

---

**<Équipe 9>**

---

**<WALL-E2>**  
**Plan de projet**

**Version 4.0**

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

## Historique des révisions

Date	Version	Description	Auteur
2017-01-27	1.0	Rédaction version initiale	Essaid Djoudi Morgane Renard
2017-01-31	2.0	Correction et complétion des parties : 2 , 5, 6	Essaid Djoudi
2017-01-31	2.1	Correction et complétion des parties : 1 , 3, 4	Morgane Renard
2017-02-08	3.0	Correction finale et mise en page	Essaid Djoudi Morgane Renard
2017-02-08	3.1	Version pour le livrable final	Essaid Djoudi Morgane Renard
2017-04-11	4.0	Version finale corrigée suite aux rétroactions d'olivier	Essaid djoudi

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

# Table des matières

1.	Introduction	4
2.	Énoncé des travaux	4
2.1	Solution proposée	4
2.2	Hypothèses et contraintes	5
2.3	Biens livrables du projet	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
3.4	Gestion de configuration	9
4.	Échéancier du projet	9
4.1	Livrable 1 : Réponse à l'appel d'offre et remise des prototypes (Date de tombée : 10 Février 2017)	9
4.1.1	Documents du projet et appel d'offre	9
4.1.2	Liste des exigences	9
4.1.3	SRS (Spécification des requis du système)	9
4.1.4	Plan de projet	10
4.1.5	Document d'architecture logicielle	10
4.1.6	Prototypes de communication	10
4.1.7	Protocole de communication	10
4.2	Livrable 2 : Remise du produit final (Date de tombée : 11 Avril 2017)	10
4.2.1	Serveur	10
4.2.2	Site Web	11
4.2.3	Client lourd	11
4.2.4	Client léger	13
4.2.5	Artefacts	15
5.	Équipe de développement	15
6.	Entente contractuelle proposée	17
6.1	Description du type de l'entente contractuelle	17
6.2	Objet de l'entente	17
6.3	Échéancier de l'entente	17
6.4	Couts et charges liés à l'entente	17
6.5	Critères acceptation et méthodes d'évaluation	17
6.6	Correspondance	18

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

# Plan de projet

## 1. Introduction

Le projet à développer est une continuité du projet intégrateur INF2990 proposé en hiver 2016. L'appel d'offres a été publié par le client, l'école Polytechnique de Montréal représenté par Olivier Gendreau, le lundi 9 janvier 2017. Ce dernier demande la remise de deux livrables. Le premier comprend deux prototypes de communication : Client lourd – serveur et Client léger – serveur et le second, le produit final. Le but est de concevoir un simulateur de robot en réseau accessible par le biais du Client lourd ou du Client léger tous deux connecté au même serveur. Ce document permet de définir le projet de façon globale et de révéler les moyens mis en œuvre pour répondre à l'appel d'offres. Tout d'abord, il donne l'énoncé des travaux. Puis il explique les différentes stratégies de gestion et le suivi de l'avancement. Après il détaille l'échéancier du projet. Ensuite il présente l'équipe de développement. Et enfin, il décrit l'entente contractuelle proposée.

## 2. Énoncé des travaux

### 2.1 Solution proposée

L'école polytechnique de Montréal a lancé un appel d'offres consistant à réaliser une application interactive en ligne (connectée à un serveur) simulant les déplacements d'un robot sur une scène.

Le simulateur de robot réalisé dans le cadre du cours INF2990 constitue la base du produit demandé dans le cadre de cet appel d'offres. Le présent projet est une extension de ce simulateur dont l'objectif principal est d'ajouter deux nouvelles fonctionnalités : la possibilité d'éditer simultanément en réseau ainsi que l'édition de zone de simulation à partir d'un iPad.

L'équipe de projet numéro 9 a répondu à cet appel d'offres et s'est engagée envers l'École polytechnique de Montréal à réaliser le produit demandé tout en proposant la solution suivante :

Diviser le produit issu du projet de deuxième année en trois exécutables distincts:

- 1) Un client lourd (éditeur et simulation sur PC) et un serveur : l'utilisateur non connecté au serveur doit pouvoir utiliser les fonctionnalités qu'offre le projet de deuxième année (édition, simulation). S'il est connecté au serveur, il doit pouvoir éditer et de simuler des cartes en même temps que d'autres utilisateurs connectés au même serveur et à temps réel. Aussi l'application client lourd doit offrir un environnement de clavardage pour les utilisateurs connectés au même serveur.
- 2) Un client léger (éditeur sur iPad) : un utilisateur sur client léger non connecté au serveur doit pouvoir profiter des fonctionnalités liées à l'édition présente dans le projet 2, mais importées sur l'iPad mini. Si cet utilisateur se connecte au serveur, alors il doit pouvoir éditer et de simuler des cartes en même temps que d'autres utilisateurs connectés au même serveur et à temps réel. Aussi l'application client léger doit également offrir un environnement de clavardage pour les utilisateurs connectés au même serveur.
- 3) Un site web internet : à partir de ce serveur, l'utilisateur doit être capable de visualiser toutes les informations relatives à son profil.

