

---

**Équipe 106**

---

**Projet «Fais moi Dessin»  
Plan de projet**

**Version 1.0**

## Historique des révisions

Date	Version	Description	Auteur
2020-09-14	1.0	Version initiale	Sohou Aubierge Gloria
2020-09-14	1.0	Première version de l'échéancier	Sohou Aubierge Gloria
2020-09-17	1.0	Rédaction de la solution proposée	Sophie Dorval
2020-09-19	1.0	Rédaction des risques, de l'introduction et des biens livrables du projet	Sohou Aubierge Gloria
2020-09-21	1.0	Rédaction de la section gestion des exigences et de l'entente contractuelle	Sophie Dorval
2020-09-21	1.0	Rédaction de l'échéancier	Tous
2020-09-22	1.0	Révision de l'échéancier et ajout de profil individuel	Kevin Nguyen
2020-09-23	1.0	Révision et correction de faute d'orthographe	Sophie Dorval
2020-10-02	1.0	Révision finale	Tous

# Table des matières

<b>1. Introduction</b>	<b>4</b>
<b>2. Énoncé des travaux</b>	<b>4</b>
2.1. Solution proposée	4
2.2. Hypothèses et contraintes	6
2.3. Biens livrables du projet	7
<b>3. Gestion et suivi de l'avancement</b>	<b>7</b>
3.1. Gestion des exigences	7
3.2. Contrôle de la qualité	7
3.3. Gestion de risque	8
3.4. Gestion de configuration	10
<b>4. Échéancier du projet</b>	<b>10</b>
<b>5. Équipe de développement</b>	<b>16</b>
<b>6. Entente contractuelle proposée</b>	<b>17</b>

# Plan de projet

## 1. Introduction

Le présent document fait suite à l'appel d'offres pour la mise en place du projet "Fais-moi un dessin" de Polytechnique Montréal. Il s'agit du plan du projet proposé par l'équipe 106.

Tout d'abord, nous exposerons un énoncé des travaux en décrivant brièvement la solution proposée par l'équipe, les hypothèses et contraintes ainsi que les biens livrables du projet. Ensuite, nous présenterons notre solution pour la gestion et le suivi de l'avancement. De plus, nous présenterons l'échéancier du projet en prenant le soin d'expliquer les principaux lots de travail puis l'effort, les dates de début et de fin associés à ces lots. Enfin, nous présenterons l'équipe de développement et l'entente contractuelle proposée.

## 2. Énoncé des travaux

### 2.1. Solution proposée

Notre solution est de développer l'application 'Fais moi Dessin' qui est un logiciel de jeu de dessin et de devinettes multijoueur. Cette application sera disponible sur un PC en tant que client lourd et sur une tablette Android en tant que client léger. Pour le client lourd, l'application offrira à son utilisateur de nombreuses fonctionnalités. Il y aura une fonctionnalité de chat intégrée dans l'application. L'utilisateur pourra démarrer un nouveau canal de discussion et rejoindre ou supprimer un canal existant. L'utilisateur aura également la possibilité de voir l'historique du canal de discussion. De plus, l'utilisateur devra se connecter à l'application et aura ainsi accès à son profil. Son profil comporte son nom, ses statistiques d'utilisation du jeu et son historique de connexion ou déconnexion, qui sont visibles uniquement à lui seul. Tout utilisateur peut accéder au pseudo et à l'avatar de tout autre utilisateur. Le jeu offre le mode de jeu mêlée générale. Cela implique qu'un minimum de deux joueurs humains s'affrontent pour deviner le plus de mots afin d'accumuler le plus de points possible. Le dessinateur changera à chaque tour. De plus, l'utilisateur pourra ajouter des mots à la banque de jeu en donnant toutes les informations nécessaires telles que la difficulté, le dessin, le mot et les indices. Les joueurs virtuels rejoindront les jeux en tant que dessinateur seulement. Ces joueurs virtuels auront des personnalités différentes. Enfin, un tutoriel non interactif sera accessible aux utilisateurs à tout moment pour faciliter l'apprentissage du fonctionnement de l'application. Pour le client léger, l'application offrira essentiellement les mêmes fonctionnalités. Cependant l'utilisateur ne pourra pas ajouter de mots à la banque de jeu et des sons de notification seront ajoutés à la fonctionnalité de chat.

