du client léger doit être compatible avec la version 9.0 Pie et les versions supérieures de Android.
- 4.5.16. Apollo-server Test Suite doit être utilisé pour tester GraphQL.
- 4.5.16. Karma doit être utilisé pour tester Angular.

## 4.6. Sécurité

- 4.6.1. Le système doit rendre inaccessible chaque mot de passe d'un utilisateur aux autres utilisateurs.
- 4.6.2. Le système doit permettre à l'utilisateur de choisir un mot de passe alphanumérique sans caractère particulier.
- 4.6.3. Le système doit permettre à l'utilisateur de choisir un mot de passe entre 6 et 20 caractères.
- 4.6.4. Le système doit permettre à l'utilisateur de choisir un nom d'utilisateur ayant entre 6 et 20 caractères.
- 4.6.5. Le système doit permettre à l'utilisateur de se déconnecter en tout temps.

4.6.6. Le système doit permettre à l'utilisateur de choisir un nom d'utilisateur unique.

## ANNEXE A: Glossaire

Terme	
SRS	Software Requirements Specifications ou requis du système.
Android	Noyau d'exploitation pour mobile.
Utilisateur	Toute personne utilisant l'application.
Mode dessin	Mode permettant à un joueur de dessin, offrant la surface de dessin et les outils de dessin.
Client lourd	Le logiciel qui propose des fonctionnalités complexes avec un traitement autonome. Dans notre cas, il s'agit d'un ordinateur.
Client léger	Le logiciel qui n'a presque pas de logique d'application. Dans notre cas, il s'agit d'un téléphone Android..
Joueur	Un utilisateur qui participe à un jeu.
Joueur virtuel	Le serveur est un participant au jeu. Les joueurs virtuel sont gérés par le serveur.
Image	Une représentation visuelle de quelque chose.
Dessin	Technique de représentation visuelle sur un support.
Groupe	Ensemble d'utilisateurs de l'application.
Partie	Temps écoulé entre le moment où les joueurs commencent à jouer à un jeu et le moment où ils le délaissent.
Zone de dessin ou surface de dessin	Surface où un joueur dessine.
Grille	Une grille affichée en arrière-plan de la surface de dessin.
Crayon	Le crayon fait partie des outils de dessin permettant de tracer la zone de dessin.
Efface	L'efface fait partie des outils de dessin permettant d'effacer des dessins qu'il a préalablement faits sur la zone de dessin.
Menu principal	Il est le point de départ pour commencer le jeu
Canal principal de discussion	Canal de discussion incluant la totalité des utilisateurs, présent par défaut.
Thème	Définit les couleurs de base de l'application. Ex: Dark mode(couleur sombre)
Invitation à un défi	Un utilisateur envoie un message à un autre utilisateur pour le défier à

	une partie
Lobby du jeu	Avant de commencer une partie, les joueurs sont placés au fur et à mesure dans une salle d'attente. Ils peuvent discuter entre eux.
Personne extérieure	Toute personne n'ayant pas un profil dans l'application "Fais-moi un dessin".
Email d'invitation	Courrier invitant une personne à joindre l'application.
API	Une interface de programmation d'application.
Paire mot-image	L'image associée au mot
Tour	Un tour fini lorsque le temps alloué pour deviner un mot est écoulé
Dessinateur	L'utilisateur qui dessine le mot secret.
Framework	Un framework désigne un ensemble cohérent de composants logiciels structurels, qui sert à créer l'architecture d'un logiciel.
SVG	Scalable Vector Graphics est un format de données conçu pour décrire des ensembles de graphiques vectoriels et basé sur XML
GraphQL	GraphQL est un langage de requêtes et un environnement d'exécution.
Node.js	Node.js est une plateforme logicielle libre en JavaScript orientée vers les applications réseau événementielles
Angular	Angular est un Framework open source écrit en JavaScript qui permet la création d'applications Web
HTML	HyperText Markup Language est le langage de balisage conçu pour représenter les pages web.
Salle d'attente /Lobby du jeu	Lorsqu'une partie est à pleine capacité, l'utilisateur attend dans une salle d'attente jusqu'à ce qu'une place se libère dans la partie
Canal de discussion d'une partie	Le canal de discussion incluant seulement les utilisateurs réels et virtuels présents dans la partie.