Équipe 204

Projet d’évolution d’un logiciel
Spécifications des requis du système (SRS)

Version 1.2

Historique des révisions

Date	Version	Description	Auteur
2021-01-29	1.0	Première écriture du document de spécification des requis.	Maxime Fecteau Étienne Plante Antoine Morcel Guillaume Beausoleil Samuel Poulin Simon Malouin
2021-02-19	1.1	Complétion du document pour la première version.	Maxime Fecteau Antoine Morcel Guillaume Beausoleil Samuel Poulin Simon Malouin
2021-04-13	1.2	Révision complète basée sur la correction de la première version et la remise finale.	Maxime Fecteau

Table des matières

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

Android Studio : est l'application que nous utilisons pour développer le client léger. Cette application nous permet de rouler un émulateur d'un appareil android ainsi que de faire rouler notre code sur cet appareil émulé.

Android : est un système d'exploitation pour appareil mobile.

Client léger : est l'application qui roule sur un appareil Android que nous allons développer. Le client léger communique avec notre serveur pour pouvoir jouer avec d'autres clients légers et/ou lourds.

Client lourd : est l'application qui roule sur un ordinateur qui roule Windows avec Électron. Le client lourd communique avec notre serveur pour pouvoir jouer avec d'autres clients légers et/ou lourds.

Electron : Outil de construction d'application native à partir d'application web qui utilise les technologies ci-haut.

GIF : Graphics Interchange Format (Format d'échange d'images)

Hôte : Joueur qui crée la partie

HTML : HyperText Markup Language, est utile pour désigner les pages web. Le HTML définit le squelette de la page web.

Joueur actif : Joueur qui est en train de deviner ou de dessiner.

Joueur passif : Joueur qui n'est pas en train de deviner ou dessiner.

Équipe active : Dans le mode de jeu classique, cela désigne l'équipe qui est en train de dessiner et de deviner.

Équipe passive : Dans le mode de jeu classique, cela désigne l'équipe qui est inactive durant la période de dessin ou l'équipe qui obtient le droit de réplique lorsque l'équipe active ne trouve pas le bon mot.

Joueur réel : Une personne qui utilise soit un client léger ou un client lourd pour utiliser notre application.

Joueur virtuel : Une intelligence artificielle très rudimentaire qui remplace un joueur réel pour le dessin des mots à deviner.

Manche: Période durant laquelle le dessinateur de l'équipe active dessine un dessin et le devineur essaye de trouver le mot dessiné.

Dessinateur: Personne dont le rôle dans l'équipe active qui dessine le mot reçu du serveur à l'aide des outils.

Devineur: Personne dont le rôle est de deviner le mot qui est dessiné par le dessinateur.

Paire mot-image : Une paire d'un mot qui concorde avec une image. Plusieurs de ces paires forment la base de données des images que les joueurs virtuels utilisent pour dessiner les mots.

PC : Personnal Computer (Ordinateur personnel)

Rôle: Il rôle dicte les actions disponibles à l'utilisateur et son but. Les rôles sont devineur et dessinateur.

SCSS : Sassy Cascading StyleSheet, est utile pour styler les squelettes définis en HTML. Le SCSS définit l'apparence de la page web.

TypeScript : Une variante typée du javascript, un langage de programmation utilisé pour programmer les fonctionnalités interactives des pages web.

Windows : est un système d'exploitation pour un ordinateur personnel.

1.3. Vue d'ensemble du document

La section 2 du document offre une description générale du produit. La section 3 contient une description des exigences fonctionnelles et la section 4 contient une description des exigences non-fonctionnelles.

2. Description globale

Le logiciel à développer est un jeu qui se base sur une application de dessin, qui offre différents outils de dessin, développée précédemment dans le cadre du projet 2: PolyDessin. Le but de celui-ci est de faire deviner des mots ou expressions choisis à ses coéquipiers en représentant ceux-ci par des dessins. Cette application doit permettre le jeu en ligne et du clavardage avec les autres joueurs ainsi qu'offrir plusieurs modes de jeu différents. Le jeu doit pouvoir être joué sur ordinateur et sur tablette mobile.