Les fonctionnalités sont détaillées de façon plus approfondie dans le document de spécifications des requis du système (SRS).

## 2.2. Hypothèses et contraintes

Cette section présente les hypothèses et contraintes du plan de projet.

### 2.2.1. Hypothèses

- On suppose que tous les membres de l'équipe travaillent sur le projet un minimum de 6 heures par semaine sur le projet.
- Les membres de l'équipe disposent d'un ordinateur portable avec une connexion internet.
- On suppose qu'aucun membre n'abandonnera le projet.
- On suppose qu'il y a une cohésion d'équipe favorisant l'avancement des tâches.

### 2.2.2. Contraintes

- La solution proposée doit prendre en compte toutes les caractéristiques obligatoires listées dans le « Document de vision ».
- Certains membres de l'équipe n'avaient jamais utilisé les technologies comme GraphQL, NodeJs... nécessaires au projet.
- L'équipe doit respecter les normes de programmation minimales afin d'uniformiser le code.
- Le code du projet doit être documenté.
- Les fonctionnalités et l'interface utilisateur du jeu doivent être cohérentes pour le client lourd et pour le client léger.
- Les dates de remise des différents livrables présentées à la section 2.3 ne peuvent pas être repoussées.
- L'équipe doit se réunir pendant une heure par semaine en dehors des heures du cours et du laboratoire pour des mises à jour sur l'avancement des tâches de chaque membre.
- Les membres de l'équipe doivent se connecter au groupe de discussion Discord afin de favoriser d'éventuelles prises de décisions urgentes nécessaires à l'avancement des tâches.
- Le client lourd doit fonctionner sous Windows 10.
- Le client léger doit fonctionner sous Android.

## 2.3. Biens livrables du projet

Le tableau suivant tiré de l'avis d'appel d'offre fourni par l'autorité contractante présente les artefacts du projet avec leurs dates prévues de livraison.

Livrable	Artéfacts	Date de remise
Réponse à l'appel d'offres	<ul style="list-style-type: none"><li>• Plan de projet</li><li>• Spécification des requis du système (SRS)</li><li>• Liste d'exigences</li><li>• Document d'architecture logicielle</li><li>• Protocole de communication client lourd-serveur</li><li>• Protocole de communication client léger-serveur</li></ul>	2 Octobre 2020
Remise finale	<ul style="list-style-type: none"><li>• SRS mis à jour</li><li>• Protocole de communication client lourd-serveur mis à jour</li><li>• Protocole de communication client léger-serveur mis à jour</li><li>• Architecture logicielle mis à jour</li><li>• Plan de tests</li><li>• Résultats des tests</li><li>• Code source et exécutable du produit final</li></ul>	2 Décembre 2020

### **3. Gestion et suivi de l'avancement**

#### **3.1. Gestion des exigences**

Nous avons commencé par élaborer en détail les exigences obligatoires et souhaitables qui seront implémentées dans notre application dans le document de spécification des requis (SRS). Ainsi chaque membre de l'équipe peut se référer à ce document en cas de doute. Chaque semaine, en dehors de la réunion obligatoire avec l'enseignant, une réunion virtuelle supplémentaire a lieu le lundi ou le mardi pour discuter de l'avancement du projet dans son ensemble et de l'état d'avancement des tâches de chaque membre de l'équipe. Cela nous permet également de répondre aux questions et suggestions des autres membres de l'équipe. Le suivi des exigences se fera dans Jira, principalement en utilisant la section sprint. Nous décomposons nos tâches en sous-tâches plus petites. De plus, cet outil permet de déplacer les tâches dans les catégories en cours et terminé pour nous permettre de suivre les multiples tâches assignées à différents membres de l'équipe.

#### **3.2. Contrôle de la qualité**

Le contrôle de la qualité se fera dès le plus bas niveau de codage grâce à la mise en place de système permettant de linter notre code dans nos deux dépôts de code sur Github. Des tests unitaires seront implémentés également sur tous les clients ainsi que sur le serveur. Un suivi sur le lint et les tests sera fait automatiquement sur tous les push fait au dépôt principal grâce à la mise en place d'un pipeline implémenté avec Github Actions,.

