
Équipe 207

**Fais-moi un dessin
Plan de projet**

Version 2.0

Historique des révisions

Date	Version	Description	Auteur
2021-02-08	1.0	Soumission du projet	Julien Witty
2021-02-18	1.1	Modification de l'horaire du projet	Marc-Olivier Riopel
2021-02-19	1.2	Révision du document	Marie-Ève Patron
2021-04-17	2.0	Révision final pour remise du projet	Guillaume Thibault

Table des matières

1. Introduction	4
2. Énoncé des travaux	5
2.1. Solution proposée	5
2.2. Hypothèses et contraintes	5
2.3. Biens livrables du projet	5
2.3.1. 19 février 2021	5
2.3.2. 19 avril 2021	5
3. Gestion et suivi de l'avancement	6
3.1. Gestion des exigences	6
3.2. Contrôle de la qualité	6
3.3. Gestion de risque	7
3.3.1 Risques humains	7
3.3.2 Risques techniques	9
3.4. Gestion de configuration	10
4. Échéancier du projet	12
4.1 Appel d'offres et conception	12
4.2 Développement	12
4.3 Assurance qualité	12
4.4 Horaire	12
5. Équipe de développement	19
6. Entente contractuelle proposée	21
6.1 Titre de l'acte	21
6.2 Désignation des parties	21
6.3 Objet de l'entente	22
6.4 Terme et renouvellement	22
6.5 Prix	22
6.6 Remboursement	22
6.7 Terme de paiement	22
6.8 Les droits de propriété intellectuelle	22
6.9 Acceptation des clauses et des conditions	23

Plan de projet

1. Introduction

L'entreprise PolyApp a fait appel à Olivier Gendreau, responsable du programme de baccalauréat en génie logiciel et maître d'enseignement à Polytechnique Montréal, dans le but de développer une application. Ce dernier est à la recherche de développeurs pour soumissionner sur le projet. Le but du document suivant est de décrire les modalités de fonctionnement du projet proposé. La section 2 décrit plus spécifiquement le travail à faire pour mener à terme le projet. Ceci inclut la solution proposée, les hypothèses et contraintes du projet et finalement une description des produits acheminés au client. La section 3 du document a pour but de spécifier les modalités de gestion du projet. Ceci implique différentes sphères essentielles au processus de développement. Cette section permettra aussi d'évaluer les priorités du projet. La section 4 aura pour but de décrire l'échéancier et les objectifs pour chaque itération du projet afin de bien évaluer l'efficacité du processus. La section 5 a pour but de définir les rôles des membres de l'équipe et présenter leur expertise. Finalement, la section 6 concerne l'entente contractuelle pour respecter l'appel d'offres.

2. Énoncé des travaux

2.1. Solution proposée

En réponse à l'appel d'offres de l'autorité contractante, l'équipe de développement propose une solution respectant les contraintes établies dans les documents de visions et compléments pédagogiques. L'équipe de développement propose également des fonctionnalités additionnelles par l'ajout d'une combinaison d'exigences essentielles et souhaitables. Toutes les exigences essentielles seront développées et feront partie du livrable final alors que seulement la moitié des exigences souhaitables seront implémentées. La décision de garder ou rejeter une exigence souhaitable reviendra à l'équipe de développement.

Pour cela, l'équipe propose de créer le logiciel Poly-Dessin en partant du logiciel développé en projet 2. Grâce à cette base, il sera facile d'utiliser les fonctionnalités déjà implémentées pour dessiner et de créer des jeux. Les jeux sont basés sur le concept de devinette, une personne devra dessiner et les autres deviner. Le mode de jeu classique sera constitué de deux équipes ou les membres d'une équipe ont pour but de se faire deviner leur dessin entre eux pour gagner plus de points que l'équipe adverse. Le mode de jeu sprint, qui pourra se jouer seul ou à plusieurs, aura pour but de deviner un maximum de dessins en un certain laps de temps. De plus plusieurs fonctionnalités comme des canaux de clavardage, un fil d'actualité avec les derniers dessins dessinés et un système de création de paire mot image seront créés.

