



**Polygram
Plan de projet**

Version 1.0

Historique des révisions

Date	Version	Description	Auteur
2022-01-27	1.0	Rédaction initiale	Chaima & Noursen

Table des matières

1. Introduction	4
2. Énoncé des travaux	4
2.1. Solution proposée	4
2.2. Hypothèses et contraintes	4
2.2.1. Ressources humaines	4
2.2.1. Équipement	5
2.2.1. Échéancier	5
2.3. Biens livrables du projet	5
3. Gestion et suivi de l'avancement	5
3.1. Gestion des exigences	5
3.2. Contrôle de la qualité	6
3.3. Gestion de risque	7
3.4. Gestion de configuration	8
4. Échéancier du projet	9
5. Équipe de développement	16
6. Entente contractuelle proposée	17

Plan de projet

1. Introduction

Ce document présente la planification du projet Polygram, une application de dessin collaboratif avec un système de clavardage. La section 2 décrit la solution que nous proposons, énumère les hypothèses sur lesquelles repose cette planification et les contraintes applicables au projet en matière de ressources humaines, d'équipements et d'échéancier. Cette section comporte également une description des artefacts créés durant le projet avec les dates prévues de livraison. La section 3 aborde les volets de gestion des exigences, des risques et de configuration ainsi que le contrôle de la qualité. Par la suite, un échéancier du projet est développé dans la section 4 suivie par une présentation des membres de l'équipe de développement dans la section 5. Finalement, une entente contractuelle respectant l'appel d'offres est proposée dans la section 6.

2. Énoncé des travaux

2.1. Solution proposée

La solution que notre équipe propose est le logiciel Polygram qui est disponible sous deux versions : une version de bureau compatible avec les ordinateurs ayant un système d'exploitation Windows 10 (client lourd) et une version mobile compatible avec une tablette Android (client léger).

Avec Polygram, les utilisateurs pourront démarrer des séances de dessins individuellement ou bien en collaboration grâce au mode de dessin collaboratif. Ils pourront également communiquer ensemble en envoyant des messages via le mode de clavardage. Les utilisateurs pourront également personnaliser les fenêtres de clavardage. De plus, nous donnons la possibilité aux utilisateurs de partager leurs dessins en mode « story » ce qui confère à notre plateforme une dimension sociale dans le but de rapprocher les utilisateurs. Un système de récompense est également intégré à la solution afin d'encourager les utilisateurs à dessiner et à interagir avec les autres par le biais des mentions « j'aime ».

La version de bureau inclura l'option de partager des dessins vers les plateformes infonuagiques populaires telles que Dropbox et Google Drive afin que les utilisateurs puissent sauvegarder leurs travaux dans les supports de leurs choix. Quant à la version mobile, elle comportera la possibilité d'ajouter les informations de géolocalisation sur les dessins afin que l'utilisateur puisse partager sa localisation avec les autres.

Notre solution est basée sur une architecture client-serveur qui permet l'envoi et la réception des données entre les clients (lourd et léger) et le serveur. L'utilisation de Socket.IO va nous permettre d'établir ce genre de communication entre les différentes parties de notre application.

Dans sa version bureau, la solution Polygram sera développée à partir du logiciel PolyDessin et héritera de certaines de ses fonctionnalités principalement les outils de dessins tels que : le sceau de peinture, le crayon, l'efface, etc.

Pour sa version mobile, la solution Polygram sera développée à partir de zéro.

2.2. Hypothèses et contraintes

2.2.1. Ressources humaines

L'équipe de développement est composée de cinq étudiants, chacun devant fournir un minimum de 12 heures de travail par semaine sur le projet et un total de 900 heures-personnes est attendu pour la remise du produit final. Afin de pouvoir démarrer activement le développement de Polygram, l'équipe devra prendre un temps d'adaptation et d'apprentissage afin d'approprier les nouvelles technologies qui vont être utilisées.

2.2.1. Équipement

Pour le développement du client léger, il sera nécessaire d'utiliser un émulateur pour la tablette qui respecte les spécifications techniques de la Galaxy Tab A 2019 afin de s'assurer que la version mobile répond bien aux exigences établies. Pour le client lourd, tout notre développement et nos tests seront effectués sur des ordinateurs réguliers avec un environnement Windows. Nous allons également utiliser un serveur distant.