2.1. Caractéristiques des usagers

Les utilisateurs principaux de l'application seront des ingénieurs logiciels et dans notre cas, des étudiants en génie logiciel.

2.2. Interfaces

2.2.1. Interfaces usagers

Pour nos deux clients, il y aura une page de connexion. Cette page permet soit de se connecter avec un compte existant ou de créer un compte. Une fois connecté, l'usager est amené à un menu principal qui permet de choisir un mode de jeu puis une difficulté pour ce mode de jeu. Dans le menu principal, il y a aussi l'option de passer à travers un tutoriel interactif qui explique le fonctionnement général de l'application. Une fois le mode de jeu choisi, l'usager sera bougé vers une nouvelle vue que l'on qualifie de salle de jeu pour attendre d'autres joueurs. Dans le menu principal et dans une salle de jeu, un usager pourra aussi voir sa liste d'amis et inviter ceux-ci à une partie. De plus, un usager peut voir son profil qui contient son historique de parties ainsi que certaines statistiques, par exemple le temps total joué, le nombre de parties jouées, le pourcentage de victoires, etc. Une fois que tous les joueurs dans la salle de jeu sont prêts, les joueurs seront transférés sur la vue du jeu qui est essentiellement une grosse zone de dessin avec un panneau qui permet de choisir l'outil de dessin à utiliser, soit un crayon ou une efface. La salle de jeu ainsi que la vue du jeu inclut aussi un clavardage avec les autres joueurs de la partie. Une fois la partie terminée, une vue de résumé de la partie sera accessible qui permet d'exporter les dessins appréciés en format GIF, ainsi un "slideshow" d'un ou plusieurs dessins pourra être sauvegardée. Le serveur permet la communication entre les différents clients à l'aide d'une infrastructure basée sur les sockets de la librairie SocketIO et des requêtes http. Toute la communication comme les messages et les évènements de parties sont gérés par les sockets tandis que toutes les données qui requièrent des informations sauvegardées dans nos base de données sont gérées par les requêtes http faites au serveur.

2.2.2. Interfaces matérielles

L'application sera supportée par un client lourd sur PC et par un client léger sur tablette Android. Le client lourd utilisera un clavier et une souris pour les entrées, alors que le client léger utilisera un écran tactile et un clavier virtuel. Pour les extrants du système, le client lourd nécessite un système de son ainsi qu'un écran pour visualiser l'application et le client léger doit utiliser l'écran tactile de la tablette comme écran de sortie.

2.2.3. Interfaces logicielles

Pour la conception, les cadriciels de programmation utilisés pour les différentes sphères de la création seront Angular dans une fenêtre Electron pour le client lourd, Kotlin pour le client léger sur mobile. La version PC sera développée sur Windows 10.

Ensuite, Jira sera utilisé dans le but de maintenir un calendrier ainsi qu'un rappel des tâches à accomplir. En complément, Gitlab sera notre outil principal pour le partage de fichiers ce qui permettra à tous les développeurs de garder une version à jour du projet localement.

2.2.4. Interfaces de communication

Le client léger et le client lourd communiqueront via internet avec le serveur afin d'envoyer et de recevoir des données en temps réel. La connexion physique des appareils clients à internet peut être établie de plusieurs façons (ethernet, Wi-Fi, réseau cellulaire). Pour assurer une performance optimale, le serveur devra être connecté à internet via ethernet.

2.3. Contraintes générales

Le jeu doit avoir une bonne fluidité lors de l'utilisation et l'interface doit être cohérente entre chacun des joueurs sur le réseau et dans la partie.

Pour le client lourd : L'utilisateur du client lourd doit utiliser un poste Windows 7 ou plus récent ayant accès à plus de 4GB de mémoire vive.