Cette solution repose sur une architecture client/serveur utilisant les protocoles WebSockets et HTTP pour la communication avec l'intermédiaire d'une connexion internet (Wifi ou Ethernet). Le client sera développé pour être fonctionnel comme application de bureau (Windows, macOS et Linux) ainsi que pour mobile (Samsung Galaxy Tab A 201). Toute information devant être conservée sera entreposée sur une base de données MongoDB qui sera déployée sur MongoDB Atlas pour permettre l'accessibilité à celle-ci en tout temps. MongoDB, une base de données de type NoSQL, offre l'avantage d'avoir de la flexibilité sur les modèles de données entreposées. Le serveur de l'application sera hébergé sur Amazon Web Service pour son avantage économique et sa fiabilité.

Les informations détaillées sur les exigences sont disponibles dans le document de spécifications des requis du système (SRS).

2.2. Hypothèses et contraintes

L'équipe de développement devra travailler sous certaines contraintes et hypothèses. Les développeurs n'étant pas tous familiers avec les technologies utilisées dans le cadre du projet, ils devront prendre en considération une période d'adaptation dans l'échéancier. L'équipe devra également adapter son fonctionnement interne au télétravail puisque les rencontres en personnes à Polytechnique Montréal seront peu fréquentes d'ici la remise du livrable final.

Puisque le client dépend fortement du serveur, l'équipe responsable de développer le serveur devra s'assurer de prendre les devants pour faciliter le travail du reste de l'équipe. De plus, le serveur sera redéployé au fil des mises à jour pour que tout le monde soit en mesure de travailler avec la même version lors du développement.

2.3. Biens livrables du projet

2.3.1. 19 février 2021

- Artéfacts: Plan de projet, SRS, listes d'exigences, architecture logicielle et document protocole de communication
- Prototype de communication (client lourd, client léger et serveur)

2.3.2. 19 avril 2021

- Artéfacts: Mise à jour des artéfacts de la première remise, le plan de tests, les résultats de tests et le code source du produit final

3. Gestion et suivi de l'avancement

3.1. Gestion des exigences

Les exigences du projet, telles que définies dans le document de spécification des requis (SRS), sont une décomposition logique du projet qui permettra aux développeurs de séparer le travail en tâches découplées l'une de l'autre. Toutes les exigences essentielles feront partie du livrable final, alors que certaines exigences souhaitables ne feront pas partie du livrable final.

Le suivi des exigences est réalisé à l'aide de l'outil de gestion de projet Jira. Celui-ci permet de suivre l'avancement de toutes les exigences du SRS et de les attribuer aux différents membres de l'équipe de développement. Un suivi est réalisé chaque mardi midi afin de s'assurer que les exigences à développer au travers des différents sprints ont été réalisées. De plus, cette rencontre permet de mettre à jour le calendrier. Les membres de l'équipe ont la responsabilité d'enregistrer les heures de travail accompagnées d'une description des tâches effectuées. En combinant l'utilisation de Jira avec nos rencontres hebdomadaires, il nous est possible de garder une trace et de gérer facilement des changements aux exigences.

L'équipe devra se conformer à l'échéancier établi dans la section 4 de ce document. Un suivi sera fait chaque semaine avec le client pour discuter de l'avancement du travail. Également, l'équipe se rencontrera quatre fois par semaine à 11h30 (lundi, mardi, mercredi et vendredi) pour coordonner leurs efforts et suivre l'avancement de leurs pairs.

Une modification des exigences devra être acceptée par le client et par toute l'équipe de développement. Tous les artéfacts affectés par ce changement seront modifiés en conséquence.

En cas de changement des exigences, ce qui peut arriver dû à des révisions du client ou dû à des changements de priorités de l'équipe, une révision des échéanciers devra être réalisée pour s'assurer que les changements pourraient bien être réalisés. Pour cela, le SRS devra être révisé et les changements acceptés par le client. Lorsque les changements seront approuvés, le document d'architecture et le protocole de communication seront changés pour incorporer la nouvelle fonctionnalité. De plus, si les changements comportent un retrait d'une fonctionnalité, cette fonctionnalité devra être barrée dans le SRS pour pouvoir avoir un historique des changements et pour ce qui est des autres documents, la fonctionnalité devra être retirée.

3.2. Contrôle de la qualité

L'assurance qualité du client lourd sera faite à l'aide de tests manuels utilisés pour faire des tests d'intégration ainsi que des tests pour s'assurer que les fonctionnalités respectent les requis.

Les tests du serveur seront des tests d'intégration qui s'effectueront avec l'application Postman pour la logique impliquant REST et à partir du client lourd et léger pour la logique impliquant Socket.io.

Pour le client léger, des tests d'intégration seront faits à la main afin de s'assurer du bon fonctionnement du système lorsqu'il interagit avec le serveur et les autres utilisateurs.