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

## 2.2 Hypothèses et contraintes

La réalisation du projet WALL-E2 par les membres de l'équipe 9 suite à la réponse à l'appel d'offres lancé par l'école polytechnique de Montréal rencontre les contraintes suivantes :

- Pour la satisfaction des exigences demandées par le client, l'équipe 9 doit prendre comme base le produit réalisé en projet de deuxième année du génie logiciel/informatique (simulateur de robot). Il faut préciser que deux membres de l'équipe 9 ont réalisé un autre projet au cours de leur projet de deuxième année à savoir : un hockey sur cousin d'air.
- Le projet 2 qui constitue le point de départ du projet 3 a été réalisé sur l'environnement de développement Visual Studio 2013, alors que le projet 3 exige que le produit final compile sur l'environnement de développement Visual Studio 2015.
- La partie réseautique de ce projet exige quelques connaissances acquises lors du cours INF3405 (Réseaux informatiques). Alors qu'un membre de l'équipe 9 n'a pas encore suivi ce cours.
- L'implémentation des interfaces du client lourd doit être en C#.
- L'application client lourd doit être fonctionnelle sous le système d'exploitation Windows 10.
- Le client léger doit fonctionner sur IOS à partir d'un iPad mini fourni par l'école Polytechnique de Montréal.
- L'appel d'offres lancé par l'école polytechnique de Montréal impose un échéancier précis (du 09-01-1017 jusqu'au 11-04-2017).
- Les membres de l'équipe 9 sont des étudiants en troisièmes années de génie logiciel. Par conséquent ils ont d'autres cours en parallèle avec le projet 3 ce qui signifie qu'ils ont une charge de travail importante. Cela exige une très bonne organisation des activités.

## 2.3 Biens livrables du projet

L'entrepreneur s'est engagé à fournir deux livrables pour ce projet 3.

Voici le contenu des deux livrables suivis de leur date de livraison :

- Livrable 1 (doit être livré au plus tard le 10/02/2017) :

Le livrable 1 consiste à fournir au client un ensemble d'artéfacts se résumant dans les documents suivants :

- Plan de projet
- Spécification des requis du système (SRS)
- Liste d'exigences
- Document d'architecture logicielle
- Protocole de communication

En plus de ces artefacts, l'entrepreneur doit réaliser deux prototypes de communication : un pour le client lourd et un pour le client léger.

- Livrable 2 (doit être livré au plus tard le 11/04/2017) :

Après l'approbation des artefacts fournis lors du premier livrable par le client. L'entrepreneur doit remettre un second livrable qui contient les exécutables du produit final (client lourd, client léger, serveur, site web)

En plus les artefacts suivants doivent être fournis lors du livrable 2 :

- Plan de tests
- Résultats de tests

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

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

#### 3.1 Gestion des exigences

Le choix des exigences se fait en équipe plus précisément en sous-équipes de travail. Une première sous équipe détermine requis liés au client léger et une deuxième sous équipe s'occupe des requis du client lourd et du serveur. Les requis sont alors énumérés dans la liste des exigences et décrits de façon plus détaillée dans le SRS. Toute l'équipe de développement doit approuver ces deux documents avant de les transmettre au client. C'est la première phase de la spécification des requis.

La deuxième phase consiste alors à attendre les commentaires du client à propos des nouvelles exigences envoyées. Celui-ci peut les modifier, les refuser ou les accepter. À partir de ce moment, deux situations sont possibles.

La première situation est le cas où toutes les exigences sont approuvées par le client. Si le projet n'est pas entamé, il peut être amorcé. Mais si les modifications sont faites en cours de projet, il faut d'abord que l'équipe de développement revoie son organisation et sa planification. Elle doit prévoir des mesures correctives, les évaluer et les mettre en place rapidement. Une fois cette démarche achevée, l'équipe peut continuer à développer le projet.

La deuxième situation est le cas où certaines nouvelles exigences proposées par l'équipe sont refusées ou modifiées par le client. Il faut alors répéter la première phase et la seconde phase de la spécification des requis dans le temps imparti jusqu'à ce que toutes les exigences soient approuvées par le client et l'équipe.