Dans le but de conserver une seule source de vérité dans notre dépôt principal chaque nouveau feature sera développé sur une branche unique avant d'être mergé avec le code de la branche Master. Un pull request nécessitant la révision et l'approbation d'au moins un autre membre de l'équipe sera nécessaire, sinon les changements seront refusés en indiquant un message constructif de la raison du refus.

Dans le cas d'un bug, le membre de l'équipe qui a travaillé sur cette partie va devoir créer une branche hotfix/nom-du-feature et essayer de régler le problème.

#### **3.3. Gestion de risque**

Le tableau suivant décrit les différents risques liés au projet.

L'ampleur est l'estimation sur une échelle de 1 à 10 de la criticité du risque basée sur la probabilité d'occurrence du risque, ainsi que ses impacts. L'impact d'un risque est une échelle définissant la portée du risque :

- C pour critique (affecte le projet en entier),
- E pour élevé (affecte les fonctionnalités principales du système),
- M pour moyen (devrait être maîtrisable en appliquant une stratégie d'atténuation adéquate)
- F pour faible (l'acceptation du risque est une stratégie envisageable).

Les facteurs donnent les aspects métriques du système pouvant être compromis. Enfin dans la stratégie de gestion, on présente les mesures à prendre afin de gérer le risque.

#### R1- Risque lié à la mauvaise interprétation des requis du système

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
10	Les exigences du système sont mal interprétées. Il peut s'agir du fait que tous les membres de l'équipe n'ont pas une compréhension unique des requis. Cela conduit à des incohérences dans le produit final, à l'implémentation de fonctionnalités non demandées par le client ou incomplètes.	C	Exigences	<p>L'équipe doit prendre le temps de bien comprendre les requis. Ce travail de compréhension doit être fait avant toute démarche d'implémentation ou d'écriture de code.</p> <p>Si des ambiguïtés persistent sur les requis, l'équipe doit se tourner vers l'autorité contractante pour plus d'explications.</p> <p>Dans le cas où, plusieurs membres de l'équipe travaillent sur une même fonctionnalité, ils doivent communiquer afin de rester d'assurer la cohérence dans les fonctionnalités ainsi qu'au niveau du code.</p>

#### R2 - Risque lié au manque de cohésion dans l'équipe

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	<p>Le manque de cohésion se manifeste par une mauvaise communication voire une communication inexistante au sein de l'équipe.</p> <p>Il peut aussi se manifester par la présence régulière de conflits d'équipe.</p>	M	Efficacité de l'équipe	<p>L'équipe met en place une bonne stratégie de communication.</p> <p>Les membres de l'équipe doivent idéalement s'entraider entre eux et encourager le partage de connaissance entre pairs.</p> <p>L'équipe pourrait organiser au cours du projet des rencontres ou des activités (virtuelles de préférence) afin de favoriser un climat de cohésion.</p>

#### R4- Risque lié aux délais limités

Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	<p>La mauvaise gestion du temps et les situations imprévues au cours du projet peuvent conduire à un manque de temps pour implémenter toutes les fonctionnalités.</p> <p>Ce risque est aussi lié à une mauvaise estimation du temps nécessaire pour faire certaines tâches.</p>	M	Complétude des fonctionnalités	<p>L'équipe a très tôt mis en place un plan global de réalisation du projet qui sera mis à jour régulièrement en fonction de l'avancement du projet. Cela permet d'avoir une vue globale sur le projet et les échéanciers.</p>

R5 - Risque lié au manque d'expertise de l'équipe				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	La plupart des membres de l'équipe n'avait pas auparavant utilisé les technologies du projet. Ce qui implique que les membres auront besoin d'une période d'apprentissage.	M	Fonctionnalité	L'équipe pourra compter sur l'entraide entre les pairs impliquant que les coéquipiers connaissant une technologie vont essayer d'orienter les autres vers les bonnes ressources .  De plus, l'équipe a décidé que les membres doivent commencer le plus tôt possible à se familiariser avec les nouveaux outils avant le démarrage des activités d'écriture de code.