En plus des tests unitaires, il est important de faire des tests manuels en boîte noire afin de détecter des comportements non désirés. Un protocole d'actions précises à effectuer sera élaboré et ajouté au plan de test pour être en mesure de bien tester les composantes déjà en place après l'ajout d'une composante. Si un bogue est détecté, le testeur devra ajouter le bogue sur Jira avec un degré de priorité adéquat. Une fois le bogue réglé, le testeur devra ajouter une nouvelle procédure de tests en boîte noire pour vérifier si le bogue se répétera lors de futur ajout.

Pour maintenir un haut standard de qualité de code, chaque "commit" sur la branche "master" du dépôt git devra être révisé par un membre de l'équipe en utilisant le mécanisme de "merge request" de Gitlab. Également, l'équipe de développement s'engage à suivre les normes officielles d'Angular pour le client lourd, les normes officielles de Kotlin pour le client léger et les normes officielles d'Express pour le serveur.

Pour intégrer une fonctionnalité terminée, la personne devra faire une "merge request" pour ajouter sa fonctionnalité au reste du projet. La personne qui regardera la "merge request" devra tester la fonctionnalité de façon manuelle et avec les derniers changements du projet incluse. Si des problèmes sont trouvés, la personne devra rapporter les problèmes à la personne qui a développé la fonctionnalité et celle-ci devra les régler avant que quelqu'un accepte sa fonctionnalité.

Les documents devront être révisés par au moins une autre personne lorsque quelqu'un y apporte une modification. De plus, les documents devront être passés sur Antidote pour s'assurer de la validité de la langue avant de les envoyer aux clients.

3.3. Gestion de risque

La description des risques suit la convention suivante :

- Ampleur : sur une échelle de 1 à 10, 10 étant le risque le plus élevé. Cette analyse est basée sur la probabilité d'occurrence du risque, ainsi que ses impacts.
- Description : une description textuelle du risque ainsi que les problèmes attendus.
- Impact : échelle définissant la portée du risque
 - C – critique (affecte le projet en entier)
 - E – élevé (affecte les fonctionnalités principales du système)
 - M – moyen (devrait être maîtrisable en appliquant une stratégie d'atténuation adéquate)
 - F – faible (l'acceptation du risque est une stratégie envisageable)
- Facteurs : aspects (**métriques**) du système pouvant être compromis.
- Stratégie de gestion : mesures à prendre afin de gérer le risque.

3.3.1 Risques humains

Tableau 1. Risque de conflit d'équipe

1a - Conflit d'équipe				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Des membres dans l'équipe n'ont jamais travaillé ensemble par le passé. Il est probable que certaines personnalités ne soient pas compatibles et engendrent des conflits dans l'équipe. Cela peut avoir une incidence très négative.	C	Délais dans la planification Communication Cohésion d'équipe	Avoir recours aux conseillers HPR (transférer)

Tableau 2. Risque lié au peu d'expérience des utilisateurs

2a - Expérience des développeurs en développement Android				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
3	Aucun de nos développeurs n'a réalisé une application Android par le passé et l'estimation de la complexité de certaines exigences n'est peut-être pas précise.	M	Délais dans la planification Nombre de fonctionnalités développées	Les développeurs Android devront s'assurer d'apprendre le langage avant le début du projet et de vérifier les heures dans la planification afin de s'assurer que les objectifs sont réalisables. D'autres membres de l'équipe seront aussi disponibles pour venir prêter main-forte en cas de retard sur l'échéancier. (Accepter)

Tableau 3. Risque lié à la COVID-19

3a - Covid-19				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	Si un des développeurs contracte le virus, il est possible que celui-ci ne soit pas en mesure de participer au développement du produit et cela pourrait entraîner un retard sur l'échéancier	M	Délais dans la planification	S'assurer de respecter la distanciation sociale et prendre les mesures nécessaires pour réduire les risques de contracter le virus. (Accepter)

Tableau 4. Risque lié à une mauvaise estimation du temps de développement

4a - Mauvaise estimation du temps de développement				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Il est difficile d'estimer le temps que prendra le développement d'une exigence. Si nous avons estimé les exigences à la baisse, plusieurs délais inattendus pourraient en découler.	M	Accumulation de travail	Organisation de quatre rencontres par semaine à 11h30 (lundi, mardi, mercredi et vendredi) pour faire le suivi de l'état du développement. Les bloquants seront annoncés lors de ces rencontres et nous tenterons de trouver des solutions le plus rapidement possible. En cas d'un trop grand retard, nous enlèverons des fonctionnalités. (Mitiger et contourner)