Pour le client léger : L'utilisateur du client léger doit être sur une tablette Android ayant un écran de 10.1 pouces ou plus ayant une résolution d'au moins 1920 x 1200 pixels. De plus, le processeur de cette tablette doit être au moins équivalent au Exynos 7904A. Cette tablette doit aussi avoir 2GB de mémoire vive et au moins 1GB d'espace de stockage disponible. Finalement, la version d'Android minimale supportée par ce projet est 9.0 (Pie).

2.4. Hypothèses et dépendances

Pour accéder au jeu en ligne, l'utilisateur doit être connecté à l'internet et posséder des connaissances générales sur l'utilisation d'un ordinateur. Il doit posséder une adresse courriel pour pouvoir créer un compte dans l'application. L'utilisateur doit comprendre la langue française puisque c'est l'unique langage et le langage par défaut de l'application.

3. Exigences fonctionnelles

3.1 Lobby (Essentielle)

3.1.1 Le système doit permettre au joueur de voir la liste des joueurs avec leur équipe associée s'ils en ont une selon le mode de jeu.

3.1.2 Le système doit permettre au joueur de quitter le lobby afin de retourner au menu principal.

3.1.3 Le système doit permettre à l'hôte de débuter la partie lorsque le nombre minimum de joueurs pour débuter est atteint.

3.1.4 Le système doit ajouter le joueur au canal de discussion de la partie lorsqu'il rejoint un lobby.

3.1.5 Le système doit permettre au propriétaire de la partie de changer l'accessibilité de la partie entre privé et public.

3.1.6 Le système doit permettre au propriétaire d'enlever manuellement des joueurs réel de la partie.

3.2 Modes de jeu (Essentielle)

3.2.1 Classique

3.2.1.1 Le système doit séparer les joueurs en deux équipes.

3.2.1.2 Le système doit avoir deux équipes d'au maximum deux joueurs pour ce mode de jeu.

3.2.1.3 Le système doit permettre d'ajouter, d'enlever et d'avoir au maximum un joueur virtuel par équipe.

3.2.1.4 Le système donne, dans l'équipe active, le rôle de dessinateur à une personne et le rôle de devineur à l'autre.

3.2.1.5 Le système doit changer l'équipe active après chaque manche et il doit inverser les rôles des joueurs de celle-ci entre dessinateur et devineur entre chaque rotation d'équipes.

3.2.1.6 Le système doit permettre à tous les joueurs de voir le dessin en temps réel.

3.2.1.7 Le système donne, durant la manche, de un à trois essais, dépendamment de la difficulté de la partie, au devineur pour deviner le mot associé au dessin

3.2.1.8 Le système donne le droit aux devineurs d'effectuer leurs essais uniquement durant la période de temps alloué de la manche ou de la période de réplique.

3.2.1.9 Le système accorde un point à l'équipe du devineur si sa réponse est correcte, puis exécute un changement d'équipe active.

3.2.1.10 Le système n'accorde pas de point à l'équipe active lorsque les réponses du devineur sont incorrectes ou le temps limite de la manche est épuisé et donne le droit de réplique à l'équipe passive.

3.2.1.11 Le système donne un essai à l'équipe passive lors du droit de réplique, puis il y a un changement d'équipe active à la fin du droit de réplique.

3.2.1.12 Le système n'accorde pas de point à l'équipe passive lors du droit de réplique si les réponses sont incorrectes ou le temps limite de la période de réplique est épuisé

3.2.1.13 Le système affiche le score de chaque équipe en temps réel.

3.2.1.14 Le système affiche le temps restant de la manche ou de la période de droit de réplique en temps réel.

3.2.1.15. Le système donne toujours le rôle de dessinateur au joueur virtuel si l'équipe active en possède un et il ne peut jamais assumer le rôle de devineur.

3.2.2 Sprint Solo

3.2.2.1 Le système doit avoir un seul joueur réel pour ce mode de jeu.

3.2.2.2 Le système possède toujours un seul joueur virtuel qui ne peut pas être enlevé et celui-ci a toujours le rôle de dessinateur.

