

---

**Équipe 209**

---

**PolyDiff**  
**Spécifications des requis du système (SRS)**

**Version 1.0**

## Historique des révisions

Date	Version	Description	Auteur
2024-01-14	0.1	Première ébauche de la partie 1	Mohamed Chaoui
2024-01-18	0.2	Première ébauche de la partie 2	Mohamed Chaoui
2024-01-21	0.3	Exigences fonctionnelles partie 1	Zied Kaabi
2024-01-23	0.4	Première ébauche des parties 4.1, 4.2, 4.3 et 4.5	Mohamed Chaoui
2024-01-24	0.5	Ajout de définitions et reformulation des exigences fonctionnelles.	Olivier Tremblay-Noël
2024-01-24	0.6	Complétion des parties 4.3, 4.4, 4.5 et 4.6	Mohamed Chaoui
2024-01-25	0.7	Exigences fonctionnelles 3.1 à 3.11	Olivier Tremblay-Noël
2024-01-25	0.8	Exigences fonctionnelles 3.12 à 3.15 + rectification 4.4	Zied Kaabi
2024-01-25	0.9	Mise en page	Olivier Tremblay-Noël
2024-01-25	1.0	Relecture finale	Omar Ben Thami

# 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	8
<b>2. Description globale</b>	<b>8</b>
2.1. Caractéristiques des usagers	8
2.2. Interfaces	8
2.3. Contraintes générales	9
2.4. Hypothèses et dépendances	9
<b>3. Exigences fonctionnelles</b>	<b>10</b>
3.1. Système de compte - Authentification	10
3.2. Intégration du clavier	11
3.3. Clavier – Canaux de discussion	12
3.4. Avatar	12
3.5. Commun à tous les modes de jeu	13
3.6. Modes de jeu Classique	15
3.7. Modes de jeu Temps limité 1 différence	15
3.8. Administration & Création	15
3.9. Reprise vidéo	17
3.10. Observateur	19
3.11. Personnalisation	20
3.12. Système d'amis	20
3.13. Système de blocage	21
3.14. Système de récompense - Boutique	21
3.15. Système de récompense - Roue de fortune	22
<b>4. Exigences non fonctionnelles</b>	<b>22</b>
4.1. Utilisabilité	22
4.2. Fiabilité	23
4.3. Performance	23
4.4. Maintenabilité	23
4.5. Contraintes de conception	24
4.6. Sécurité	24

# Spécifications des requis du système (SRS)

## 1. Introduction

### 1.1. But

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

### 1.2. Définitions, acronymes et abréviations

#### 1.2.1. Utilisateur :

Tout individu qui se connecte avec un compte utilisateur. Un utilisateur a un profil comprenant son courriel, son nom, son mot de passe, son historique de points, ses captures vidéo. Il est nécessaire de se connecter à son compte utilisateur pour accéder aux options du jeu.

#### 1.2.2. Pseudonyme :

Chaîne de caractères alphabétique de 1 à 16 caractères. Les noms d'utilisateurs sont uniques et non sensibles à la case.

#### 1.2.3. Mot de passe :

Chaîne de caractères alphanumériques de 5 à 8 caractères permettant à un utilisateur de se connecter à son compte.

#### 1.2.4. Courriel :

Chaîne de caractères alphabétique de 3 à 254 caractères de forme <utilisateur>@<domaine>. Le courriel n'a aucune utilité dans la forme actuelle du programme.

#### 1.2.5. Administrateur :

Utilisateur connaissant le mot de passe suivant, "admin", et pouvant ainsi administrer et créer des fiches.

#### 1.2.6. Fiche :

Ensemble de deux images de 640\*480 pixels et de leurs différences sur lequel un jeu va être lancé.

#### 1.2.6. Image modifiée :

Image comportant les différences par rapport à l'image de référence.

#### 1.2.8. Création de partie :

Action de dessiner une fiche à l'aide des outils de dessin.

#### 1.2.9. Clavardage :

Action d'échanger et de visualiser des messages et images entre participants à un canal de discussion.

#### 1.2.10. Canal de discussion :

Zone d’affichage et d’écriture de messages accessible sur la page d’accueil pour tous les joueurs présents sur la page d’accueil, pouvant être créés ou détruits par les utilisateurs. Chaque canal a un nom unique.

1.2.11. Canal de discussion principal :

Canal de discussion ne pouvant pas être détruit, ni quitté.

1.2.12. Canal de discussion de partie :

Canal de discussion exclusif aux joueurs et observateurs d’une partie.

1.2.13. Amitié :

Lien réciproque unissant deux utilisateurs suite à une demande d’amitié et son acceptation par l’autre utilisateur.

1.2.14. Utilisateur bloqué :

Lien unidirectionnel empêchant l’utilisateur bloqué de voir le compte de celui qui l’a bloqué, ni de joindre une partie auquel il participe.

1.2.15. Partie :

Partie en cours d’exécution d’un des modes de jeu à l’aide des fiches.

1.2.16. Joueur :

Utilisateur ayant rejoint une salle d’attente ou en train de participer à une partie.

1.2.17. Observateur :

Utilisateur ayant rejoint une partie déjà entamée.