3.3.2 Risques techniques

Tableau 5. Risque lié à une erreur dans le design d'architecture

1b - Erreur dans le design d'architecture				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	Un design d'architecture comportant des erreurs peut entraîner des retards majeurs lors de la production, car des changements peuvent comporter la réécriture complète de certaines parties du logiciel.	E	Délais majeurs dans la planification	Design d'un prototype comprenant les technologies que l'équipe n'a jamais utilisées dans le passé. Ajouter de petites fonctionnalités au logiciel fréquemment pour éviter d'investir trop de temps. (Mitiger)

Tableau 6. Risque lié aux informations sensibles

2b - Informations sensibles sur le dépôt git (credentials)				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
1	Plusieurs informations sensibles (clé d'accès, mot de passe, etc.) seront utilisées pour se connecter à différents systèmes (base de données, serveur, etc.) et pourraient potentiellement se retrouver sur le dépôt git.	M	Sécurité	Utiliser une librairie pour la gestion de secrets comme dotenv. (Transférer)

Tableau 7. Risque lié à des divergences d'interface entre les plateformes utilisées

3b - Interface différente entre le client lourd et léger				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
5	Les deux applications seront développées par des développeurs différents ce qui peut causer une différence entre les interfaces et la vision du projet.	E	Cohésion de l'application entre les plateformes	Des réunions régulières entre les développeurs seront effectuées afin de vérifier que les maquettes sont respectées. Un des développeurs sera également chargé de faire le pont entre les deux interfaces pour s'assurer de la cohésion. (Transférer)

Tableau 8. Risque lié à des performances du serveur

4b - Performance du serveur liée à la capacité de données traitées				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Le serveur est développé et testé pour un nombre limité d'utilisateurs. Un haut nombre d'utilisateurs pourrait ralentir le serveur et causer des problèmes de délais pour les utilisateurs. De plus, le serveur utilisé n'offre pas l'option de faire du multithreading. Une mauvaise conception de l'application pourrait entraîner des traitements de données inutiles qui augmenteraient le risque d'avoir des problèmes de synchronisation.	M	Performance du système Disponibilité du système	L'implémentation de certaines fonctionnalités réalisables du côté client permettra de diminuer la charge du serveur et ainsi accueillir davantage d'utilisateurs. L'utilisation d'un système asynchrone pour permettre au serveur de ne pas bloquer les requêtes peut également contribuer à éviter que tous les utilisateurs soient impactés par des requêtes plus lourdes. (transférer)

Tableau 9. Risque lié à une mauvaise estimation du temps de débogage

5b - mauvaise estimation du temps pour corriger les bogues				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	Les bogues sont difficiles à prévoir, surtout lorsque les développeurs utilisent de nouvelles technologies. Il peut donc y avoir un risque que l'équipe n'évalue pas bien le temps requis pour corriger les bogues qui apparaîtront lors du développement, ce qui aurait un impact négatif sur le produit final.	M	Fiabilité	Une bonne planification du temps alloué pour les phases de tests et de correction des bogues permettra d'éviter cette problématique. Des tests d'intégration continue vont également contribuer à éviter de rencontrer des bogues majeurs du système à la fin du projet. (Mitiger)

3.4. Gestion de configuration

Chaque semaine, une personne sera responsable de monter le calendrier du sprint suivant avec les différentes fonctionnalités à faire et ce calendrier sera révisé lors du "scrum" du mardi. Lors de la révision du calendrier, la distribution des tâches sera faite et l'estimation des heures. Cette façon de faire permettra à chaque personne de connaître ses tâches ainsi que d'avoir une estimation réaliste du nombre d'heures que chaque tâche prendra à être réalisé.

Les différentes versions de code seront nommées de la façon X.Y.Z. Le X représente une nouvelle version drastique, le Y, une nouvelle fonctionnalité et le Z une correction de bogue. Le logiciel partira de 1.0.0 et les nombres augmenteront par incrément.

Les différentes versions des artefacts seront nommées de la façon X.Y. Le X représente une version remise au client et le Y représentent les différentes améliorations de ce document entre deux présentations au client.

Lors de la découverte de bogues dans l'application, ceux-ci sont répertoriés dans Jira. Une personne sera attribuée à chaque bogue et assumera la responsabilité de le régler.