3.2.2.3 Le système doit changer de mot et d'image lorsque le nombre d'essaie maximal est atteint.

3.2.2.4 Le système doit effacer l'espace de dessin lorsque le nombre d'essaie maximal est atteint ou le temps limite est atteint.

3.2.2.5 Le système doit afficher un temps descendant représentant le temps restant pour la partie qui diffère selon la difficulté.

3.2.2.6 Le système doit afficher le nombre d'essaie restant pour le mot.

3.2.2.7 Le système doit afficher le score du joueur.

3.2.2.8 Le système doit donner un point au joueur s'il devine le mot.

3.2.2.9 Le système doit donner plus de temps selon le niveau de difficulté lorsque le joueur devine le mot.

3.2.2.10 Le système doit mettre fin à la partie lorsque le temps tombe à zéro.

3.2.3 Sprint coopératif

3.2.3.1 Le système doit avoir de 2 à 4 joueurs réels pour ce mode de jeu..

3.2.3.2 Le système possède toujours un seul joueur virtuel qui ne peut pas être enlevé et celui-ci a toujours le rôle de dessinateur.

3.2.3.3 Le système doit changer de mot et d'image lorsque le nombre d'essaie collectif maximal est atteint.

3.2.3.4 Le système doit effacer l'espace de dessin lorsque le nombre d'essaie collectif maximal est atteint ou le temps limite est atteint.

3.2.3.5 Le système doit afficher un temps descendant représentant le temps restant pour la partie qui diffère selon la difficulté.

3.2.3.6 Le système doit afficher le nombre d'essaie collectif restant pour le mot.

3.2.3.7 Le système doit afficher le score collectif des joueurs.

3.2.3.8 Le système doit donner un point aux joueurs s'ils devinent le mot.

3.2.3.9 Le système doit donner plus de temps selon le niveau de difficulté lorsque les joueurs deviennent le mot.

3.2.3.10 Le système doit mettre fin à la partie lorsque le temps tombe à zéro.

3.3 Création de paire mot-image sur client lourd(Chaque mode de création contient les exigences des modes antérieures sauf indication contraire)

3.3.1 Le système doit permettre à l'utilisateur de choisir son mode de création de dessin entre le mode manuel et le mode assisté.

3.3.2 Manuelle 1 ***(Essentielle)***

3.3.2.1 Le système doit afficher une interface afin de permettre à l'utilisateur d'entrer le mot ou l'expression pour le dessin qu'il va faire.

3.3.3.2.2 Le système doit afficher une interface afin de permettre à l'utilisateur d'entrer un à plusieurs indices pour le dessin qu'il va faire.

3.3.3.2.3 Le système doit afficher une zone de dessin afin de permettre à l'utilisateur de créer un dessin à l'aide des outils disponibles (crayon, efface, couleur, etc).

3.3.3.2.4 Le système doit permettre à l'utilisateur de sélectionner le niveau de difficulté associé au dessin.

3.3.3 Manuelle 2 ***(Essentielle)***

3.3.3.1 Le système doit permettre à l'utilisateur de choisir le mode de dessin.

3.3.3.2 Le système possède le mode conventionnel

3.3.3.2.1 Le système s'occupe de dessiner chaque trait dans le même ordre que lors de la création du dessin par l'utilisateur dans ce mode.

3.3.3.3 Le système possède le mode aléatoire

3.3.3.3.1 Le système s'occupe de dessiner chaque trait dans un ordre aléatoire pour chaque partie.

3.3.3.4 Le système possède le mode panoramique

3.3.3.4.1 Le système s'occupe de dessiner chaque trait en ordre de de position sur les axes cartésiens.

3.3.3.4.2 Le système doit permettre de choisir la direction de progression du dessin (droite à gauche, gauche à droite, haut à bas, bas à haut).