2.2.1. Échéancier

Le 11 février 2022, nous allons effectuer la livraison d'une réponse à l'appel d'offres et un prototype pour la communication serveur-client lourd et serveur-client léger. Quant au produit final, il sera remis le 11 avril 2022. Pour s'assurer du bon déroulement du projet, l'équipe de développeurs maintiendra des suivis hebdomadaires avec l'équipe académique. De plus, des rencontres entre les membres de l'équipe de développeurs seront planifiées afin de gérer la progression dans les différentes phases de développement.

Une description plus détaillée des biens livrables est présente dans la section 2.3 ci-dessous.

2.3. Biens livrables du projet

- Le 11 février 2022 : premier livrable – réponse à l'appel d'offres
 - Artefact :
 - Document d'architecture logicielle
 - Plan de projet
 - Protocole de communication
 - Liste d'exigences
 - Spécification des requis du système (SRS)
 - Code :
 - Prototype de communication client lourd-serveur
 - Prototype de communication client léger-serveur
- Le 11 avril 2022 : dernier livrable – Produit final
 - Artefact :
 - Mise à jour des artefacts remis dans la phase précédente
 - Plan de tests logiciels
 - Résultat des tests logiciels
 - Code :
 - Code source et exécutable du produit (client lourd, client léger, serveur)

3. Gestion et suivi de l'avancement

3.1. Gestion des exigences

En équipe, nous avons sélectionné les exigences en nous basant sur la liste d'exigences fournie par l'équipe académique et en effectuant des séances de remue-méninges afin de compléter les exigences pour le client lourd et le client léger. À la lumière de nos discussions, nous avons rempli une liste qui a été évaluée et approuvée par les chargés de laboratoire afin de se conformer aux critères spécifiés dans l'appel d'offres. Cela nous a permis de détailler ces exigences dans le document de spécification des requis (SRS).

Chaque exigence dans la liste approuvée sera transcrite sur l'outil de gestion Jira sous la forme d'un ticket et nous allons nous assurer d'inclure des sous-tâches afin de détailler la quantité de travail et gérer sa complexité. Jira nous permettra de faire le suivi pour chacune des tâches en équipe en y incluant le nombre d'heures de travail estimé ainsi que l'état d'avancement et le niveau de priorité.

Cette méthode facilitera notre travail comme développeurs et nous permettra de quantifier le travail en termes de tâche à effectuer, complexité et priorité. Cela nous aidera à suivre l'évolution des exigences et à décider des ajustements à apporter. Lors de nos rencontres d'équipe et à la lumière de nos avancements, nous allons décider de la priorisation et des changements qui pourraient impacter certaines exigences ou certains choix technologiques. Par conséquent, ces changements seront communiqués et révisés par l'équipe académique lors de nos rencontres hebdomadaires. Si les changements sont validés, nous apporterons les modifications nécessaires à nos artefacts et à nos tickets sur Jira.

3.2. Contrôle de la qualité

Afin de nous assurer d'avoir des livrables (artefact et code) qui respectent les attentes de nos responsables de projet (équipe académique) plusieurs mesures vont être prises par notre équipe. Nous mettrons en place des mécanismes afin de garantir un contrôle continu sur nos livrables.

Tout d'abord, afin de respecter nos échéanciers, nous avons décidé en équipe du contenu de nos artefacts et avons décidé de nous partager en sous-équipe pour faire avancer tous les artefacts en même temps. Cette approche garantit une vue d'ensemble du projet pour chaque membre de notre équipe. Une fois la phase de rédaction achevée, une révision générale de chaque artefact aura lieu, et ce avant la livraison.

Pour la qualité de code, nous avons décidé de suivre un guide de codage particulier afin d'établir une norme au sein de notre équipe qui garantit une solution finale homogène. Nous nous engageons à respecter ces règles de programmation que nous établirons en équipe dans le but d'avoir un code uniforme.

Des tests unitaires seront écrits pour toutes nos fonctionnalités du côté de nos deux clients (lourd et léger). Nous continuerons de respecter le critère de couvertures de 90%, établi pour le projet PolyDessin. De plus, nous essaierons d'effectuer des tests de performance relatifs au serveur afin de vérifier qu'il retourne bien les données requises et des tests système en exécutant plusieurs scénarios complets qui constituent les cas d'utilisation de notre logiciel. Les différents types de tests seront détaillés dans le document de plan de tests qui fait partie du livrable final.