#### 3.2 Contrôle de la qualité

Le premier livrable rassemble deux prototypes de communication (client lourd – serveur et client léger – serveur) et d'un ensemble d'artefacts (SRS, liste des exigences, plan de projet, document d'architecture logicielle et protocole de communication).

La majorité des biens livrables sont des documents à compléter et à rédiger. Même si certains éléments à transcrire doivent être discutés et fixés par l'ensemble du groupe, les documents sont divisés en plusieurs parties et la rédaction de chacune de ces parties se fait de façon individuelle. Par conséquent, pour assurer la qualité de ce livrable, il faut que chaque membre de l'équipe s'assure que toutes les informations inscrites dans les documents sont justes, pertinentes et complètes, que la mise en page et l'orthographe sont irréprochables et qu'aucune partie n'est manquante. Ce livrable possède également deux prototypes de communication qui doivent absolument être testés par toute l'équipe. Le but est d'implémenter le clavardage dans chacun des clients (lourd et léger). Cela signifie que chaque membre de l'équipe doit tenter de faire communiquer plusieurs clients sur le client lourd, plusieurs clients sur le client léger et pour finir plusieurs clients à la fois sur le client lourd et sur le client léger. Pour se faire, il faut échanger plusieurs messages entre les utilisateurs. L'idée est de tester que le clavardage est bien conforme aux requis des prototypes de communication demandés.

Le second livrable regroupe en majeure partie du code et des exécutables ainsi que deux documents à rédiger liés aux tests logiciels (plan de tests logiciels et résultats de tests logiciels). Pour s'assurer de la qualité du second livrable, chaque bien livrable doit être traité différemment.

Le code doit être testé par des tests logiciels. Évidemment les membres du groupe chargé d'implémenter les tests logiciels ne peuvent pas tester tout le code. Le but est de concevoir des tests pertinents sur certaines parties du code sensibles afin de déceler les anomalies non détectées et de vérifier que le logiciel respecte les spécifications.

Les exécutables doivent être testés par chacun des membres de l'équipe. L'idée est de se comporter comme un utilisateur quelconque qui découvre l'application tout en restant focalisée sur les exigences. Dans un premier temps chaque membre examine le plus sérieusement que possible la partie qu'il a conçue. Puis dans un second temps il peut vérifier une partie du logiciel qu'il n'a pas implémentée tout en se basant sur les requis de cette partie. Le but est d'essayer le maximum de situations possibles afin de repérer les éventuels bogues ou erreurs. Et enfin, s'il reste suffisamment de temps et que cela n'implique pas de dévoiler des informations privées, une personne extérieure au groupe peut également tester l'application afin de donner son avis à l'équipe de développement.

Les artefacts liés aux tests logiciels sont, quant à eux, traités de la même façon que ceux du premier livrable.

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

### 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. Ex : Fiabilité, Complétude, Compatibilité, Cohésion, etc. (problèmes métriques logiciels) LOG 1000
- Stratégies de gestion : mesures à prendre afin de gérer le risque.

< Risque n°1 > - mauvaise spécification des requis				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
8	Les exigences sont un facteur de risque non négligeable dans un projet. En effet parmi tous les principaux problèmes rencontrés en cours de projet les exigences incomplètes et/ou changeantes sont le premier facteur de risque rapporté. En cas de mauvaise spécification des requis, le client risque de ne pas être satisfait du produit final proposé par l'équipe de développement. De plus la planification et l'organisation du projet risquent d'être lourdement affectées par ce risque. Le temps alloué à chaque tâche doit être recalculé, les délais risquent de ne pas être respectés. Aussi plus un problème lié aux exigences est découvert tard au cours du processus plus il est coûteux. Et enfin la qualité du produit peut être diminuée.	E	Validité (le logiciel est conforme aux fonctionnalités décrites dans les spécifications)	Prévenir le risque : Consacrer beaucoup de temps à la spécification des requis. S'entretenir régulièrement avec le client, comprendre et traduire ses besoins, bien définir les fonctionnalités dans le SRS et communiquer souvent avec son équipe.