L'outil de gestion de version git sera utilisé (Gitlab). L'utilisation des branches permettra de diviser le code en production et en développement. Les différentes branches seront:

- Une branche production qui comporte le code utilisé en production, soit le code présenté au client lors du processus agile. Ce code devra être étiqueté à la fin de chaque sprint pour pouvoir revenir à d'anciennes versions plus facilement.
- Une branche développement comportant les fonctionnalités complétées durant le sprint en cours
- Des branches pour chaque fonctionnalité en cours de développement.
- Des branches pour la correction de bogues

Le tableau ci-dessous montre un exemple de la manière dont les branches seront divisées pour permettre une gestion efficace des problèmes et des changements tout au long du projet.

Tableau 10. Exemple de branches utilisées pour le projet

Type de branche	Nomenclature
Production	master
Développement	dev
Fonctionnalités	feature-123
Bogues	fix-123

Une branche de fonctionnalité ou de bogue devra être approuvée par un membre de l'équipe avant que le code de celle-ci soit ajouté à la branche de développement. Également, les membres seront dans l'impossibilité de soumettre leur code directement sur la branche de production. Ils devront passer par la branche de développement, puisque celle-ci servira pour les tests manuels avant sa mise en production.

4. Échéancier du projet

Le déroulement du projet se fera sur 12 semaines qui seront séparées en sprint d'une semaine chacune. Les quatre premières semaines seront utilisées pour répondre à l'appel d'offres en réalisant un SRS, un plan de projet, un protocole de communication, un prototype et un document d'architecture. Par la suite, nous allons réaliser le développement de l'application. Sept semaines sont prévues pour le développement et une semaine de plus pour la révision du logiciel et les corrections de dernières minutes.

4.1 Appel d'offres et conception

Les deux premiers sprints du projet serviront à réaliser un SRS afin de répondre à l'appel d'offres du projet. Par la suite, un sprint est prévu pour la réalisation d'un plan de projet d'un document d'architecture et d'un protocole de communication. Finalement, le dernier sprint de cette phase est prévu pour le développement d'un prototype pour démontrer la viabilité du projet.

4.2 Développement

Le développement de l'application commencera le 18 février et finira le 13 Avril. Ce temps sera divisé en sept sprints d'une semaine. Chaque sprint a pour but de réaliser quelques fonctionnalités pouvant être présentées au client pour que celui-ci puisse suivre l'avancement de l'équipe et donner de la rétroaction. Un dernier sprint est prévu entre le 14 avril et le 19 avril afin de régler tout problème restant avec la livraison finale du produit. L'estimation de temps a été réalisée une première fois par une attribution pondérée des heures selon le nombre de points des fonctionnalités. Les heures ont par la suite été réestimées avec des estimations plus réalistes en fonction de leur intégration dans l'application selon le document d'architecture.

4.3 Assurance qualité

L'assurance qualité du projet sera réalisée au même moment que le développement des fonctionnalités de l'application. Le temps alloué pour le développement de test est compris dans l'attribution des heures pour les différentes fonctionnalités dans l'horaire.

4.4 Horaire

Tableau 11. Horaire du projet

Sprint	Date de début	Date de fin	Lot de travail	Temps par tâche estimé	Temps total estimé
Réponse d'appel d'offres	27 jan	17 févr	SRS - Introduction du document	4h	150h
			SRS - Exigences fonctionnelles	18h	
			SRS - Exigences non-fonctionnelles	18h	
			Protocole de communication - Introduction	1h	

			Protocole de communication - Communication client-serveur		3h	
			Protocole de communication - Description des paquets		20h	
			Prototype - Mise en place du serveur sur le nuage (cloud) avec une connection vers la BD		6h	
			Prototype - Implémentation de la communication pour le serveur		18h	
			Prototype - Implémentation du clavardage pour le client lourd		13h	
			Prototype - Implémentation du clavardage pour le client léger		12h	
			Document d'architecture - Introduction		1h	
			Document d'architecture - Vue des cas d'utilisation		10h	
			Document d'architecture - Vue logique		7h	
			Document d'architecture - Vue des processus		12h	
			Document d'architecture - Vue de déploiement		2h	
			Document d'architecture - Taille et performance		1h	
			Gestion du projet		4h	
			17-02-2021		Jalon: Fin de la réponse à l'appel d'offres	
Sprint 4	18 févr	23 févr	Lourd	Clavardage intégré à l'application	4h	79h
				Clavardage dans une fenêtre à part	10h	
				Notifications pour le clavardage	16h	
				Authentification d'un profil existant	14h	
				Création d'un profil pour un nouvel utilisateur	10h	
			Léger	Section de clavardage	12h	
				Création d'un profil pour un nouvel utilisateur	9h	