R6 -Risques technologiques				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	Des problèmes majeurs pourraient survenir en utilisant les nouvelles technologies pour la réalisation du projet .Il peut s'agir par exemple de la complexité dans l'utilisation d'une technologie ou de la l'incompatibilité entre différentes technologies. Ceci pourrait ralentir le projet.  Par ailleurs, des risques peuvent être liés à la défaillance de la communication entre le serveur et les clients.	E	Exigences et Performance	Au cours de la session, l'équipe pourra toujours décider d'utiliser des technologies plus adaptées aux différentes tâches afin d'une implémentation efficace.  Pour gérer ces risques, l'équipe réalise le maximum des tests pour s'assurer que la communication entre les clients et le serveur soit fluide.

### 3.4. Gestion de configuration

La gestion de version et la configuration logicielle que nous allons mettre en place vont nous permettre d'avoir un produit incrémental.

Nous allons utiliser un dépôt git pour la soumission de code afin de garder une trace de toutes les modifications effectuées. Nous nous sommes entendus sur certaines bonnes pratiques qui vont nous permettre de bien faire la gestion de notre dépôt git.

Une branche est créée pour chacune des fonctionnalités décrites dans le document SRS, en suivant une convention de nommage qui est: Feature/nom-du-feature. Une branche par correction de bug qui devrait se nommer: hotfix/nom-du-bug. Lorsqu'un changement est effectué sur une branche, il faut mettre un message de commit constructif et significatif. Sans lire la partie du code modifiée, le message devrait être suffisant pour comprendre la modification apportée au code. Également, nous allons mettre des messages en anglais qui doivent commencer par le nom du feature.

Une fois qu' une branche est prête au déploiement vers la branche 'master', il faut au moins une personne qui vérifie le travail effectué pour confirmer le travail effectué.



#### 4. Échéancier du projet

Lots de travail	Début-Fin	Objectifs	Temps requis (heures-personne)
Lot 1	3-9 Sept 2020	Introduction au cours (10h) Formation de l'équipe et règles générales de travail (10h)	20h
Lot 2	10 -16 Sept 2020	<ul style="list-style-type: none"> <li>- Finir le SRS afin de savoir exactement le travail à faire (20h)</li> <li>- Début des réflexions sur l'architecture du serveur et du client (20h)</li> <li>- Planification de la session (15h)</li> <li>- Implémentation générale du modèle MVVM du client léger (5h)</li> </ul>	60h
Lot 3	17 -23 Sept 2020	<ul style="list-style-type: none"> <li>- Plan de projet (6h)</li> <li>- Document d'architecture logicielle (8h)</li> <li>- Commencer à travailler sur protocole de communication (6h)</li> <li>- Commencer à travailler sur les prototypes (4h)</li> </ul> <p>Client Léger:</p> <ul style="list-style-type: none"> <li>- Début du codage de l'interface dessin <ul style="list-style-type: none"> <li>- Implémentation du canvas (4h)</li> <li>- Implémentation du crayon (4h)</li> <li>- Implémentation du color-picker (3h)</li> <li>- Implémentation du UI initial incluant la surface de dessin et les outils (5h)</li> </ul> </li> </ul> <p>Serveur:</p> <ul style="list-style-type: none"> <li>- Début de l'implémentation de l'architecture</li> </ul> <p>Client Lourd:</p> <ul style="list-style-type: none"> <li>- Début de l'implémentation de l'architecture <ul style="list-style-type: none"> <li>- Retrait des fonctionnalités non-nécessaires du projet 2 (5h)</li> <li>- Implémenter le framework Electron (5h)</li> </ul> </li> </ul>	60h Léger:16h Lourd:10h Serveur:10h Autre: 24h
Lot 4	24 -30 Sept 2020	<ul style="list-style-type: none"> <li>- Finaliser le document protocole de communication (5h)</li> <li>- Intégrer et Finaliser le prototype (5h)</li> <li>- Réviser et corriger tous les documents (5h)</li> </ul> <p>Client Léger:</p>	150h Léger:45h Lourd:45h Serveur:45h Autre: 15h