3.3.3.5 Le système possède le mode centré

3.3.3.5.1 Le système s'occupe de dessiner chaque trait selon l'ordre de leur distance au centre de l'image.

3.3.3.5.2 Le système doit permettre de choisir la direction de progression du dessin (intérieur vers extérieur, extérieur vers intérieur).

3.3.4 Assistée 1 ***(Essentielle)***

3.3.4.1 Le système doit permettre à l'utilisateur d'importer une image sous format bmp, jpg ou png.

3.3.4.2 Le système doit enlever le choix de mode de dessin conventionnel à l'utilisateur.

3.3.4 Assistée 2 ***(Souhaitable)***

3.3.4.1 Le système doit afficher une zone d'affichage de dessin.

3.3.4.2 Le système doit remplacer la zone pour entrer un mot ou une expression par une zone d'affichage de mot ou expression.

3.3.4.3 Le système doit afficher une combinaison mot/expression et dessin à l'utilisateur.

3.3.4.4 Le système doit permettre à l'utilisateur de changer de combinaison affichée.

3.3.5 Assistée 3 ***(Souhaitable)***

3.3.5.1 Le système doit permettre à l'utilisateur d'entrer un mot ou une expression dans une barre de recherche.

3.3.5.2 Le système doit afficher une série d'image à l'utilisateur selon le mot ou l'expression entrée.

3.3.5.3 Le système doit permettre à l'utilisateur de visionner la prochaine série d'images.

3.3.5.4 Le système doit permettre à l'utilisateur de sélectionner une image parmi la série d'images.

3.3.5.5 Le système doit passer l'image sélectionnée dans l'engin de conversion d'image afin d'afficher le dessin obtenu.

3.4 Personnalité des joueurs virtuels (Essentielle)

3.4.1 Au début de chaque partie, les joueurs virtuels envoient parfois un message dans le canal de discussion de la partie.

3.4.2 À la fin de chaque partie, les joueurs virtuels envoient parfois un message dans le canal de discussion de la partie.

3.4.3 Les joueurs virtuels ont des personnalités différentes à chaque partie qui influencent les messages qu'ils envoient. Les personnalités sont : agressif, gentleman et bizarre.

3.4.4 Les joueurs virtuels ont une chance d'envoyer un message qui indique un changement au niveau de la vitesse de dessinage.

3.4.5 Les joueurs virtuels ont une chance d'envoyer un message lorsque le devineur à une mauvaise, bonne ou presque bonne réponse.

3.5 Effets visuels et sonores (Essentielle)

3.5.1 Le système doit avoir une réaction sonore positive lorsqu'un joueur réussit un essai.

3.5.2 Le système doit avoir une réaction sonore négative lorsqu'un joueur échoue un essai.

3.5.3 Le système doit afficher un effet de particule lorsqu'un joueur gagne ou perd la partie en mode classique.

3.5.4 Le système doit faire un son à chaque seconde lorsqu'il reste 10 secondes ou moins sur un minuteur.

3.6 Page de connexion (Essentielle)

3.6.1 Le système doit permettre à l'utilisateur d'entrer un nom d'utilisateur.

3.6.2 Le système doit permettre à l'utilisateur d'entrer un mot de passe.

3.6.3 Le système doit permettre à l'utilisateur de se connecter à l'application seulement si les champs de nom d'utilisateur et de mot de passe sont remplis et validés par le serveur.

3.6.4 Le système doit notifier l'utilisateur si les informations entrées sont invalides.

3.6.5 Le système doit empêcher la connexion de l'utilisateur si les informations entrées sont invalides.

3.6.6 Le système doit permettre à l'utilisateur de réinitialiser son mot de passe.

*3.6.7 Réinitialisation de mot de passe **(Souhaitable)***

3.6.7.1 Le système doit permettre à l'utilisateur d'entrer l'adresse courriel associée à son compte.

3.6.7.2 Le système doit envoyer un courriel électronique à l'adresse courriel entré avec un lien de réinitialisation de mot de passe.