			Gestion du projet			4h		
	23-02-2021		Jalon: Fin du sprint 4					
Sprint 5	24 févr	2 mars	Lourd	Menu Principal - Liste des parties disponibles	10h	98h		
				Menu principal - Rejoindre une partie	5h			
				Menu principal - Gestion des joueurs virtuels dans les paramètres de la partie	7h			
				Salle d’attente d’une partie	11h			
				Interface pour les informations d’une paire mot-image	10h			
				Zone de dessin pour la création d’une paire mot-image	5h			
			Léger	Prévisualisation de la création d’une image en accéléré	12h			
				Menu Principal - Liste des parties disponibles	10h			
				Menu principal - Rejoindre une partie	5h			
				Menu principal - Gestion des joueurs virtuels dans les paramètres de la partie	7h			
					Salle d’attente d’une partie		12h	
					Gestion du projet		4h	
	02-03-2021		Jalon: Fin du sprint 5					
	Sprint 6	3 mars	9 mars	Lourd	Mode de Jeu Classique - Système d’équipes		15h	118h
Mode de Jeu Classique - Système de points					12h			
Mode de Jeu Classique - Système de tentatives de devinette					7h			
Mode de Jeu Classique - Système de visualisation du dessin en temps réel					10h			
Mode de Jeu Classique - Jouer avec un joueur virtuel					15h			
Léger				Mode de Jeu Classique - Système d’équipes	13h			

				Mode de Jeu Classique - Système de points	12h		
				Mode de Jeu Classique - Système de tentatives de devinette	7h		
				Mode de Jeu Classique - Système de visualisation du dessin en temps réel	10h		
				Mode de Jeu Classique - Jouer avec un joueur virtuel	15h		
			Gestion du projet		4h		
09-03-2021		Jalon: Fin du sprint 6					
Sprint 7	10 mars	16 mars	Lourd	Mode de Jeu Sprint - Système de gestion de temps	10h	87h	
				Mode de Jeu Sprint - Système de présentation des dessins	8h		
				Mode de Jeu Sprint - Système de score	11h		
				Mode de Jeu Sprint Coop - Système de gestion de la collaboration	12h		
			Léger	Mode de Jeu Sprint - Système de gestion de temps	10h		
				Mode de Jeu Sprint - Système de présentation des dessins	10h		
				Mode de Jeu Sprint - Système de score	10h		
				Mode de Jeu Sprint Coop - Système de gestion de la collaboration	12h		
			Gestion du projet		4h		
			16-03-2021		Jalon: Fin du sprint 7		
Sprint 8	17 mars	23 mars	Lourd	Profil utilisateur public	7h	114h	
				Profil utilisateur privé	10h		
				Affichage de l'historique des canaux de discussion	5h		
				Création et suppression de canaux de discussion	7h		

				Gestion des canaux de discussion (Rejoindre, quitter, visualiser la liste des canaux existants)	9h	
				Système de messages personnalisés des joueurs virtuels	12h	
				Indices donnés par les joueurs virtuels	5h	
			Léger	Profil utilisateur public	7h	
				Profil utilisateur privé	10h	
				Affichage de l'historique des canaux de discussion	5h	
				Création et suppression de canaux de discussion	7h	
				Gestion des canaux de discussion (Rejoindre, quitter, visualiser la liste des canaux existants)	9h	
				Système de messages personnalisés des joueurs virtuels	12h	
				Indices donnés par les joueurs virtuels	5h	
			Gestion du projet		4h	
23-03-2021	Jalon: Fin du sprint 8					

Sprint 9	24 mars	30 mars	Lourd	Tutoriel	13	87h
				Effets visuels - Sonores	12	
				Tableau de classement	20	
				Suggestion de mots	10	
			Léger	Tutoriel	13	
				Effets visuels - Sonores	9	
				Suggestion de mots	6	