		<ul style="list-style-type: none"> <li>- Codage de l'interface dessin <ul style="list-style-type: none"> <li>- Implémentation de l'efface (5h)</li> <li>- Implémentation de la grille (5h)</li> <li>- Implémentation du undo-redo (5h)</li> <li>- Implémentation d'un prototype de communication <ul style="list-style-type: none"> <li>- Implémenter le langage de requête Apollo-GraphQL pour permettre la connection entre le client et le serveur (15h)</li> <li>- Envoyer des requêtes au serveur et obtenir la réponse (15h)</li> </ul> </li> </ul> </li> </ul> <p>Serveur:</p> <ul style="list-style-type: none"> <li>- Implémentation d'un prototype de communication <ul style="list-style-type: none"> <li>- Recevoir une requête du client lourd et une du client léger puis envoyer une réponse à chacun(10h)</li> <li>- Gérer l'envoi et la réception de message des clients(10h)</li> <li>- Gérer la création des canaux de discussion</li> <li>- Gérer la création de profil(10h)</li> <li>- Gérer la connexion au profil et la déconnexion(15h)</li> </ul> </li> </ul> <p>Client Lourd:</p> <ul style="list-style-type: none"> <li>- Implémentation d'un prototype de communication <ul style="list-style-type: none"> <li>- Implémenter le langage de requête Apollo-GraphQL pour permettre la connection entre le client et le serveur (20h)</li> <li>- Envoyer plusieurs différentes requêtes au serveur puis recevoir la réponse (25h)</li> </ul> </li> </ul>	
<b>Lot 5</b>	<b>1-14 Oct 2020</b>	<p><b>Premier livrable: réponse à l'appel d'offre ( 2 Octobre)</b></p> <ul style="list-style-type: none"> <li>- Mettre en place les architectures des clients, et du serveur (15h)</li> </ul> <p>Client Lourd:</p> <ul style="list-style-type: none"> <li>- Début des features essentielles sur le client lourd <ul style="list-style-type: none"> <li>- Implémenter le UI pour le menu principale permettant de naviguer entre les pages (5h)</li> <li>- Implémenter l'interface de clavardage (10h)</li> <li>- Implémenter l'authentification (15h)</li> <li>- Implémenter le profil utilisateur (10h)</li> </ul> </li> </ul> <p>Client Léger:</p> <ul style="list-style-type: none"> <li>- Début des features essentielles</li> </ul>	150h Léger:45h Lourd:45h Serveur:45h Autre: 15h

		<ul style="list-style-type: none"> <li>- Implémenter l'interface de clavardage (10h)</li> <li>- Implémenter le UI de navigation entre les fragments/layout (10h)</li> <li>- Implémenter la feature qui permet de créer, supprimer et rejoindre un canal de discussion(10h)</li> <li>- Implémenter l'authentification (5h)</li> <li>- Implémenter le profil utilisateur (10h)</li> </ul> <p>Serveur:</p> <ul style="list-style-type: none"> <li>- Suite des features sur le serveur <ul style="list-style-type: none"> <li>- Implémenter le système d'authentification (20h)</li> <li>- Implémenter les services de requêtes pour envoyer les informations sur le profil de l'utilisateur (25h)</li> </ul> </li> </ul> <p>Tests de toutes les fonctionnalités implémentées</p>	
Lot 6	15-29 Oct 2020	<ul style="list-style-type: none"> <li>- Intégration de la feature clavardage soit clavarder à même environnement pour les deux clients avec le serveur (8h)</li> <li>- Intégration du feature profil utilisateur pour les deux client avec le serveur (7h)</li> </ul> <p>Client lourd:</p> <ul style="list-style-type: none"> <li>- Suite des features essentielles <ul style="list-style-type: none"> <li>- Implémenter et finir le mode de jeu mêlée générale <ul style="list-style-type: none"> <li>- Implémenter la création d'une partie ou joindre une partie (15h)</li> <li>- Implémenter la saisi des mots (10h)</li> <li>- Implémenter le processus de dessin (20h)</li> <li>- Implémenter le système de pointage (5h)</li> </ul> </li> <li>- Création de jeu pair mot-image Manuelle <ul style="list-style-type: none"> <li>- Implémenter l'interface de saisi de mot-image (5h)</li> <li>- Implémenter le processus de dessin comme un joueur virtuel (10h)</li> </ul> </li> </ul> </li> </ul> <p>Client léger:</p> <ul style="list-style-type: none"> <li>- Suite des features essentielles <ul style="list-style-type: none"> <li>- Implémenter et finir le mode de jeu mêlée générale</li> </ul> </li> </ul>	180h Léger:50h Lourd:65h Serveur:50h Autre: 15h