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

< Risque n°2 > - mauvaise estimation de l'effort				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
6	<p>Dans un projet logiciel, il est très difficile d'estimer l'effort que demande chacune des tâches à réaliser. En effet certaines tâches peuvent être accomplies plus rapidement que prévu, mais en général c'est plutôt le contraire. L'effort est souvent mal estimé à cause d'oublies ou d'équipes de projet trop optimistes.</p> <p>Une mauvaise estimation de l'effort peut avoir des conséquences négatives sur la planification et l'organisation du projet. Si plusieurs tâches sont liées, le retard provoqué par la première va forcément entraîner un retard chez la seconde et ainsi de suite. L'échéancier risque de ne pas être respecté.</p>	M	<p>Validité (le logiciel est conforme aux fonctionnalités décrites dans les spécifications)</p> <p>Vérifiabilité (la conception du logiciel simplifie la préparation aux procédures de tests)</p> <p>Fiabilité (le logiciel peut fonctionner même dans des conditions anormales)</p>	<p>Prévenir le risque : Essayer de planifier le plus sérieusement que possible le projet même la durée de certaines tâches est difficile à estimer. Mettre en place un calendrier. Faire un suivi de l'avancement régulièrement.</p> <p>Accepter le risque : L'équipe de développement est consciente que ce risque est l'un des plus fréquents dans un projet. Il faut accomplir la tâche même si elle est plus longue que prévu et il faut essayer de perdre le moins de temps que possible.</p>

< Risque n°3 > - Problème de compatibilité entre les différentes technologies				
Ampleur	Description	Impact	Facteurs	Stratégie de gestion
7	<p>Ce projet demande de mettre en relation un certain nombre de technologies ce qui peut provoquer des problèmes de compatibilité. En effet il faut implémenter et mettre en place un client lourd, un serveur, une base de données, un site web et un client léger. Ces technologies sont toutes différentes. Certaines doivent échanger des informations et des données, communiquer et s'exécuter ensemble parfois de manière synchrone. Elles possèdent toutes des environnements, librairies et langages différents et propres à chacune ce qui augmente les possibilités qu'elles soient incompatibles.</p>	E	<p>Validité (le logiciel est conforme aux fonctionnalités décrites dans les spécifications)</p> <p>Compatibilité (le logiciel est facilement compatible avec d'autres logiciels)</p> <p>Portabilité (le logiciel peut facilement être transféré sous d'autres environnements logiciels et matériels)</p>	<p>Prévenir le risque : Avant de commencer à réaliser les différentes tâches, faire des recherches sur les technologies à implémenter et à mettre en relation afin de savoir si, par exemple, des librairies ou des langages sont plus compatibles que d'autres.</p> <p>Réduire l'incertitude : Faire des tests fréquents de façon à vérifier que les technologies sont compatibles. Communiquer de façon régulière avec les membres de l'équipe qui travaillent sur l'autre technologie. Vérifier continuellement le suivi de l'avancement.</p>



<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

### 3.4 Gestion de configuration

La gestion de configuration se fait grâce au logiciel de gestion de version « Git ». L'idée est d'utiliser un système de branches pour faciliter l'intégration de chaque partie du projet. La branche « master » est la branche principale du projet. Elle est utilisée uniquement pour remettre les biens livrables une fois qu'ils sont terminés. Dans cette situation elle est utilisée deux fois, la première fois pour remettre le premier livrable et la seconde pour le produit final. Seul un produit fonctionnel et possiblement achevé peut être placé sur cette branche. Juste en dessous de la branche « master » une branche « develop » est créée. Cette branche permet l'intégration de chaque lot de travaux et fonctionnalités du projet afin d'avoir un seul et même produit comportant tout le code, tous les fichiers et tous les objets utilisés jusqu'à présent. Une fois les biens livrables complets cette branche est simplement fusionnée avec la branche « master ». Et enfin, à partir de la branche « develop » de multiples autres branches sont créées. Il est essentiel de préciser que chaque personne possède son propre lot de travail et ses propres fonctionnalités à implémenter et que chaque nouvel ajout peut avoir un impact sur la partie de quelqu'un d'autre. Par conséquent il est difficile de travailler tous ensemble sur la même branche. C'est pourquoi de nouvelles branches sont créées et utilisées pour chaque lot de travail ou fonctionnalité par chaque membre de l'équipe à chaque fois que cela est nécessaire.

En parallèle au logiciel « Git » l'équipe a décidé de se réunir régulièrement et d'utiliser Redmine et Facebook afin de planifier et de vérifier les tâches de chacun, de se tenir informé de l'avancement du projet et de signaler les changements, les problèmes et les oublis. Facebook facilite la communication rapide, l'échange d'informations, la prise de rendez-vous et l'envoi de certains documents. Redmine permet un suivi constant de l'avancement et donne le calendrier des tâches à effectuer. Et enfin les réunions permettent à l'équipe d'être à jour, d'échanger des indications et de s'entre-aider. Tout cela participe au processus qui permet de revoir et disposer des problèmes et changements.