3.6.7.3 Le système doit ouvrir l'application lorsque l'utilisateur clique sur le lien.

3.6.7.4 Le système doit s'ouvrir sur une page de réinitialisation de mot de passe.

3.6.7.5 Le système doit permettre à l'utilisateur d'entrer son nouveau mot de passe.

3.6.7.6 Le système doit obliger l'utilisateur à remplir un champ de confirmation de mot de passe.

3.6.7.7 Le système doit refuser le nouveau mot de passe s'il correspond à un vieux mot de passe.

3.7 Système de clavardage (Essentielle)

3.7.1 Le système doit avoir une boîte de clavardage après la connexion de l'utilisateur.

3.7.2 Le système doit permettre à l'utilisateur de détacher la boîte de clavardage en fenêtre de clavardage.

3.7.3 Le système doit permettre à l'utilisateur de rattacher la fenêtre de clavardage en boite de clavardage dans l'application.

3.7.4 Le système doit rattacher la fenêtre de clavardage en boite de clavardage dans l'application lorsque la fenêtre est fermée.

3.7.5 Le système doit ajouter l'utilisateur au canal de clavardage de la partie lorsqu'il est ajouté à celle-ci.

3.7.6 Le système doit permettre aux utilisateurs de créer des canaux de discussion.

3.7.7 Le système doit permettre aux utilisateurs de supprimer des canaux de discussion.

3.7.8 Le système doit permettre à l'utilisateur de joindre un canal créé en entrant son nom dans une barre de recherche.

3.7.9 Le système doit permettre à l'utilisateur de joindre un canal créé à partir d'une liste affichant les canaux de discussions disponibles.

3.7.10 Le système doit permettre à l'utilisateur de quitter un canal de discussion qu'il a rejoint.

3.7.11 Le système doit permettre à l'utilisateur d'afficher l'historique de clavardage complet du canal de discussion.

3.7.12 Le système doit permettre à l'utilisateur de créer un canal de discussion privée avec un utilisateur de sa liste d'amis.

3.8 Page de choix de partie (Essentielle)

3.8.1 Le joueur doit pouvoir créer une partie selon le mode de jeu (Classique, solo ou coop), la difficulté (facile, intermédiaire ou difficile) ainsi que le niveau d'accessibilité : public ou privé.

3.8.2 Le système doit afficher chacune des parties disponibles que le joueur peut rejoindre.

3.8.3 Le système doit afficher chacun des joueurs dans les différentes parties affichées.

3.9 Fin de partie (Souhaitable)

3.9.1 Le système doit présenter à l'utilisateur une page qui lui indique que la partie est terminée.

3.9.2 Le système doit permettre à l'utilisateur de sauvegarder n'importe quel dessin effectué durant la partie en image.

3.9.3 Le système doit permettre à l'utilisateur de sauvegarder un GIF récapitulatif de la création de son dessin.

3.9.4 Le GIF récapitulatif doit montrer les étapes de la création du dessin image par image.

3.9.5 Le GIF récapitulatif doit avoir une longueur prédéterminée pour éviter des GIFs trop longs ou trop courts.

3.10 Menu Principal (Essentielle)

3.10.1 Le système doit permettre de joindre ou créer une partie avec les caractéristiques désirées.

3.10.2 Le système doit permettre de consulter son profil d'utilisateur.

3.10.3 Le système doit permettre d'accéder à la création d'une paire mot-image.

3.11 Profil Utilisateur (Essentielle)

3.11.1 Le système permet à l'usager de consulter et modifier la partie privée de son compte, qui contient un prénom, un nom et un mot de passe.

3.11.2 Le système permet à l'usager de consulter et modifier la partie publique de son compte, qui contient un pseudo ainsi qu'un avatar.

3.11.3 Le système permet à l'usager de consulter ses statistiques de jeu tel que le nombre de parties jouées, le pourcentage de victoires, le temps moyen d'une partie et le temps total passé à jouer.