Pour toutes nos requêtes, nous ferons des tests d'appel avec soit Swagger ou bien Postman (le choix de la plateforme reste au développeur) pour nous assurer d'avoir la réponse voulue.

De plus, pour le code du client lourd devra passer l'analyse du TSLint et doit être sans avertissement du côté client léger.

Notre équipe va adopter l'approche CI/CD (intégration continue/déploiement continu) à l'aide de la plateforme Gitlab. Nous utiliserons les pipelines ainsi que le mécanisme de « merge request ». Avant de propager un changement vers la branche principale du projet, il faudrait que le code passe en revue et soit révisé par au moins deux personnes de l'équipe afin que la fusion soit validée. Mais encore, un code qui ne passe pas ses tests ni son Lint se verra refuser sa requête de fusion. De plus, nous pensons établir une sorte de *check-list* pour les fonctionnalités qui sont opérationnelles et cocher chaque élément de cette liste après un *merge* pour nous assurer que nous n'avons introduit aucun bogue lors de notre intégration.

Si un bogue est localisé, une demande de correction sera créée dans Jira sous la catégorie « Bogue » et nous déciderons en équipe de sa priorité et des actions correctives nécessaires.

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.

1 - Travailler avec de nouvelles technologies				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	Les membres de l'équipe ont peu d'expériences avec les technologies utilisées dans ce projet. Notamment, l'utilisation d'Android Studio et Kotlin pour le côté du client léger ainsi que la communication en temps réel client-serveur via des sockets.	E	Nombre d'heures d'apprentissage. Nombre d'heures de développement. Nombre de bogues.	Se décider entre les membres pour la tâche de chacun dans le projet (selon son niveau de connaissance) et se familiariser dès le début avec les nouvelles technologies.

2 - Manque de communication				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	Un manque de communication interne entre les membres de l'équipe ou externe entre l'équipe et le client.	C	Nombre de bogues. Nombre d'heures de correction de problèmes. Qualité de la remise finale.	Les membres doivent être transparents et demander de l'aide au besoin. Les membres doivent être présents pendant toutes les rencontres d'équipes ou aviser en avance si cas contraire. Chaque membre doit être à l'écoute de ses coéquipiers. Les membres doivent poser leurs questions de manière efficace à l'équipe académique et bien comprendre leurs besoins.

--	--	--	--	--

3 - Cohérence entre toutes les parties du projet				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
4	Les parties de chaque artefact ainsi que l'implémentation de chaque fonctionnalité du côté client lourd et client léger seront implémentées par des membres différents.	M	Qualité du projet final.	Les membres de l'équipe doivent se parler et se tenir au courant fréquemment et une maquette peut être faite afin de garantir la cohérence entre toutes les parties. Avant la remise, chaque document doit être révisé par tous les membres.

4 - Le temps prévu n'est pas suffisant				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
5	Sous-estimer le temps qu'une tâche peut prendre à être implémentée ou le temps nécessaire pour fixer un bogue.	M	Nombre d'heures de développement. Nombre de fonctionnalités développées. Date prévue de livraison.	Les membres de l'équipe doivent utiliser Jira fréquemment et de manière efficace. Chaque membre de l'équipe est responsable d'aviser ses coéquipiers si le temps prévu n'était pas suffisant ou s'il bloque et a besoin d'aide.

5 - Livrer toutes les exigences prévues				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	Ne pas être en mesure de livrer toutes les exigences essentielles et souhaitables du projet.	E	Nombre de fonctionnalités développées. Date prévue de livraison.	Diviser les exigences en sous-tâches de façon claire et précise et les partager efficacement entre les développeurs. Bien communiquer. Se rencontrer fréquemment et employer la méthode de programmation par les pairs si la tâche semble difficile.

3.4. Gestion de configuration

Lorsqu'un problème est détecté, la personne qui a détecté ce bogue communique de façon claire et précise avec les autres membres en décrivant le problème et son contexte. À ce moment, l'équipe crée un ticket sur

Jira pour fixer le problème et précise le temps estimé pour cette tâche ainsi que la ou les personnes assignées. Ensuite, une branche Git sera créée et nommée adéquatement pour régler ce bogue. Finalement, quand le bogue est résolu, on fusionne cette dernière branche avec la branche de développement.