1.2.19. Interaction :

Action de donner en tant qu’observateur un indice à un ou des joueurs (ou de l’induire en erreur) en montrant une région rectangulaire dans le dessin.

1.2.18. Salle d’attente :

Page permettant à des utilisateurs de joindre une partie tant que la salle n’est pas pleine.

1.2.19. Créateur de partie :

Utilisateur créant une salle d’attente à partir d’une fiche et d’un mode de jeu, pouvant déclencher le début d’une partie et de choisir le temps alloué et l’accès au mode triche.

1.2.20 Différence :

Différence d’un groupe contigu de pixels différents entre le dessin original et le dessin modifié.

1.2.21. Mode de jeu Classique :

Mode de jeu pour 2 à 4 joueurs où les joueurs partagent une même fiche. Lorsqu’un joueur trouve une différence, elle apparaît pour les autres joueurs et le point de partie accordé à cette différence est alloué au joueur qui l’a trouvée. Le jeu se termine lorsque le compte à rebours de la partie est écoulé ou que le nombre de différences restantes ne peut pas permettre de dépasser le meneur.

1.2.22 Mode de jeu Temps Limité :

Mode de jeu pour 2 à 4 joueurs où les joueurs partagent une même fiche. Une seule des différences originales de la fiche est présente dans l'image modifiée. Le premier joueur à trouver la différence se fait accorder un point de partie et le nombre de différences trouvées total est incrémenté, suite à quoi le compte à rebours est incrémenté d'un nombre prédéterminé de secondes, jusqu'à concurrence d'un seuil, lui aussi prédéterminé. Ensuite, une nouvelle fiche comprenant une seule différence apparaît, et ainsi de suite jusqu'à ce qu'il n'y ait plus de fiches disponibles ou que le compte à rebours atteigne zéro.

1.2.23. Point de partie :

Métrieque permettant de déterminer qui est le gagnant d'une partie.

1.2.24. Dinar :

Monnaie gagnée suite à une partie. Tous les joueurs ayant terminé une partie reçoivent un dinar, sauf le gagnant qui reçoit 5 dinars. Un match nul donne 2 dinars pour les meneurs.

1.2.25. Boutique :

Page où les utilisateurs peuvent acheter des effets sonores de découverte ou des médailles, ou faire tourner la roue de fortune.

1.2.26. Effet sonore de découverte :

Son entendu par tous les joueurs lorsqu'une différence est trouvée par un joueur. Chaque son alternatif de la boutique coûte 40 dinars.

1.2.27. Médailles :

Les médailles sont des icônes (bronze, argent et or) qui apparaissent au bas de l'avatar de l'utilisateur. Leurs prix respectifs sont : 20 points pour bronze, 50 points pour argent et 100 points pour or.

1.2.28. Roue de fortune :

Les joueurs peuvent, contre 5 points, faire tourner une roue afin de gagner des récompenses aléatoires, soit des multiplicateurs de points valides pour une seule partie ou rien du tout.

1.2.29. Reprise vidéo :

Séquence d'exécution d'une partie classique dans la perspective d'un joueur donné.

1.2.30 Client léger :

Version mobile (Android) de l'application du jeu.

1.2.31. Client lourd :

Version bureau (Windows) de l'application du jeu.

1.2.32. Thème visuel :

Configuration de l'interface disponible sous format clair ou foncé.

1.2.33. Langue :

Configuration de toutes les composantes de l'interface en deux langues, soit le français ou l'anglais.

1.2.34. API:

- Interface de programmation d'application

#### 1.2.35. BD :

Base de donnée

#### 1.2.16. UX :

Expérience utilisateur

### 1.3. Vue d'ensemble du document

La suite de ce document sera divisée en trois chapitres : description globale, exigences fonctionnelles, exigences non fonctionnelles.

Le chapitre portant sur la description globale donnera une brève description du logiciel qui sera développé. Cela tiendra compte des différentes caractéristiques des usagers, des différentes interfaces utilisées et implémentées, des contraintes générales et des hypothèses et dépendances liées au programme.

Le chapitre sur les exigences fonctionnelles couvrira toutes les fonctionnalités qui seront implémentées ou non, dépendamment du type de celle-ci, dans le logiciel. Une description de chacune d'elles sera faite dans un langage naturel et le type (essentiel, souhaitable) sera clairement indiqué.

Le chapitre sur les exigences non fonctionnelles, quant à lui, indiquera toutes les exigences, ne touchant pas à une fonctionnalité, que le logiciel devra atteindre. C'est-à-dire tout ce qui touche à l'utilisabilité, la fiabilité, la performance, la maintenabilité, les contraintes de conception, la sécurité, les exigences de la documentation usager en ligne et du système d'assistance.

## 2. Description globale

Le logiciel est une application multiplateforme d'un jeu des 7 différences. Il est divisé en deux types de clients. Un client dit « lourd » qui sera un exécutable sur ordinateur. Et un autre client « léger » qui sera une application de type apk sur Android. Ces deux clients pourront communiquer entre eux via un serveur commun pour permettre aux joueurs des deux plateformes de jouer ensemble.