			Gestion du projet			4h	
	30-03-2021		Jalon: Fin du sprint 9				
Sprint 10	31 mars	6 avril	Lourd	Création d’une partie privée	13h	114h	
				Rejoindre une partie privée	10h		
				Sélectionner un dessin à la fin d’une partie pour le téléverser	15h		
				Entrer les informations nécessaires sur un dessin sélectionné pour le téléverser	17h		
			Léger	Création d’une partie privée	13h		
				Rejoindre une partie privée	10h		
				Sélectionner un dessin à la fin d’une partie pour le téléverser	15h		
				Entrer les informations nécessaires sur un dessin sélectionné pour le téléverser	17h		
			Gestion du projet				4h
			06-04-2021		Jalon: Fin du sprint 10		
Sprint 11	7 avril	13 avril	Lourd	Visionnement d’un fil d’images sur le menu principal	7h	95h	
				Système de sélection de dessins à la fin d’une partie pour mettre un des dessins sur le fil d’image	15h		
				Liste des dessins de la partie à la fin de celle-ci	11h		
				Possibilité de voter si un utilisateur a aimé ou non chaque dessin fait par des joueurs virtuels	16h		
				Système de retrait de dessins dont les votes sont trop négatifs	7h		
			Léger	Liste des dessins de la partie à la fin de celle-ci	12h		
				Possibilité de voter si un utilisateur a aimé ou non chaque dessin fait par des joueurs virtuels	16h		
				Système de retrait de dessins dont les votes sont trop négatifs	7h		

			Gestion du projet	4h	
	13-04-2021		Jalon: Fin du sprint 11		
Sprint 12	14 avril	19 avril	Révision finale pour la livraison	20	60h
			Tests finaux	16	
			Débordement	20	
			Gestion du projet	4h	
	19-04-2021		Jalon: Fin du sprint 12		
Date de livraison				19 Avril 2021	
Total des heures				1008h	1008h

5. Équipe de développement

Julien Witty - Équipe Mobile

Julien Witty est un membre de l'équipe de développement mobile. Ce développeur a participé à l'élaboration de plusieurs projets d'envergure dans le cadre de son cheminement académique et professionnel. Ce développeur a de l'expérience sur le marché du travail en assurance qualité et en développement logiciel C# sur des logiciels de vision en temps réel à application industrielle. Ce candidat sera d'une aide indispensable pour l'élaboration d'une application robuste et fiable.

Guillaume Thibault - Équipe Mobile et Équipe Client Lourd

Guillaume est un membre de l'équipe de développement du client lourd. Ce développeur a une expérience pour ce genre de développement, car il a réalisé un projet semblable lors d'un projet scolaire. Ce projet lui a permis d'acquérir de bonnes connaissances en Angular, Typescript, HTML et CSS, ce qui fait de lui un candidat idéal pour ce poste. De plus, il a de bonnes connaissances en conception d'API de type RESTful ayant fait un stage dans ce domaine. L'API qu'il a réalisée était en Python, ses connaissances pourront aider l'équipe de serveur lors de la conception de celui-ci.

Marc-Alain Tétreault - Équipe Client Lourd

Marc-Alain est un membre de l'équipe de développement du client lourd. Il a acquis de l'expérience en développement web (frontend et backend) et DevOps par l'intermédiaire de stages, projets personnels, projets académiques et autres implications. Sa bonne connaissance du cadre Angular l'aidera à être productif dès le début du projet. Il pourra également assister ses coéquipiers pour le serveur et pour le déploiement continu de celui-ci si nécessaire.

Marie-Ève Patron - Équipe Mobile

Marie-Ève est une membre de l'équipe de développement du client léger. Elle a participé au développement d'un lecteur web utilisant l'API canvas en JavaScript. Elle possède également de l'expérience dans les technologies backend permettant le traitement en parallèle de plusieurs vidéos en direct et leur transmission sur le web à divers clients. Marie-Ève a aussi de l'expérience en gestion et détient une formation en administration. Au cours de l'un de ces stages, elle a eu la chance de participer à une formation intensive sur la philosophie agile et sur le processus Scrum. Marie-Ève pourra apporter ces connaissances pour soutenir l'équipe dans son organisation, ainsi que pour le développement des fonctionnalités du client léger.

Marc-Olivier Riopel - Équipe Serveur

Marc-Olivier est un membre de l'équipe de développement du serveur. Il a acquis de l'expérience en développement d'infrastructures infonuagiques lors de projets personnels et dans son cheminement professionnel. Ce développeur a également de l'expérience avec Node.js, le framework Express.js ainsi qu'avec les REST APIs, ce qui sera utile pour le développement du serveur. Ses connaissances en Typescript et d'Angular pourront être mises à profit pour contribuer à l'équipe de développement du client lourd au besoin.