On utilisera les outils de gestion Gitlab et Github pour contrôler les versions.

Les artefacts du projet seront nommés de la façon suivante : [titre_du_document][numéro d'équipe][numéro de version] exemple : Plan_de_projet_203_1.2 . Toute modification entraînera une incrémentation dans le numéro de versions soit une incrémentation de 1 si on a modifié la majorité des parties du document et de 0.1 si la modification est mineure.

4. Échéancier du projet

Dans le cadre du projet, nous allons respecter la base de 45h/crédit/personne, étant donné que nous sommes une équipe de 5 et que le cours LOG3900 est un cours à 4 crédits, le nombre d'heures total est de $45h \times 4 \text{ crédits} \times 5 \text{ personnes} = 900 \text{ heures-personnes}$. Par conséquent, l'échéancier que nous proposons sera séparé en lot et en jalon. Les lots de travail correspondent aux lignes du tableau et les jalons marquent la fin de chaque phase signifiant que les objectifs ont été atteints.

L'échéancier suivant comporte deux volets essentiels: la réponse à l'appel d'offres et la remise du produit final.

La remise du produit final comporte quant à elles 3 sprints d'environ 11 jours ouvrables chacun.

	Lot de travail	Effort estimé (en heures- personnes)	Date de début	Date de fin
Réponse à l'appel d'offres (10 janvier au 11 février)				
	Première rédaction du document SRS	8	18-01-2022	23-01-2022
	Révision en équipe du SRS	2	23-01-2022	24-01-2022
	Proposition et validation d'exigences fonctionnelles par l'équipe	4	24-01-2022	26-01-2022
	Modifications finales du SRS à la suites des révisions et changements des exigences fonctionnelles	2	27-01-2022	27-01-2022

	Révision individuelle et en équipe du SRS suivi par la remise finale	1	29-01-2022	29-01-2022
	Première rédaction du plan de projet	6	27-01-2022	31-01-2022
	Révision individuelle et en équipe du plan de projet et mise en place de l'échéancier	1	31-01-2022	01-02-2022
	Modifications et ajout de l'échéancier au plan de projet à la suite des révisions	4	02-02-2022	04-02-2022
	Première rédaction du document d'architecture logicielle	10	27-01-2022	01-02-2022
	Révision individuelle et en équipe du document d'architecture logicielle	1	01-02-2022	02-02-2022
	Ajustements finaux du document d'architecture logicielle à la suite des révisions	1	04-02-2022	04-02-2022
	Première rédaction du protocole de communication	10	01-02-2022	05-02-2022
	Révision en équipe et ajustements finaux	2	06-02-2022	06-02-2022
	Prototype serveur-client léger	10	31-01-2022	05-02-2022

	Prototype serveur-client lourd	10	31-01-2022	05-02-2022
	Intégration et première version du prototype	4	04-02-2022	04-02-2022
	Test en équipe du prototype	2	07-02-2022	07-02-2022
	Ajustement du UI et production de la version finale du prototype	2	08-02-2022	08-02-2022
<i>Totales heures-personnes pour la préparation de la réponse à l'appel d'offres</i>		80 heures-personnes		
Remise de la réponse à l'appel d'offres				11-04-2022
Développement produit final (11 février au 11 avril)				
Sprint 1 (11 février au 25 février)				
	<u>Client lourd</u>			
	Préparation de l'environnement de travail (nettoyage ancienne solution)	3	11-02-2022	11-02-2022
	Clavardage en termes d'intégration (mode fenêtré)	12	11-02-2022	13-02-2022
	Clavardage en termes de canaux de discussions	15	14-02-2022	15-02-2022
	Paramétrage du compte utilisateur et statistiques	28	16-02-2022	19-02-2022
	Gestion des avatars	12	20-02-2022	21-02-2022

	Album de dessins	49	21-02-2022	25-02-2022
	Fonctionnalité profil avancé-accessibilité	12	11-02-2022	12-02-2022
	<u>Client léger</u>			
	Outils de dessins de base (formes, couleurs, crayon)	10	11-02-2022	17-02-2022
	Clavardage en termes d'intégration (mode fenêtré)	30	12-02-2022	17-02-2022
	Clavardage en termes de canaux de discussions	18	15-02-2022	17-02-2022
	Paramétrage du compte utilisateur et statistiques	18	17-02-2022	19-02-2022
	Album de dessins	24	19-02-2022	22-02-2022
	Utilisation appareil photo pour changer l'avatar	12	22-02-2022	25-02-2022
<u>Total (heures-personnes) pour le sprint 1</u>		243		
Sprint 2 (7 mars au 28 mars)				
	<u>Client lourd</u>			
	Effets visuels et sonores	18	07-03-2022	10-03-2022
	Édition de dessins (sélection, translation, redimensionnement, suppression)	91	10-03-2022	28-03-2022