### 2.1. Caractéristiques des usagers

PolyDiff est conçu pour être intuitif et accessible, ciblant les joueurs âgés de 12 à 30 ans. Ces utilisateurs, bien qu'ils ne possèdent pas nécessairement de connaissances avancées en informatique, ont une bonne familiarité avec le jeu des 7 différences. L'installation et l'exécution du jeu seront le plus simple possible pour l'utilisateur, même pour ceux qui n'ont que des connaissances de base en matière d'utilisation d'appareils et de logiciels.

### 2.2. Interfaces

#### 2.2.1. Interfaces usagers

Les interfaces usagers devront aider à l'UX du logiciel. Pour ce faire, elles suivent des normes qui sont souvent utilisées dans les autres applications pour ne pas désorienter l'utilisateur et rendre son expérience la plus fluide

possible. Elles seront aussi simplifiées au maximum pour que l'utilisateur n'ait pas trop à réfléchir par rapport à l'interface. Ce dernier pourra se concentrer plus facilement sur ce qui importe vraiment, dans ce cas-ci, la partie en cours du jeu. L'interface pour le client lourd sera codée avec Angular et pour le client léger Flutter sera utilisé. Pour ce qui est du serveur, une interface de type ligne de commande sera utilisée pour le démarrer.

#### *2.2.2. Interfaces matérielles*

Le client lourd sera exécuté sur un ordinateur, donc les interfaces matérielles habituelles telles qu'un clavier et une souris seront prises en charge. Pour ce qui est du client léger, celui-ci sera exécuté sur un appareil mobile Android. Donc, le logiciel devra pouvoir supporter les différentes interactions avec l'écran tactile et la capture de photo avec l'appareil photo.

#### *2.2.3. Interfaces logicielles*

Pour le client lourd, l'application requiert Windows 10. Le client léger requiert Android 11.

#### *2.2.4. Interfaces de communication*

La communication du logiciel est assurée à l'aide d'un serveur implémenté avec Node.js et express. HTTP et WebSocket sont utilisés pour assurer les différentes communications. Le serveur permettra de créer des parties en ligne avec d'autres joueurs, qu'ils soient sur le client léger ou lourd. Il permettra aussi aux joueurs de se connecter à leur compte utilisateur. De plus, il pourra gérer le stockage de données directement sur le serveur ou dans une base de données MongoDB. La communication avec la BD se fera aussi au travers du serveur.

### **2.3. Contraintes générales**

Le client lourd n'a pas de contraintes de mémoire particulières. Cependant, puisque le client léger est sous forme d'apk, Google ne permet pas de mettre des applications de plus de 100Mo sur le PlayStore, donc respecter cette limite serait une bonne idée.

Le serveur devra pouvoir supporter des parties de 2 à 4 joueurs simultanément. La gestion de multiples canaux de discussion engendrera une nécessité de synchroniser les messages du canal avec toutes les personnes présentes dans celui-ci. La synchronisation des informations (chronomètre, différences trouvées, etc.) d'une partie en cours est primordiale pour s'assurer du bon fonctionnement du jeu. L'interface utilisateur entre le client léger et lourd se devra d'être cohérente. Celles-ci doivent être attrayantes pour l'utilisateur pour améliorer son expérience.

### **2.4. Hypothèses et dépendances**

Une hypothèse de base est que les usagers ont une connexion internet avec un débit suffisant pour pouvoir se connecter, jouer en ligne et converser avec les autres usagers. On estime que le joueur est assez proche de l'endroit où le serveur est hébergé pour avoir une latence acceptable.



### 3. Exigences fonctionnelles

#### 3.1. Système de compte - Authentification

	Client	Type	Description
3.1.1.	Lourd & léger	Essentiel	Si le pseudonyme, le courriel et le mot de passe utilisés par l'utilisateur sont valides et que le pseudonyme est unique, le système doit permettre à un utilisateur de créer un compte.
3.1.2.1	Lourd & léger	Essentiel	Si l'un des champs spécifiés en 3.1.1 est invalide, le système doit en avertir l'utilisateur.
3.1.2.2	Lourd & léger	Essentiel	Si l'un des champs spécifiés en 3.1.1 est invalide, le système doit l'empêcher de créer son compte.
3.1.3.	Lourd & léger	Essentiel	Si le nom d'utilisateur ou le mot de passe sont invalides, le système doit notifier l'utilisateur et l'empêcher d'accéder au système.
3.1.4.	Lourd & léger	Essentiel	Si le nom d'utilisateur et le mot de passe sont valides et que personne n'est déjà connecté sur ce compte, le système doit permettre à l'utilisateur d'accéder au compte.
3.1.5.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de choisir son avatar parmi une liste prédéfinie d'images lors de la création du compte.
3.1.6.	Lourd & léger	Essentiel	Si le nouveau pseudonyme est unique, le système doit permettre à un utilisateur de le modifier.
3.1.7.	Lourd & léger	Essentiel	Un utilisateur doit pouvoir visualiser, depuis la page de paramètres de compte, les statistiques d'utilisation de la plateforme, soit le nombre de parties jouées, le nombre de parties gagnées, la moyenne de différences trouvées par partie et le temps moyen par partie.