La norme des artefacts est la suivante :

Nom\_De\_Artefact\_ÉquipeXX.extension (où XX est le numéro de l'équipe)

Par exemple la liste des exigences est nommée de la façon suivante : Liste\_Des\_Exigences\_Équipe09.xlsx

## 4. Échéancier du projet

### 4.1 Livrable 1 : Réponse à l'appel d'offres et remise des prototypes (Date de tombée : 10 Février 2017)

- Documents du projet et appel d'offres
- Prendre connaissance de tous les documents liés au projet et comprendre l'appel d'offres.
- Effort estimé : 3 h de travail par personne (effectuées en partie en groupe et en partie individuellement)
- Date de début : 9 Janvier 2017
- Date de fin : 15 Janvier 2017

#### 4.1.1 Liste des exigences

- Proposer des exigences essentielles, souhaitables et optionnelles pertinentes liées au projet et les lister dans un document type Excel (extension .xlsx).
- Effort estimé : 15 h de travail effectué en groupe
- Date de début : 11 Janvier 2017
- Date de fin : 8 Février 2017

#### 4.1.2 SRS (Spécification des requis du système)

- Rédiger l'artefact « SRS » en s'aidant de la liste des exigences. Mettre à jour le document si besoin (modification de la liste des exigences).

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

- Effort estimé : 36 h de travail soit 6h/personne
- Date de début : 16 Janvier 2017
- Date de fin : 9 Février 2017

#### 4.1.3 *Plan de projet*

- Rédiger l'artefact « Plan de projet » afin de décrire le projet de façon générale (les travaux à effectuer, gestion et suivi de l'avancement, échéancier, équipe de développement et entente contractuelle).
- Effort estimé : 12 h de travail soit 6h/personne (2 personnes)
- Date de début : 30 Janvier 2017
- Date de fin : 6 Février 2017

#### 4.1.4 *Document d'architecture logicielle*

- Rédiger l'artefact « Document d'architecture logicielle ».
- Effort estimé : 36 h de travail soit 9h/personne (4 personnes)
- Date de début : 30 Janvier 2017
- Date de fin : 10 Février 2017

#### 4.1.5 *Prototypes de communication*

- Mettre en place un prototype de communication entre le client lourd et le serveur ainsi qu'un prototype de communication entre le client léger et le serveur. Le but de ces prototypes est de pouvoir envoyer, recevoir et diffuser des messages (clavardage).
- Effort estimé : 30 h de travail soit 15h/personne (2 personnes)
- Date de début : 11 Janvier 2017
- Date de fin : 6 Février 2017

#### 4.1.6 *Protocole de communication*

- Rédiger l'artefact « Protocole de communication » afin d'expliquer comment fonctionnent les prototypes de communication et d'indiquer quelles sont les données échangées.
- Effort estimé : 6 h de travail soit 3h/personnes (2 personnes)
- Date de début : 1 Février 2017
- Date de fin : 10 Février 2017

### 4.2 **Livrable 2 : Remise du produit final (Date de tombée : 11 Avril 2017)**

#### 4.2.1 *Serveur*

##### 4.2.1.1 *Liaison client-serveur*

- Mettre en place un serveur « relais » capable de communiquer avec dans le but de rendre possible le clavardage.
- Effort estimé : 10 h de travail soit 5h/personne (2 personnes)

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

- Date de début : 16 Janvier 2017
- Date de fin : 4 Avril 2017

#### 4.2.1.2 Enregistrement des données et gestion de la base de données

- Mettre en place la communication entre le serveur et la base de données et enregistrer dans la base de données les données échangées pertinentes.
- Effort estimé : 40 h de travail soit 20h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.1.3 Liaison des activités d'édition

- Diffuser et rendre accessibles à partir du serveur les activités d'édition effectuées sur les deux clients.
- Effort estimé : 40 h de travail soit 10h/personne (4 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.1.4 Liaison des activités de simulation

- Diffuser et rendre accessibles à partir du serveur les activités de simulation effectuées sur le client lourd.
- Effort estimé : 20 h de travail soit 10h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.2 Site Web

- Mettre en place un site Web. Ce site affiche des informations concernant les différentes cartes disponibles et des renseignements privés sur chaque profil.
- Effort estimé : 40 h de travail soit 20h/personnes (2 personnes)
- Date de début : 25 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3 Client lourd

##### 4.2.3.1 Clavardage – intégration

- Implémenter la possibilité d'avoir le clavardage en mode fenêtré ou en mode intégré.
- Effort estimé : 30 h de travail soit 15h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

##### 4.2.3.2 Clavardage – canaux de discussion

- Implémenter la possibilité d'avoir plusieurs canaux de discussions au sein du clavardage.
- Effort estimé : 15 h de travail soit 5h/personne (3 personnes)
- Date de début : 13 Février 2017