	Interaction (mention “j’aime” sur les dessins)	8	12-03-2022	16-03-2022
	Mode artistique (ajout étampe avancée)	8	12-03-2022	16-03-2022
	Profil avancé (suite: paramétrage et sélection d’effets visuels et sonores)	18	07-03-2022	10-03-2022
	<u>Client léger</u>			
	Édition de dessins(sélection, translation, redimensionnement, suppression)	109	15-03-2022	28-03-2022
	Édition collaborative	36	07-03-2022	13-03-2022
	Effets visuels et sonores	18	07-03-2022	10-03-2022
	Fonctionnalité profil avancé- accessibilité	8	10-03-2022	12-03-2022
	Profil avancé (suite: paramétrage et sélection d’effets visuels et sonores)	12	13-03-2022	15-03-2022
	Éditeur de dessin- ajout d’un dessin en <i>story</i>	8	23-03-2022	28-03-2022
	Album de dessins- filtrage de la liste de dessins en utilisant un champ de texte	8	23-03-2022	28-03-2022
<u>Total (heures-personnes) pour le sprint</u> <u>2</u>		342		

Sprint 3 (28 mars au 11 avril)				
	<u>Client lourd</u>			
	Édition collaborative	60	05-04-2022	11-04-2022
	Système de badge en fonction des mentions “j’aime”	8	31-03-2022	05-04-2022
	Profil avancé - système de favoris	8	31-03-2022	05-04-2022
	Profil avancé - top x des dessins aimés	8	31-03-2022	05-04-2022
	Récupération du mot de passe oublié	8	31-03-2022	05-04-2022
	Recherche avancée dans les albums	8	31-03-2022	05-04-2022
	Album de dessins- <i>story</i> du jour	18	31-03-2022	06-04-2022
	Éditeur de dessins- <i>story</i> téléversement	6	28-03-2022	01-04-2022
	Clavardage-intégration en mode fenêtré ou intégré (souhaitable)	6	28-03-2022	01-04-2022
	Clavardage - personnalisation avancée (souhaitable)	6	28-03-2022	01-04-2022
	Production du plan de tests logiciels <i>(toute la durée du sprint car sujet à des</i>	6	28-03-2022	11-04-2022

	<i>changements</i>			
	Révision finale	4	11-04-2022	11-04-2022
	<u>Client léger</u>			
	Interaction (mention j'aime sur les dessins)	6	28-03-2022	29-03-2022
	Édition de photo prise par la caméra de l'appareil	18	30-03-2022	02-04-2022
	profil avancé - top x des dessins aimés	12	06-04-2022	10-04-2022
	Géolocalisation et <i>timestamp</i>	12	06-04-2022	10-04-2022
	Album de dessins - ajout de <i>story</i> du jour	18	30-03-2022	02-04-2022
	Production du plan de test (<i>toute la durée du sprint car sujet à des changements</i>)	8	28-03-2022	11-04-2022
	Révision finale	4	11-04-2022	11-04-2022
	Préparation de la présentation du produit final	11	09-04-2022	11-04-2022
Total (heures-personnes) pour la phase de développement et livraison finale (Sprint 3)		235		
Remise du produit final (11 avril)				
Total pour tout le projet		900	10-01-2022	11-04-2022

5. Équipe de développement

Chaïma Soussi

Est une étudiante de troisième année en génie logiciel ayant réalisé son premier stage en développement web chez Prodco Analytics. Cette expérience et les projets réalisés pendant son cursus scolaire lui ont permis d'acquérir de l'expérience en développement Front-end. Elle maîtrise également plusieurs langages de programmation, notamment C/C++, JAVA, Python. Motivée et prête à apprendre de nouvelles technologies, son principal rôle sera de développer le client-leger et de travailler du côté serveur afin d'améliorer ses compétences en Back-end. Finalement, Chaïma a de bonnes aptitudes de travail en équipe et adore collaborer pour livrer le meilleur produit.