		<ul style="list-style-type: none"> <li>- Implémenter la création d'une partie ou joindre une partie (15h)</li> <li>- Implémenter la saisi des mots (10h)</li> <li>- Implémenter le processus de dessin (20h)</li> <li>- Implémenter le système de pointage (5h)</li> </ul> <p>Serveur:</p> <ul style="list-style-type: none"> <li>- Suite des service de requêtes <ul style="list-style-type: none"> <li>- Implémentation des joueurs virtuels <ul style="list-style-type: none"> <li>- Implémenter l'ajout et supprimer des joueurs virtuel d'une partie (20h)</li> <li>- Implémenter le système de dessins (20h)</li> </ul> </li> <li>- Implémentation du système de pointage <ul style="list-style-type: none"> <li>- Implémenter l'assignation de point au joueur ayant saisi le mot (10h)</li> </ul> </li> </ul> </li> </ul> <p>Tests de toutes les fonctionnalités implémentées</p>	
Lot 7	29 -11 Nov 2020	<ul style="list-style-type: none"> <li>- Intégration du feature de mode de jeu mêlée générale pour les deux clients avec le serveur (15h)</li> </ul> <p>Client léger:</p> <ul style="list-style-type: none"> <li>- Suite des features essentielles <ul style="list-style-type: none"> <li>- Implémentation Mode de jeu mêlée générale (25h)</li> <li>- Implémentation du tutoriel non-interactif (5h) <ul style="list-style-type: none"> <li>- Définir le contenu du tutoriel</li> <li>- Implémenter l'interface du tutoriel</li> </ul> </li> <li>- Implémentation de trois groupes d'effets présents (10h) <ul style="list-style-type: none"> <li>- Implémenter les effets visuel et sonore lors de la saisi de mot</li> <li>- Implémenter bruit sonore lorsqu'il ne reste plus beaucoup de temps et lorsqu'on saisi mauvais mot</li> </ul> </li> <li>- Implémentation de l'historique du clavardage (10h) <ul style="list-style-type: none"> <li>- Implémenter l'interaction entre le serveur et le client et obtenir l'historique</li> </ul> </li> </ul> <p>Client lourd:</p> </li></ul>	<p>160h</p> <p>Léger:50h</p> <p>Lourd:50h</p> <p>Serveur:45h</p> <p>Autre: 15h</p>