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

- Date de fin : 4 Avril 2017

#### 4.2.3.3 Clavardage – exigences souhaitables

- Améliorer le clavardage en implémentant diverses options et fonctionnalités additionnelles (8 en tout).
- Effort estimé : 60 h de travail soit 30h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.4 Édition en ligne

- Améliorer le projet intégrateur INF2990 d'hiver 2016 de façon à rendre l'édition en ligne possible. Rendre cette édition visible même pour un simple spectateur. Augmenter les possibilités des utilisateurs lors de l'édition en ligne (annuler/rétablir une action et enregistrer les actions dans un fichier) en se référant aux exigences souhaitables.
- Effort estimé : 80 h de travail soit 40h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.5 Simulation en ligne

- Améliorer le projet intégrateur INF2990 d'hiver 2016 de façon à rendre la simulation en ligne possible. Rendre cette simulation visible même pour un simple spectateur.
- Effort estimé : 80 h de travail soit 40h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.6 Gestion de la déconnexion en simulation

- Mettre en place une gestion de la déconnexion lorsqu'un usager se déconnecte au cours d'une simulation.
- Effort estimé : 15 h de travail soit un peu plus de 7h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.7 Profil utilisateur

- Mettre en place un système de profil permettant de s'authentifier au serveur. Avoir un profil utilisateur permet d'accéder à un certain nombre de données? De bénéficier d'un certain nombre d'avantages et de profiter d'options supplémentaires (exigences essentielles et souhaitables).
- Effort estimé : 70 h de travail soit 35h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.8 Confidentialité des cartes en ligne

- Implémenter la possibilité de rendre une carte publique ou privée lors de sa création.

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

- Effort estimé : 15 h de travail soit un peu plus de 7h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.9 Sauvegarde des cartes

- Implémenter une sauvegarde automatique des cartes sur le client lourd.
- Effort estimé : 10 h de travail soit 5h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.10 Tutoriel

- Implémenter un tutoriel dans le mode édition et un autre dans le mode simulation pour guider l'utilisateur sur le client lourd.
- Effort estimé : 40 h de travail soit 20h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.11 Autres exigences souhaitables

- Implémenter d'autres exigences souhaitables telles que la déconnexion automatique en cas de non-activité, la présence de notifications sonores, la possibilité de noter les cartes, la présentation des cartes ou encore la sauvegarde des éléments du profil en cas de déconnexion et reconnexion.
- Effort estimé : 50 h de travail soit 25h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.3.12 Exigences optionnelles

- Implémenter des fonctionnalités optionnelles en lien avec les cartes : afficher le nom de la carte ainsi que l'emplacement de sa sauvegarde (local ou distant) et enregistrer les cartes sous forme d'images.
- Effort estimé : 10 h de travail soit 5h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

### 4.2.4 Client léger

#### 4.2.4.1 Éditeur

- Implémenter en Swift toutes les fonctionnalités de l'édition présentes dans le projet intégrateur INF2990 d'hiver 2016 pour que l'utilisateur puisse éditer à partir de l'application mobile sur son iPad.
- Effort estimé : 80 h de travail soit 40h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

#### 4.2.4.2 Interface utilisateur

- Implémenter plusieurs cas d'utilisation utilisant trois gestuels de façon concurrente.
- Effort estimé : 25 h de travail soit un peu plus de 12h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.3 Effets sonores et visuels

- Implémenter des cas d'utilisation utilisant trois groupes d'effets différents.
- Effort estimé : 15 h de travail soit un peu plus de 7h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.4 Tutoriel

- Implémenter un tutoriel afin d'expliquer le mode édition du client léger à l'utilisateur.
- Effort estimé : 22 h de travail soit 11h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.5 Présentation des cartes

- Mettre en place un système de vignettes automatiquement mises à jour afin de présenter les cartes disponibles
- Effort estimé : 10 h de travail soit 5h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.6 Sauvegarde des cartes

- Mettre en place une synchronisation automatique entre le contenu du serveur et du client léger et une gestion des droits d'auteur.
- Effort estimé : 40 h de travail soit 20h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.7 Édition en ligne

- Rendre l'édition en ligne possible sur le client léger.
- Effort estimé : 50 h de travail soit 25h/personnes
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.8 Application mobile

- Implémenter un certain nombre d'exigences souhaitables en lien avec l'application mobile : présence d'une

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

animation lors de l'ouverture de l'application, possibilité de personnaliser l'application, accès au site web à partir de l'application, possibilité de consulter les statistiques des cartes disponibles et utilisation du TouchID.