3.1.8.	Lourd & léger	Essentiel	Un utilisateur doit pouvoir visualiser, depuis la page de paramètres de compte, un relevé des connexions et déconnexions au compte de l'utilisateur avec dates et heures correspondantes.
3.1.9.	Lourd & léger	Essentiel	Un utilisateur doit pouvoir visualiser, depuis la page de paramètres de compte, une historique des parties jouées doit aussi être affiché à l'utilisateur, avec, pour chaque entrée, l'heure et la date de la partie ainsi qu'un indicateur si le joueur a gagné ou non la partie.
3.1.10	Léger	Souhaitable	Après la première authentification sur l'appareil, le système doit demander à l'utilisateur s'il veut utiliser le trousseau de clefs de l'appareil pour les prochaines connexions.
3.1.11	Léger	Souhaitable	Si le mot de passe est enregistré dans le trousseau de clefs, que le pseudonyme est correct et que l'utilisateur a correctement entré le code de l'appareil, le système doit autoriser l'utilisateur à se connecter.

### 3.2. Intégration du clavier

	Client	Type	Description
3.2.1.	Lourd	Essentiel	Si la boîte de clavier est en mode intégré, le système doit permettre à l'utilisateur de la visualiser en tout temps dans l'interface.
3.2.2. 1	Lourd	Essentiel	Le système doit permettre à l'utilisateur d'alternar l'affichage de la boîte de clavier en mode intégré.
3.2.2. 2	Lourd	Essentiel	Le système doit permettre à l'utilisateur d'alternar l'affichage de la boîte de clavier en mode fenêtré.
3.2.3.	Léger	Essentiel	Le système doit permettre à l'utilisateur connecté d'accéder au clavier en tout temps depuis n'importe quelle page de l'application.
3.2.4.	Léger	Essentiel	Le système doit notifier l'utilisateur lorsqu'il reçoit un nouveau message à l'aide d'un indicateur visuel et sonore.
3.2.5.	Léger	Essentiel	Le système doit garder l'indicateur de nouveau message visible tant qu'il reste des boîtes de clavier non lues.

3.2.6.	Léger	Essentiel	Si l'application roule en arrière-plan, le système doit notifier l'utilisateur à l'aide d'une notification système.
--------	-------	-----------	---

### 3.3. Clavardage – Canaux de discussion

	Client	Type	Description
3.3.1.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur d'accéder à la liste de ses conversations actives, incluant minimalement le canal principal.
3.3.2.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de créer un nouveau canal de discussion avec un nom unique (les mêmes règles que pour le pseudonyme s'appliquent).
3.3.3.	Lourd & léger	Essentiel	Le système doit automatiquement supprimer le canal de discussion quand le dernier joueur a quitté.
3.3.4.	Lourd & léger	Essentiel	Un utilisateur doit pouvoir joindre un canal de discussion choisi parmi une liste.
3.3.5.	Lourd & léger	Essentiel	Un utilisateur doit voir l'historique d'un canal de discussion lorsqu'il le joint, incluant le canal principal lorsqu'il se connecte à l'application.
3.3.6.	Lourd & léger	Essentiel	Les joueurs d'une partie doivent automatiquement avoir accès à un canal de discussion dédié dès la création d'une partie puis de se faire déconnecter dès la fin de la partie.
3.3.7.	Lourd	Souhaitable	Un utilisateur doit pouvoir mettre et démettre un autre utilisateur dans une boîte de clavardage en sourdine et ainsi ne plus voir ses messages.
3.3.8.	Léger	Souhaitable	Un utilisateur doit pouvoir envoyer des images dans un canal de discussion.
3.3.9.	Léger	Souhaitable	Un utilisateur doit pouvoir envoyer des Gifs dans un canal de discussion.

### 3.4. Avatar

	Client	Type	Description
3.4.1.	Lourd &	Essentiel	L'utilisateur doit pouvoir changer son avatar à partir d'une banque d'images.

	léger		
3.4.2.	Lourd	Essentiel	L'avatar d'un utilisateur doit toujours accompagner son pseudonyme.
3.4.3.	Léger	Essentiel	L'utilisateur doit pouvoir téléverser son avatar à partir de la caméra de l'appareil.

### 3.5. Commun à tous les modes de jeu

	Client	Type	Description
3.5.1.	Lourd & léger	Essentiel	Un utilisateur doit pouvoir créer une partie de chaque mode de jeu.
3.5.2.	Lourd & léger	Essentiel	Un utilisateur doit pouvoir joindre la salle d'attente d'une partie créée tant que la partie n'est pas démarrée.
3.5.3.	Lourd & léger	Essentiel	Le système doit permettre au créateur de la partie de la débiter quand le nombre de membres est adéquat.
3.5.4.	Lourd & léger	Essentiel	Le système doit empêcher le créateur de la partie de débiter tant que le nombre de joueurs est insuffisant.
3.5.5.	Lourd & léger	Essentiel	Le système doit permettre de créer plusieurs parties de la même fiche.
3.5.6	Lourd & léger	Essentiel	Le système doit empêcher un utilisateur de rejoindre une salle d'attente pleine.
3.5.7	Lourd & léger	Essentiel	Le système doit permettre au créateur de la partie de configurer le temps et l'accès au mode triche pour la partie.
3.5.8	Lourd & léger	Essentiel	Le système doit permettre au créateur de la partie rendre l'accès au mode triche accessible pour la partie.
3.5.9	Lourd & léger	Essentiel	Si le créateur a rendu le mode triche accessible et qu'un joueur l'a activé, les différences clignotent à raison de 4 Hz sur les deux images seulement pour lui.
3.5.10	Lourd & léger	Essentiel	Le système doit imposer une valeur de départ pour le compte à rebours d'une