Samuel Ouvrard - Équipe Serveur

Samuel est un membre de l'équipe de développement du serveur. Il a déjà participé au développement d'une application web de dessin matriciel lors d'un projet scolaire ce qui lui donne un net avantage dans la réalisation d'un jeu de dessin en réseau. Il a aussi acquis de l'expérience en tant que développeur frontend au cours de stages, projets multiples. Son souci du design et des principes ergonomiques d'une interface lui a permis d'obtenir un poste en tant que chargé de laboratoire pour un cours d'analyse et de conception d'interfaces à la Polytechnique Montréal. Il pourra donc apporter cette expertise à l'équipe pour aider ses collègues qui travaillent sur le développement du client lourd et du client léger.

6. Entente contractuelle proposée

6.1 Titre de l'acte

Entente contractuelle avec Polytechnique Montréal

6.2 Désignation des parties

ENTRE

Polytechnique Montréal (Contractant), PME québécoise PolyApps (Client) et Équipe 207 (Soumissionnaire)

Client

Entreprise québécoise PolyApps

Contractant

Polytechnique Montréal représentée par Olivier Gendreau

Département de génie informatique et génie logiciel

2900, boulevard Édouard-Montpetit

Campus de l'Université de Montréal

2500, chemin de la Polytechnique

Montréal (Québec)

H3T 1J4

Coordonnées :

Téléphone: (514) 340-4711 poste 2102

Local: M-4017

Courriel: olivier.gendreau@polymtl.ca

Fonction : Responsable du cours LOG3900

Soumissionnaire

Équipe 207 représentée par le consortium d'étudiants composé de :

- Ouvrard, Samuel
- Patron, Marie-Ève
- Riopel, Marc-Olivier
- Tétreault, Marc-Alain
- Witty, Julien

6.3 Objet de l'entente

L'entreprise PolyApps accorde les droits du logiciel *Fais-moi un dessin* développé par l'équipe 207, selon les requis stipulés dans le document vision et tel que présenté dans le document SRS, dans le cadre du cours de LOG3900 à l'autorité contractante Polytechnique Montréal. L'entreprise PolyApps s'engage à produire une version prototype du client lourd et du client léger pour le 19 février 2021. Le produit final sera livré au contractant le 19 avril 2021 sans possibilité de report de date.

6.4 Terme et renouvellement

L'entente propose un contrat à terme. Le distributeur ne s'engage pas à offrir une assistance technique après la livraison finale de l'application. L'entente devra être renégociée après la livraison de la première version du produit.

6.5 Prix

Basé sur les coûts opérationnels du personnel développant le logiciel :

- Développeur : 100\$/h
- Gestionnaire de projet: 125\$/h

La durée de travail estimée des développeurs du projet est évaluée à 1008h.

Le temps de développement est estimé à 960h. Le temps de gestion pendant la session est estimé à 48h.

Ainsi, le coût de production associé aux ressources humaines est évalué à 101 760 \$.

$$100\$/h \times 960h + 120\$/h \times 48h = 101\,760\$$$

Les heures supplémentaires des développeurs ne seront pas payées.

6.6 Remboursement

Le contractant ne s'engage pas à rembourser le client en cas d'insatisfaction.

6.7 Terme de paiement

Le distributeur doit payer le logiciel avant la remise finale du prototype.

6.8 Les droits de propriété intellectuelle

Le développement du logiciel respectera les lois de la propriété intellectuelle du Canada.

L'entrepreneur s'engage toutefois à céder une licence non exclusive à Polytechnique Montréal pour des fins d'enseignement et de formation au Département de génie informatique et génie logiciel de Polytechnique Montréal.

L'offre peut être rejetée si le soumissionnaire ou l'un des employés est reconnu coupable en vertu de l'article 8 du document Règlements des études du baccalauréat en ingénierie pour l'année 2021-2022.

6.9 Acceptation des clauses et des conditions

Le soumissionnaire doit signer sa proposition. Cette signature implique l'acceptation des clauses de l'appel d'offres et l'ensemble des conditions présentes dans cette DDP. Aucune modification ni aucun ajout aux conditions incluses dans la proposition du soumissionnaire ne feront partie du contrat. La proposition du soumissionnaire, déposée par un consortium contractuel, doit être signée par tous les membres du consortium.