		<ul style="list-style-type: none"> <li>- Suite des features essentielles <ul style="list-style-type: none"> <li>- Implémentation Mode de jeu mêlée générale (25h)</li> <li>- Implémentation du tutoriel non-interactif (5h) <ul style="list-style-type: none"> <li>- Implémenter l'interface du tutoriel</li> </ul> </li> <li>- Implémentation de trois groupes d'effets présents (10h) <ul style="list-style-type: none"> <li>- Implémenter les effets visuel et sonore lors de la saisi de mot</li> <li>- Implémenter bruit sonore lorsqu'il ne reste plus beaucoup de temps et lorsqu'on saisi mauvais mot</li> </ul> </li> <li>- Implémentation de l'historique du clavardage(10h) <ul style="list-style-type: none"> <li>- Implémenter l'interaction entre le serveur et le client et obtenir l'historique</li> </ul> </li> </ul> </li> </ul> <p>Serveur</p> <ul style="list-style-type: none"> <li>- Suite des features essentielles <ul style="list-style-type: none"> <li>- Implémentation des personnalité des joueurs variés <ul style="list-style-type: none"> <li>- Implémenter l'envoi de message dans le chat du groupe (25h)</li> </ul> </li> <li>- Implémentation de l'historique de clavardage <ul style="list-style-type: none"> <li>- Implémenter la réponse au requête d'historique (20h)</li> </ul> </li> </ul> </li> </ul>	
Lot 8	12-18 Nov 2020	<p>Client lourd/léger:</p> <ul style="list-style-type: none"> <li>- Début des features souhaitables <ul style="list-style-type: none"> <li>- Implémentation de l'internationalisation (5h) <ul style="list-style-type: none"> <li>- Implémentation d'une interface francais et anglais (default)</li> </ul> </li> <li>- Implémentation de la friendlist (5h) <ul style="list-style-type: none"> <li>- Implémenter l'ajout et le retrait d'amis dans la liste</li> <li>- Implémentation de l'affichage de la liste d'amis</li> </ul> </li> <li>- Implémentation du leaderboard (10h) <ul style="list-style-type: none"> <li>- Implémenter l'interface du leaderboard</li> </ul> </li> </ul> </li> </ul>	<p>120h</p> <p>Léger:35h</p> <p>Lourd:35h</p> <p>Serveur:40h</p> <p>Autre: 10h</p>

		<ul style="list-style-type: none"> <li>- Implémenter les requêtes pour les divers informations au serveur</li> <li>- Implémentation du sprint coopératif (15h) <ul style="list-style-type: none"> <li>- Implémenter la saisi des mots</li> <li>- Implémenter le processus de dessin</li> <li>- Implémenter du système de pointage</li> </ul> </li> </ul> <p>Serveur</p> <ul style="list-style-type: none"> <li>- Début des features souhaitables <ul style="list-style-type: none"> <li>- Implémentation du leaderboard (20h) <ul style="list-style-type: none"> <li>- Implémenter les réponses au requêtes des divers informations sur le leaderboard</li> <li>- Implémentation des mise à jour des informations</li> </ul> </li> <li>- Implémentation du friendlist (20h) <ul style="list-style-type: none"> <li>- Implémenter l'envoi d'un message lorsqu'on ajoute un amis a la liste</li> </ul> </li> </ul> </li> </ul> <p>Tests après chaque feature</p>	
Lot 9	19 - 26 Nov 2020	<p>Client lourd/léger:</p> <ul style="list-style-type: none"> <li>- Suite des features souhaitables <ul style="list-style-type: none"> <li>- Implémentation du thème personnalisé (10h) <ul style="list-style-type: none"> <li>- Implémenter divers thèmes et les appliquer quand nécessaire</li> </ul> </li> <li>- Implémentation du rang mêlée générale (10h) <ul style="list-style-type: none"> <li>- Implémenter les requêtes pour les différents rang pour un utilisateur</li> </ul> </li> <li>- Implémentation de l'invitation de quelqu'un dans l'application (10h)</li> <li>- Implémentation d'autre points souhaitable (5h)</li> </ul> </li> </ul> <p>Serveur</p> <ul style="list-style-type: none"> <li>- Suite des features souhaitables <ul style="list-style-type: none"> <li>- Implémentation du rang mêlée générale (20h) <ul style="list-style-type: none"> <li>- Implémenter la réponse au requête pour le rang d'un utilisateur</li> <li>- Implémenter la mise à jour du rang d'un utilisateur</li> </ul> </li> <li>- Implémentation de l'invitation (20h) <ul style="list-style-type: none"> <li>- Implémenter l'envoi d'un email pour inviter une personne dans l'application</li> </ul> </li> </ul> </li> </ul>	<p>120h</p> <p>Léger:35h</p> <p>Lourd:35h</p> <p>Serveur:40h</p> <p>Autre: 10h</p>

Lot 10	26-02 Déc	<b>- Révision finale:</b> <b>-Contrôle de qualité final</b> <b>- Deuxième livrable : produit final (2 Décembre)</b>	60h Léger:20h Lourd:20h Serveur:15h Autre: 5h
--------	-----------	---------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------

## 5. Équipe de développement

### Gloria Aubierge Sohou

Gloria est une étudiante en troisième année de génie logiciel à Polytechnique Montréal. Elle travaille principalement sur le serveur. Son stage en analyse et traitement de données lui a permis d'acquérir et d'approfondir ses connaissances sur les bases de données SQL, en programmation Python et PHP. Elle a aussi fait un projet personnel de développement d'application mobile en utilisant Java et Android Studio. Cela lui a permis d'avoir une bonne base en développement d'applications avec Android Studio.