	léger		partie.
3.5.11	Lourd & léger	Essentiel	Le système doit automatiquement mettre fin à une partie si le compte à rebours atteint 0.
3.5.12	Lourd & léger	Essentiel	La vue de jeu doit présenter les images originale et modifiée, le temps, le nom des joueurs ainsi que leur nombre de différences trouvées, soit leurs points de partie ainsi qu'un bouton d'abandon de partie.
3.5.13	Lourd & léger	Essentiel	Le système doit démarrer le compte à rebours automatiquement au début d'une partie.
3.5.14	Lourd & léger	Essentiel	Un joueur doit pouvoir trouver une différence en cliquant sur la différence tant dans l'image originale que modifiée.
3.5.15	Lourd & léger	Essentiel	Si une différence est trouvée par un joueur, le système doit faire clignoter la différence sur les deux images pour tous les joueurs.
3.5.16	Lourd & léger	Essentiel	Si une différence est trouvée par un joueur, le système doit émettre un effet sonore de découverte chez tous les joueurs.
3.5.17	Lourd & léger	Essentiel	Si une différence est trouvée par un joueur, le système doit incrémenter le pointage du joueur l'ayant trouvé.
3.5.18	Lourd & léger	Essentiel	Le système doit faire disparaître la différence dès que le clignotement est terminé.
3.5.19	Lourd & léger	Essentiel	Si un joueur clique sur une zone sans différence, le système doit émettre un effet sonore d'échec, ainsi qu'un effet visuel.
3.5.20	Lourd & léger	Essentiel	Si un joueur clique sur une zone sans différence, le système doit empêcher le joueur d'interagir avec le jeu pour une seconde.
3.5.21	Lourd & léger	Essentiel	Le système doit automatiquement retirer un joueur du canal de discussion d'une partie lorsqu'il l'abandonne.

### 3.6. Modes de jeu Classique

	Client	Type	Description
3.6.1.	Lourd & léger	Essentiel	Le système doit automatiquement mettre fin à la partie lorsqu'un joueur a atteint le seuil de différences trouvées.
3.6.2.	Lourd & léger	Essentiel	Le créateur d'une partie classique doit pouvoir choisir la fiche de la partie à partir d'une liste de fiches existantes qui indique leur nombre de différences respectives.

### 3.7. Modes de jeu Temps limité 1 différence

	Client	Type	Description
3.7.1.	Lourd & léger	Essentiel	Le système doit automatiquement choisir une fiche pour le début de partie et aléatoirement choisir une seule des différences de la fiche.
3.7.2.	Lourd & léger	Essentiel	Le système doit automatiquement attribuer un point à un joueur ayant trouvé la différence et le compte à rebours doit être augmenté du temps spécifié par le créateur de la partie jusqu'à concurrence du seuil aussi déterminé par le créateur de la partie, puis une fiche qui n'a pas déjà été utilisée dans la partie doit apparaître.
3.7.2.	Lourd & léger	Essentiel	Le système doit automatiquement terminer la partie lorsque toutes les fiches existantes ont été utilisées.

### 3.8. Administration & Création

	Client	Type	Description
3.8.1.	Lourd	Essentiel	Le système doit permettre à un utilisateur connecté d'accéder à un champ de texte pour entrer le code d'administrateur.
3.8.2.	Lourd	Essentiel	Si l'utilisateur connecté a entré le bon mot de passe administrateur, le système doit lui permettre d'accéder à la page d'administration et de création.

3.8.3.	Lourd	Essentiel	Le système doit montrer à l'administrateur toutes les fiches existantes et lui permettre de les supprimer.
3.8.4	Lourd	Essentiel	Le système doit permettre à l'administrateur d'accéder à la vue de création de partie.
3.8.5	Lourd	Essentiel	Le système doit permettre à l'administrateur de téléverser une image lors de la création.
3.8.6	Lourd	Essentiel	Le système doit permettre à l'administrateur de dessiner à l'aide d'un crayon.
3.8.7	Lourd	Essentiel	Le système doit permettre à l'administrateur d'effacer à l'aide d'une efface.
3.8.8	Lourd	Essentiel	Le système doit permettre à l'administrateur d'annuler (undo) ou refaire une action (redo).
3.8.9	Lourd	Essentiel	Le système doit automatiquement vérifier la présence de différences entre les deux images.
3.8.10	Lourd	Essentiel	Le système doit permettre à l'administrateur de sauvegarder le jeu.
3.8.11	Lourd	Essentiel	Le système doit automatiquement rendre le jeu sauvegardé accessible à tous.
3.8.12	Lourd	Essentiel	Le système doit permettre à l'administrateur de tracer un rectangle.
3.8.13	Lourd	Essentiel	Lorsque la touche shift est enfoncée et que l'outil de rectangle est sélectionné, le système doit permettre à l'administrateur de tracer un carré.
3.8.14	Lourd	Essentiel	Le système doit permettre à l'administrateur de tracer une ellipse.
3.8.15	Lourd	Essentiel	Lorsque la touche shift est enfoncée et que l'outil d'ellipse est sélectionné, le système doit permettre à l'administrateur de tracer un cercle.
3.8.16	Lourd	Essentiel	Lorsque la touche shift est enfoncée et que l'outil de rectangle est sélectionné, le système doit permettre à l'administrateur de tracer un carré.
3.8.17	Lourd	Essentiel	Lorsqu'un outil de dessin est sélectionné, le système doit tracer selon la couleur sélectionnée.
3.8.18	Lourd	Essentiel	Le système doit permettre à l'administrateur de prévisualiser la forme pendant