- Effort estimé : 40 h de travail soit 20h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.9 Clavardage

- Implémenter le clavardage au sein du client léger avec la présence de notifications en cas de message, d'émoticônes et d'un historique des discussions ainsi que la possibilité de catégoriser les contacts.
- Effort estimé : 40 h de travail soit 20h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.10 Profil utilisateur

- Mettre en place un système de profil utilisateur dans le client léger avec la possibilité de consulter ses statistiques.
- Effort estimé : 16 h de travail soit 8h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.4.11 Exigences optionnelles

- Implémenter des fonctionnalités optionnelles : possibilité de noter les cartes, de consulter les astuces d'utilisation et de charger aléatoirement une carte.
- Effort estimé : 10 h de travail soit 5h/personne (2 personnes)
- Date de début : 13 Février 2017
- Date de fin : 4 Avril 2017

#### 4.2.5 Artefacts

- Mettre à jour les artefacts, rédiger les deux documents liés aux tests : plan de tests logiciels et résultats de tests logiciels.
- Effort estimé : 12h de travail soit 2h/personnes
- Date de début : 3 Avril 2017
- Date de fin : 10 Avril 2017

## 5. Équipe de développement

L'équipe 9 se compose officiellement de 6 étudiants (futurs ingénieurs) de l'école Polytechnique de Montréal en génie logiciel. Ces membres possèdent des compétences techniques pertinentes et essentielles pour l'élaboration du projet WALL-E2. Ils vont mettre en pratique, tout au long du projet, les notions de gestion de projet acquises au cours des

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

premières années de génie logiciel. Cela implique de faire la distinction entre ingénieur et gestionnaire. Par conséquent, tout le monde participe au développement, certains s'occupent des tâches de gestion.

Voici la Description de l'expertise des membres de l'équipe 9 et leurs responsabilités respectives :

- **Amrani Abdellatif :** Développeur C#, Abdellatif est un étudiant en troisièmes années de génie logiciel à l'école Polytechnique de Montréal. Avant de commencer ses études dans cette école, Abdellatif a obtenu une licence en informatique et a occupé un poste d'enseignant en informatique au secondaire. Cet emploi lui a permis d'acquérir des compétences pédagogiques qui s'avèrent utiles pour l'organisation des activités réalisées par l'équipe 9. Abdelatif travaille de son côté sur les fonctionnalités relatives au client lourd ainsi que le développement du site web tout en s'appuyant sur ses connaissances acquises lors du cours LOG4420 (conception de site web dynamique et transactionnel). D'un autre côté, il a effectué un stage de 4 mois en vision par ordinateur en C# ce qu'il lui a permis de se familiariser avec la conception des interfaces en C#.
- **Djoudi Essaid :** Développeur C++ et C#, il étudie à l'école polytechnique de Montréal. Essaid a suivi en parallèle de sa formation en génie logicielle, une formation de 1650 heures en gestion de projet. Ce qui lui permet de s'investir en tant que gestionnaire de ce projet. Il s'occupe de l'organisation et de la planification des tâches dans Redmine afin d'assurer une bonne planification et un excellent déroulement des tâches du projet. D'un autre côté, Essaid travaille sur l'implémentation des différentes fonctionnalités du client lourd ainsi que la gestion de la base de données et la communication du serveur avec cette base de données.
- **El kaoukabi Samir :** Samir est un développeur C++ et Swift. Avant d'entamer sa formation en génie logiciel, Samir a obtenu un DEC technique en système d'informations et un certificat CISCO CCNA. Il possède également une expérience préalable sur le multithreading. Au cours de son stage il a mis en pratique les notions acquises en base de données (relationnel, azur SQL, mongo DB), conception d'interface en C# ainsi que la maintenance des serveurs. Samir s'est engagé à travailler sur la conception du serveur liant les utilisateurs du client lourd et ceux du client léger, implémenter un protocole de communication pour le client léger ainsi que l'implémentation de certaines fonctionnalités du client lourd.
- **Jalil Samia :** Samia est développeur Swift et développeur C++. Avant d'entamer sa formation en génie logiciel, Samia a obtenu un DESS en administration réseau (serveur DNS, serveur web (IIS, APACHE), serveur DHCP). Dans le cadre de ce projet, Samia s'occupe de la conception du serveur liant les utilisateurs du client lourd et ceux du client léger en collaboration avec Samir. D'un autre côté, elle participe à l'immigration du code vers l'application IOS (client léger) sur l'iPad.
- **Renard Morgane :** Morgane est développeur Swift et développeur C++. Au cours du projet 2 d'INF2990 Morgane a acquis plusieurs notions dans la conception des interfaces en C# ainsi qu'en qualité des interfaces utilisateurs. Ce qui l'engage à s'occuper de la création des interfaces liées au client lourd et surtout celles liées au client léger. D'un autre côté, Morgane travaille sur la migration du code du projet de deuxième année vers IOS ainsi que sur l'implémentation de différentes fonctionnalités au niveau du client léger.
- **Winnie Tchoutang:** Développeur Swift, développeur C++, Winnie a préalablement effectué un stage à l'extérieur de l'école Polytechnique de Montréal. Dans ce dernier elle a acquis plusieurs notions sur l'optimisation et la gestion des accès aux bases de données tout en pratiquant les langages suivants VBA VB6 Excel. Elle s'occupe donc de la communication entre serveurs et la base de données. Winnie travaille aux côtés de Samia et Morgane sur la migration du code vers iOS appliqué à l'iPad ainsi que sur l'implémentation de différentes fonctionnalités au niveau du client léger.