### Sophie Dorval

Sophie est une étudiante de quatrième année en génie logiciel. Durant ces stages, Sophie a acquis une expertise en C#/.NET et dans la création d'API REST en travaillant sur le back-end des projets. Elle a également acquis des connaissances en Typescript et Java lors de son cheminement académique. Dans ce projet, elle est responsable du client-léger Android.

### Fabrice Ndui

Fabrice est un étudiant en génie logiciel à Polytechnique. Il aime jouer aux nouvelles technologies et adore régler les problèmes les plus difficiles. C'est un bon administrateur et organisateur qui a toujours un plan en tête. Il se chargera du développement du serveur ainsi que de superviser l'intégration dans les deux clients.

### Kevin Nguyen

Kevin est un étudiant en génie logiciel à Polytechnique. Durant ses stages, Kevin a acquis une expertise sur Python, Typescript et Java en travaillant sur des projets full-stack. Il a travaillé sur des projets open source tel [VScode Device Simulator Express de Microsoft](#). Dans le cadre du projet, il est responsable du client-léger Android.

### Mohamed Barouche

Mohamed est un étudiant de troisième année au baccalauréat en génie logiciel à l'école Polytechnique de Montréal. Durant son dernier stage, Mohamed a acquis une expérience sur Python et l'automatisation de tests web. Il a également acquis, lors de son cheminement scolaire, de l'expérience sur Java et typescript. Il est responsable du client-lourd lors de ce projet.

### Youbo Wang

Youbo est un étudiant de quatrième année en génie logiciel à Polytechnique Montréal. Au cours de son cheminement académique, il a acquis des bonnes connaissances en programmation orienté objet. Ayant déjà travaillé avec Angular et Typescript lors d'un projet précédent, il exercera son expérience en tant que responsable du client-lourd durant ce projet.

## **6. Entente contractuelle proposée**

### **6.1 Tâches du développeur**

L'équipe de projet devra terminer le développement du logiciel selon les échéanciers donnés dans le projet. Conformément à ces échéances, le produit final sera livré au client avant le 2 décembre 2020.

L'équipe de projet fournira au client après la date de livraison, une période cumulative de formation relative au fonctionnement du logiciel si le client le demande.

### **6.2 Livraison**

Le logiciel fonctionnera conformément aux spécifications au plus tard, la date de livraison du 2 décembre 2020.

### **6.3 Compensation**

En contrepartie du service, comme mentionné dans l'appel d'offre, le client paiera le groupe de projet les taux suivants :

- développeur : 100\$/h,
- gestionnaire de projet : 125\$/h.

En considérant que le temps total de développement du projet sera de 1080 heures, selon l'appel d'offre, le nombre d'heures accordé au projet par chaque membre de l'équipe, soit 6 au total, sera de 180 heures. De plus, considérons que chaque membre de l'équipe accordera 1/8 de leur temps sur la gestion de projet, le client s'engage à déboursier la somme de 18 562.50\$/personne soit des frais totaux de 111 375\$.

### **6.4 Modification des spécifications**

Le client peut demander que des modifications raisonnables soient apportées aux spécifications. Si le client demande un tel changement, ce dernier et l'équipe de projet se concertent et le client pourra soit retirer le changement proposé ou demander à l'équipe qu'il fournisse le logiciel avec le changement proposé entraînant ainsi un retard et des frais supplémentaires. Le client accepte et reconnaît que le jugement quant à savoir s'il y aura un retard ou des dépenses supplémentaires sera effectué uniquement par l'équipe de projet.

### **6.5 Aucune modification sauf par écrit**

Aucune modification de cet accord ne sera valable à moins que celle-ci soit écrite et acceptée par les deux parties.