			qu'il la trace.
3.8.19	Lourd	Essentiel	Le système doit permettre à l'administrateur de tracer à l'aide de sa souris grâce au clic et terminer le tracé lorsque la touche est relâchée.
3.8.20	Lourd	Essentiel	Le système doit permettre à une forme de sortir du canevas, sans toutefois que le tracé soit rendu (rendered) dans la zone externe au canevas.
3.8.21	Lourd	Essentiel	Le système doit permettre à l'administrateur de remplir de la couleur sélectionnée une région de pixels connectés qui partagent une même couleur à l'aide de l'outil de seau.
3.8.22	Lourd	Essentiel	Le système doit permettre à l'administrateur de déterminer le diamètre d'un jet d'aérosol.
3.8.23	Lourd	Essentiel	Le jet d'aérosol doit être différent à chaque fois qu'il est déclenché (i.e. à chaque nouveau clic)
3.8.24	Lourd	Essentiel	Le système doit permettre à l'utilisateur de tracer un jet d'aérosol.
3.8.25	Lourd	Souhaitable	Le système doit permettre à l'administrateur de générer aléatoirement des différences.
3.8.26	Lourd	Souhaitable	Le système doit permettre de tracer dans un canevas dont les différences ont été générées aléatoirement..
3.8.27	Lourd	Souhaitable	Deux générations de différences doivent donner des résultats uniques (non déterministes).
3.8.28	Lourd	Souhaitable	Le système doit permettre à l'administrateur de choisir le nombre de différences générées aléatoirement.

### 3.9. Reprise vidéo

	Client	Type	Description
3.9.1.	Lourd & léger	Essentiel	Le système doit permettre à un joueur de visionner ou non sa reprise vidéo à la fin d'une partie, le visionnement est plein écran et l'affichage est identique



			à la partie jouée, sauf les options additionnelles.
3.9.2.	Lourd & léger	Essentiel	Le système doit jouer la reprise vidéo à vitesse 1x par défaut.
3.9.3.	Lourd & léger	Essentiel	Le système doit permettre à un joueur de jouer la reprise vidéo en mode 1x, 2x et 4x.
3.9.4.	Lourd & léger	Essentiel	Le système doit permettre à un joueur de mettre le visionnement sur pause ainsi que de reprendre là où il a mis à pause ou de reprendre depuis le début.
3.9.5.	Lourd & léger	Essentiel	Le système doit permettre à un joueur de voir la barre d'avancement de la reprise vidéo.
3.9.6.	Lourd & léger	Essentiel	Le système doit permettre à un joueur de naviguer temporellement dans la reprise vidéo en interagissant avec la barre d'avancement de la reprise vidéo tout en conservant l'état de vitesse et de pause.
3.9.7.	Lourd & léger	Essentiel	Si le joueur n'a pas explicitement demandé la sauvegarde de la reprise vidéo, le système doit supprimer la reprise vidéo automatiquement lorsque le joueur quitte le visionnement.
3.9.8.	Lourd & léger	Essentiel	Le système doit permettre à un joueur de sauvegarder sa reprise vidéo suite à une partie.
3.9.9.	Lourd & léger	Essentiel	Même si la fiche d'une partie a été supprimée, le système doit permettre à un joueur de visionner ses reprises vidéos sauvegardées.
3.9.10.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de supprimer ses reprises vidéos.
3.9.11.	Lourd & léger	Souhaitable	Le système doit permettre à un utilisateur de partager publiquement une ou plusieurs reprises vidéo.
3.9.11.	Lourd & léger	Souhaitable	Le système doit permettre à un utilisateur de rendre privée une reprise vidéo marquée comme publique.
3.9.12.	Lourd & léger	Souhaitable	Le système doit permettre à un utilisateur de visualiser la liste des reprises vidéos publiques, comprenant le nom de la fiche et le pseudonyme de

			l'utilisateur propriétaire de la reprise vidéo.
3.9.13.	Lourd & léger	Souhaitable	Le système doit permettre à un utilisateur de filtrer la liste de reprises vidéos publiques en fonction des pseudonymes ou des noms de fiches associés.