<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

## 6. Entente contractuelle proposée

Dans le cadre du projet 3 du cours LOG3900 (Projet d'évolution d'un logiciel), l'école Polytechnique de Montréal fait appel au service de l'équipe 9 pour la réalisation du projet WALL-E2 sous une entente contractuelle du type clé en main prix ferme.

### 6.1 Description du type de l'entente contractuelle

Une entente contractuelle du type clé en main prix ferme signifie que l'équipe 9 doit livrer le produit final (projet WALL-E2). Le paiement final est émis uniquement lors de la livraison et de l'acceptation du produit par le client.

### 6.2 Objet de l'entente

Le simulateur du robot suiveur de ligne réalisé dans le cadre du cours INF2990 constitue la base du produit demandé dans le cadre de cet appel d'offres. Le projet 3 est une extension de ce simulateur dont l'objectif principal est d'ajouter trois nouvelles fonctionnalités :

- La possibilité d'éditer et de simuler des cartes simultanément par des utilisateurs en réseau à partir d'un ordinateur lié à un serveur. (Client lourd)
- La migration des fonctionnalités d'édition de zone de simulation vers un iPad mini.
- La possibilité d'éditer des cartes simultanément en réseau à partir d'un iPad.

Aussi l'appel d'offres impose aux entrepreneurs (équipe 9) de soumettre des artefacts incluant les documents suivants :

- Plan de projet
- Spécification des requis du système (SRS)
- Document d'architecture logicielle
- Protocole de communication
- Plan de tests logiciels
- Résultats de tests logiciels

Ainsi qu'un fichier Excel contenant l'ensemble des exigences fonctionnelles liées à la réalisation du projet WALL-E2.

### 6.3 Échéancier de l'entente

La réponse à l'appel d'offre soumise par l'école Polytechnique de Montréal oblige l'entrepreneur (équipe 9) de commencer les travaux faisant l'objet du présent contrat le 09 janvier 2016 et a terminé ces derniers le 11 avril 2016.

### 6.4 Coûts et charges liés à l'entente

La réalisation et la livraison du projet 3 (WALL-E2) requièrent une charge de travail de 1080 heures-personnes pour une équipe de 6 personnes. Le client s'engage à verser 113000 \$ aux entrepreneurs répondant à son appel d'offres. Après estimation, la réalisation du projet 3 comprend 900 heures de développement et 180 heures de gestion et que le salaire d'un développeur logiciel et d'un gestionnaire sont respectivement de 100\$/heure et de 125\$/heure.

### 6.5 Critères acceptation et méthodes d'évaluation

Lors de la réalisation et la livraison du projet 3 (WALL-E2) par l'entrepreneur, l'école Polytechnique de Montréal va évaluer le produit final selon une échelle d'appréciation allant de A\* à F. Cette évaluation détermine si le projet livré par l'entrepreneur est réussi.

<WALL-E2>	Version : 4.0
Plan de projet	Date : 2017-04-11

## 6.6 Correspondance

- **École polytechnique de Montréal :**

Responsable : Olivier Gendreau,

Adresse : 2500, chemin de la Polytechnique, Montréal (Québec), H3T 1J4,

Local : M-4017

Téléphone : (514) 340-4711 poste 2102

Courriel électronique: [olivier.gendreau@polymtl.ca](mailto:olivier.gendreau@polymtl.ca)

- **Équipe 9 :**

Représentant : Djoudi Essaid

Adresse : 2405 rue Bélanger, Montréal (Québec)

Téléphone : 438-405-3144

Courriel électronique : [essaid.djoudi@polymtl.ca](mailto:essaid.djoudi@polymtl.ca)

En foi de quoi, les parties ont signé à \_\_\_\_\_, ce \_\_\_\_ ième jour de \_\_\_\_\_

.

\_\_\_\_\_  
Signature du producteur

\_\_\_\_\_  
Signature de l'artiste, contractuel ou technicien