3.11.4 Le système permet à l'usager de consulter les dates et heures de connexion / déconnexion et l'historique des parties jouées de ses amis ainsi que son profil personnel.

3.12 Liste d'ami (Essentielle)

3.12.1 Le système permet d'envoyer et de recevoir des demandes d'amitiés.

3.12.2 Le système permet de consulter sa liste d'amis.

3.12.3 La liste d'amis permet d'inviter des amis dans une partie.

3.12.4 La liste d'amis permet de consulter le profil de ses amis.

3.13 Tutoriel

3.13.1 Dans le menu principal de l'application, il doit y avoir une section tutoriel qui permet de se familiariser à l'interface ainsi qu'aux mécaniques du jeu, celui-ci est interactif et requiert à l'usager d'interagir.

4. Exigences non-fonctionnelles

4.1. Utilisabilité

4.1.1. Utilisation de l'application pour un usager normal

L'utilisation de l'application devrait être un apprentissage rapide pour un usager qui est nouveau au système de jeu. Soit, après le tutoriel interactif l'usager devrait être capable de joindre une partie, dessiner, deviner et utiliser les autres fonctionnalités comme la liste d'amis et le système de clavardage.

4.1.2 Utilisation de l'application pour un usager spécialisé

L'utilisation de l'application devrait être assez intuitive pour qu'un usager spécialisé puisse utiliser l'application correctement sans même avoir fait le tutoriel interactif.

4.2. Fiabilité

4.2.1. Disponibilité des serveurs

Les serveurs devraient être disponibles 99% du temps.

4.3. Performance

4.3.1. Temps de réponse du serveur

Le temps de réponse du serveur doit être en dessous 150 millisecondes.

4.3.2. Performance du dessin

La zone de dessin devra répondre du côté client sans ralentissement même s'il y a plus de 100 traits dans le dessin.

4.4. Maintenabilité

4.4.1. Standard d'écriture de code client lourd

Le client lourd doit se conformer aux normes d'écritures de code en Angular et Typescript enseignées à Polytechnique Montréal.

4.4.2 Standard d'écriture de code client léger

Le client léger doit se conformer aux normes d'écriture de code en Kotlin enseignées à Polytechnique Montréal.

4.4.3. Standard de ré-usabilité

L'application, autant au niveau de l'interface que du serveur, doit être créée de façon à être utilisée pour des expansions futures au niveau du nombre de modes de jeu ainsi que de l'augmentation potentielle du trafic d'usagers sur les serveurs.

4.5. Contraintes de conception

4.5.1. Technologies de développement du client lourd

Le client lourd est produit avec Angular.

4.5.2 Langage de programmation du client lourd

Les langages utilisés pour le client lourd sont le Typescript, le HTML et le SCSS.

4.5.3 Technologie de déploiement du client lourd

L'application web est transformée en application native avec Electron.

4.5.4 Technologies de développement du client léger

Le client léger est produit avec l'aide de Android Studio.

4.5.5 Langage de programmation du client léger

Le langage utilisé pour notre client léger sera le Kotlin.

4.5.6 Technologies de déploiement du serveur

Le serveur est produit avec Node à l'aide de la librairie Express.

4.5.7 Technologie de communication serveur-client

La librairie de communication Socket.io sera utilisée pour la communication entre le serveur et les différents clients.

4.5.8 Hébergement du serveur

Le serveur est hébergé chez un membre de l'équipe, nous aurons un serveur production et un serveur de développement.

4.6. Sécurité

4.6.1. Encryption de la connexion

Lors de la connexion au compte de l'utilisateur, le mot de passe envoyé au serveur doit être crypté.

4.6.2 Encryption des messages

Lors de l'envoi de messages à l'aide du système de communication et messagerie, les messages envoyés doivent être cryptés.

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

4.7.1 Documentation usager en ligne

Le système ne possède pas de documentation d'usager en ligne disponible aux utilisateurs.

4.7.2 Système d'assistance

Le système possède un tutoriel dans l'application en guise de système d'assistance pour les usagers.