### 3.10. Observateur

	Client	Type	Description
3.10.1.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de joindre une partie démarrée en tant qu'observateur à partir d'une liste des parties en cours (une salle d'attente n'est pas considérée comme une partie en cours).
3.10.2.	Lourd & léger	Essentiel	Le système doit indiquer s'il y a au moins un observateur pour une partie donnée dans la liste des parties en cours.
3.10.3.	Lourd & léger	Essentiel	Le système doit permettre plusieurs observateurs par partie.
3.10.4.	Lourd & léger	Essentiel	Le système doit automatiquement ajouter un observateur à au canal de discussion de la partie.
3.10.5.	Lourd & léger	Essentiel	Le système doit empêcher un observateur de trouver des différences.
3.10.6.	Lourd & léger	Essentiel	Le système doit montrer aux joueurs le nombre d'observateurs présents.
3.10.7.	Lourd & léger	Essentiel	Le système doit permettre à un observateur de quitter une partie.
3.10.8.	Lourd & léger	Essentiel	Le système doit permettre à un observateur de mettre un rectangle en surbrillance à l'aide du clic et de la souris tout en le prévisualisant lors de l'opération..
3.10.9.	Lourd & léger	Essentiel	Le système doit permettre en tout temps à un observateur de choisir si la surbrillance est visible pour un et un seul joueur ou pour tous les joueurs.

3.10.10.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de joindre une partie démarrée en tant qu'observateur à partir d'une liste des parties en cours.
3.10.11.	Lourd & léger	Essentiel	La surbrillance doit être visible temporairement et seulement une fois le clic relâché.
3.10.12.	Lourd & léger	Essentiel	Le système doit empêcher un observateur de tracer une surbrillance pendant les trois secondes suivant une surbrillance.
3.10.13.	Lourd & léger	Essentiel	La surbrillance des autres observateurs doit être invisible pour un observateur donné.
3.10.14.	Lourd & léger	Essentiel	La surbrillance des différents observateurs doit être visuellement différente pour les joueurs.

### 3.11. Personnalisation

	Client	Type	Description
3.11.1.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de choisir son thème dans une liste.
3.11.2.	Lourd	Essentiel	Le système doit étendre le thème à toutes les interfaces uniques à un utilisateur.
3.11.3.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de choisir sa langue dans une liste.
3.11.4.	Lourd & léger	Essentiel	Le système doit étendre la langue à tous les éléments de l'interface qui n'ont pas été créés par un utilisateur ou par l'administrateur (noms, titres).
3.11.5.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de conserver l'état de langue et de thème entre chaque connexion à l'application.

### 3.12. Système d'amis

	Client	Type	Description
--	--------	------	-------------

3.12.1.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur d'ajouter ou de retirer des utilisateurs de sa liste d'amis
3.12.2.	Lourd & Léger	Essentiel	Le système doit permettre à un utilisateur la création de parties "Amis seulement" accessibles uniquement aux amis du créateur.
3.12.3.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur la création de parties "Amis et leurs amis" accessibles aux amis du créateur et à leurs amis directs.
3.12.4.	Lourd & léger	Essentiel	Le système doit permettre à un utilisateur de consulter ses amis via une interface dédiée dans la page de profil.

### 3.13. Système de blocage

	Client	Type	Description
3.13.1.	Lourd & léger	Souhaitable	Le système doit permettre à un utilisateur de bloquer un autre utilisateur.
3.13.2.	Lourd & Léger	Souhaitable	Si l'utilisateur bloque un utilisateur faisant partie de sa liste d'amis, le système retire cet utilisateur de la liste.
3.13.3.	Lourd & léger	Souhaitable	Le profil d'un utilisateur bloquant ne serait plus visible pour l'utilisateur bloqué lorsqu'il effectue une recherche de profils.
3.13.4.	Lourd & léger	Souhaitable	Un utilisateur bloqué ne peut pas rejoindre une partie si l'utilisateur qu'il a bloqué participe à celle-ci
3.13.5.	Lourd & léger	Souhaitable	Un utilisateur bloqué ne peut plus consulter les messages envoyés par l'utilisateur bloquant dans tous les canaux de discussion et vice versa.

### 3.14. Système de récompense - Boutique

	Client	Type	Description
3.14.1.	Lourd & léger	Essentiel	Le Système doit permettre à l'utilisateur de gagner des points automatiquement à la fin de chaque partie, attribuant un point pour le simple achèvement de la partie,

			deux points en cas d'égalité et cinq points au joueur gagnant.
3.14.2.	Lourd & léger	Essentiel	Le Système doit proposer une boutique où les utilisateurs peuvent échanger leurs points.
3.14.3.	Lourd & léger	Essentiel	La boutique doit offrir une variété d'options de cosmétiques, soit des médailles et des effets sonores exclusifs
3.14.4	Lourd & léger	Essentiel	L'utilisateur doit pouvoir consulter son solde de points à tout moment.

### 3.15. Système de récompense - Roue de fortune

	Client	Type	Description
3.15.1.	Lourd & léger	Essentiel	Le système doit permettre à l'utilisateur d'accéder à la roue de récompense via l'interface de la boutique.
3.15.2.	Lourd & léger	Essentiel	Le système doit permettre à l'utilisateur de tourner la roue contre 5 points
3.15.3	Lourd & léger	Essentiel	L'utilisateur doit avoir la possibilité de remporter des récompenses variées grâce à la roue, y compris la possibilité de ne recevoir aucune récompense.
3.15.4.	Lourd & léger	Essentiel	La roue doit proposer des gains, notamment des multiplicateurs de points aléatoires comme étant bonus, qui seront automatiquement appliqués lors de la prochaine partie.