Mikael Gilbert

Est un étudiant de troisième année en génie logiciel ayant complété un premier stage en entreprise chez Druide informatique (été 2020) et un second stage en recherche et développement au CHU Ste-Justine (été 2021). Il a pu développer ses compétences en programmation orientée objet, spécifiquement en C++, et a eu l'opportunité d'implémenter diverses fonctionnalités et patrons de conception. Il a également pu acquérir de l'expérience en développement macOS en Objective-C et a de l'intérêt pour la conception UI/UX tant du côté mobile que *desktop*. Il porte une attention particulière (pour ne pas dire, une obsession) pour le code écrit de façon simple et élégante. Son principal rôle sera de développer l'API entre les clients, le serveur et la base de données en plus de contribuer activement au développement du client lourd en TypeScript (Angular).

Léon Le brun

Est un étudiant de troisième année n'ayant pas encore programmé dans un cadre professionnel, mais il possède néanmoins plusieurs connaissances dans divers langages tels que C#, C++, JAVA, Python et Typescript. Plus spécifiquement, son parcours scolaire lui a permis de se familiariser avec le développement web ainsi que la conception et développement de jeux vidéo. En ce qui concerne le projet courant, Léon sera davantage impliqué dans l'élaboration du client lourd en ayant toujours un œil sur le développement du serveur afin de pouvoir se familiariser avec le back-end du projet et de venir en aide au besoin.

Noursen Houdhek

Est une étudiante de troisième année en génie logiciel ayant effectué son stage obligatoire à la session d'été 2021 chez Desjardins où elle occupe actuellement le poste de développeuse à temps partiel. Elle a pu acquérir de l'expérience en Dev Ops, en assurance qualité et en développement backend avec Java Spring et a déjà travaillé avec Kotlin sur quelques API pour l'implémentation des règles d'affaires du domaine financier. Noursen a également suivi des formations en agilité et prépare sa certification PSPO 1 et espère pouvoir aider son équipe avec le bon déroulement du projet dans un cadrage agile. Sa principale responsabilité sera de développer le client-léger et apporter du soutien du côté serveur. Noursen n'a pas beaucoup d'expérience avec le développement mobile mais est prête à relever le défi et apprendre de nouvelles technologies.

Miora Rakoto

Est une étudiante de troisième année en génie logiciel ayant fait deux stages à la Chaire de recherche en réadaptation pédiatrique. Elle a mis en pratique ses connaissances en programmation orientée objet, précisément son C++ pour la création d'une application permettant le contrôle d'un appareil qui vient d'être introduit en réadaptation pédiatrique. Actuellement, elle poursuit ses travaux de recherche à temps partiel. Sa passion pour la programmation est née au niveau collégial avec le langage Java. Dans le cadre de Polygram, Miora sera dans l'équipe client-léger et apportera aussi son aide pour le développement du serveur. Même si elle n'a pas une grande connaissance en développement mobile, Miora est très motivée à élargir son champ de connaissance.

6. Entente contractuelle proposée

L'entente contractuelle proposée par notre équipe est le contrat livraison clé en main (prix ferme) puisqu'on doit livrer un produit final et que le paiement sera émis à la livraison et à l'acceptation par le client (équipe académique). Aussi, ce type de projet requiert une connaissance exacte de la demande et des spécifications détaillées. Finalement, le projet nécessite une négociation pour établir un changement de spécifications et de la liste d'exigences.

Notre équipe s'engage à suivre les échéanciers et à livrer un produit final au plus tard le 11 avril ainsi que de respecter la charge de travail de chaque fonctionnalité.

Le projet va répondre à toutes les exigences essentielles retenues et détaillées dans le SRS et va répondre à au moins 50% des exigences souhaitables mentionnées dans le même document.

Le nombre d'heures total estimé pour ce projet est de 900 heures dont 80 heures pour la production des artefacts et 820 heures pour le développement de la solution.

Notre équipe est composée de 5 étudiants en génie logiciel qui vont prendre à la fois les rôles de développeur et de gestionnaire. le taux horaire par gestionnaire pour la conception de nos artefacts est de 145\$/h et le taux horaire par programmeur pour le développement est de 110\$/h.

Le prix total estimé pour ce projet est de : 101 800\$.