## 4. Exigences non fonctionnelles

### 4.1. Utilisabilité

4.1.1 Le jeu doit pouvoir être compris et pris en main par tous les utilisateurs en moins de 30 minutes.

4.1.2 Le processus de création d'une fiche doit pouvoir être compris et pris en main par tous les utilisateurs en moins de 30 minutes.

4.1.3 Une partie de jeu classique doit se terminer au maximum dans le temps indiqué par le créateur de celle-ci.

4.1.4 La connexion au compte doit être complétée en moins de 30 secondes.

4.1.5 La recherche d'un profil utilisateur doit être complétée en moins de 10 secondes.

4.1.6 Une fiche créée doit être instantanément disponible pour jouer.

#### **4.2. Fiabilité**

4.2.1. L'application doit être disponible 23 heures sur 24 pour laisser 1 heure pour des maintenances éventuelles.

4.2.2 L'application doit avoir un temps moyen entre les pannes (MTBF) de 75 heures.

4.2.3 L'application doit avoir un temps moyen jusqu'à la réparation (MTTR) de 4 heures.

#### **4.3. Performance**

4.3.1 Les données d'une partie de jeu doivent être synchronisées.

4.3.2 Le serveur doit supporter 4 joueurs en même temps sur une partie de jeu

4.3.3 La requête de connexion doit être traitée en un maximum de 5 secondes

4.3.4 La requête de connexion doit être traitée en moyenne en 2 secondes.

4.3.4 La requête pour rechercher un utilisateur doit être traitée en un maximum de 5 secondes.

4.3.5 La requête pour rechercher un utilisateur doit être traitée en moyenne en 2 secondes.

4.3.6 La requête pour rejoindre un lobby doit être traitée en un maximum de 2 secondes.

4.3.7 La requête pour rejoindre un lobby doit être traitée en moyenne en 1 seconde.

4.3.8 L'envoi d'un message dans le chat doit prendre au maximum 2 secondes.

4.3.9 L'envoi d'un message dans le chat doit prendre en moyenne 1 seconde.

4.3.10 Les fichiers médias tels que les images et les sons ne doivent pas prendre plus de 75% de la mémoire du serveur.

#### **4.4. Maintenabilité**

4.4.1 Les noms de fichiers et dossiers respectent le format kebab-case.

4.4.2 L'application doit générer des messages d'erreurs précis et compréhensibles par l'utilisateur moyen.

4.4.3 Les constantes sont écrites toutes en majuscule.

4.4.4 Les commentaires sont écrits en français.

4.4.5 Les variables et méthodes respectent le format camel-case.

4.4.6 Les sprints seront faits sur une période de 2 semaines.

4.4.7 Les rencontres de SCRUM se font à chaque lundi 12h.

4.4.8 Les rencontres de SCRUM se font en vocal sur Discord.

4.4.9 Les tickets Jira ayant attrait au client lourd commencent par "Electron".

4.4.10 Les tickets Jira ayant attrait au client léger commencent par "Android".

4.4.11 Le titre des tickets Jira est écrit en français.

4.4.12 Les noms des branches sur Gitlab doivent être descriptifs et significatifs

4.4.13 les noms des branches doivent suivre la convention de nommage uniforme proposée par l'équipe:

CATEGORIE\_FEATURE\_[OPTIONAL] . Exemple : Clavardage\_Canaux-de-discussion\_[fix]

4.4.14 les branches obsolètes doivent être supprimées sur le champ

4.4.15 les fonctionnalités et leurs rectifications doivent être développées sur des branches distinctes plutôt que sur la branche principale

4.4.16 les développeurs doivent fusionner fréquemment avec la branche principale et assurer une intégration continue du projet

4.4.17 les messages de commit doivent être significatifs et fournir des informations claires sur les modifications apportées

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

4.5.1 Le client lourd doit être codé à l'aide de TypeScript et Angular

4.5.2 L'exécutable du client lourd doit être généré à l'aide d'Electron.

4.5.3 Le client lourd doit être codé à l'aide de Dart et Flutter.

4.5.4 Le serveur doit être codé à l'aide Node.js, express et TypeScript

4.5.5 L'API Keychain d'android doit être utilisée pour la connexion au compte avec le mot de passe du téléphone pour le client léger.

4.5.6 L'API Mongoose doit être utilisé pour la connection à la DB de MongoDB

4.5.7 L'API Socket.IO doit être utilisé pour connecter les utilisateurs entre eux.

4.5.8 Le framework express doit être utilisé pour faire l'API de notre application.

4.5.9 L'API de Tenor doit être utilisé pour gérer l'envoi de GIF dans les canaux de discussion

#### **4.6. Sécurité**

4.6.1 L'application doit crypter le mot de passe de l'utilisateur lors de la création et de la connexion au compte.

4.6.2 L'application doit crypter l'email de l'utilisateur lors de la création et de la connexion au compte.

4.6.3 L'application doit crypter le mot de passe administrateur.