



**POLYTECHNIQUE
MONTRÉAL**

LE GÉNIE
EN PREMIÈRE CLASSE

Département de génie informatique et génie logiciel

INF3995

Projet de conception d'un système informatique

Proposition répondant à l'appel d'offres
no. H2018-INF3995 du département GIGL.

Système mobile de télémétrie en temps réel

Équipe No. 09

Anthony Abboud

Gabriel Demers

Khalil Bennani

Luc Courbariaux

Mounia Nordine

Reph Dauphin Mombrun

Février 2018

Table des matières

Vue d'ensemble du projet	2
But du projet, porté et objectifs (Q2.1 et Q4.1)	3
Hypothèse et contraintes (Q3.1)	3
Biens livrables du projet (Q2.3)	4
Organisation du projet	5
Structure d'organisation	5
Entente contractuelle	5
Solution proposée	6
Architecture logicielle sur serveur (Q4.2)	6
Architecture logicielle sur tablette (Q4.3)	6
Processus de gestion	6
Estimations des coûts du projet	6
Planification des tâches (Q2.2)	7
Calendrier de projet (Q3.3)	9
Ressources humaines du projet	11
Suivi de projet et contrôle	11
Contrôle de la qualité	11
Gestion de risque (Q2.6)	12
Tests (Q4.4)	12
Gestion de configuration	13
Références (Q3.2)	13

Vue d'ensemble du projet

1.1 But du projet, porté et objectifs (Q2.1 et Q4.1)

Le but du projet est de développer une plateforme qui sera en charge de la jonction entre des applications clients et le service de télémétrie embarqué sur la fusée Oronos, en tenant compte des limitations matérielles (notamment de mémoire) de celle-ci.

L'objectif du livrable 1 est de livrer un prototype fonctionnel répondant aux exigences d'implémentation, tandis que le livrable 2 apportera un esthétisme et une ergonomie à laquelle on s'attend dans une application mature.

1.2 Hypothèse et contraintes (Q3.1)

Ce document sert à donner une première idée des coûts et délais requis pour la livraison d'un produit répondant aux requis. Dans cette optique, nos projections sont délibérément pessimistes, pour garantir que les coûts du projet réel ne dépassent pas nos projections et que le client puisse mettre foi dans celles-ci.

Le client a le droit de modifier ses exigences en cours de route, mais devra alors s'attendre à des différences entre les coûts et délais indiqués dans ce document et ceux réalisés.

Les détails d'élaboration de l'application non-précisés dans les requis sont laissés à notre discrétion (nous demanderons cependant des précisions en cas d'omission d'éléments non triviaux dans le projet).

Le logiciel produit lors de ce projet sera gardé confidentiel et exclusif au commanditaire (pas de réutilisation ou vente à un concurrent ou partage d'une autre nature de notre part). Celui-ci en est le propriétaire à part entière et en dispose comme bon lui semble.

Une entente pourra être faite quant à la maintenance ou l'amélioration plus poussée du logiciel, une fois ce dernier livré; nos développeurs ayant alors

une expertise sur le sujet, nous serons alors la meilleure option. Dans le cas contraire bien sûr, la qualité de notre produit et de sa documentation ne sera pas un obstacle à la reprise par un tiers.

Au niveau du projet lui-même, on tient pour acquis que la fusée sera à portée d'un réseau Wifi pour retransmettre les données télémétriques, autrement la récupération de celles-ci devra être faite hors-ligne, et que ce réseau sera fourni par un tiers; la fusée sera bel et bien équipée du matériel sur lequel a lieu le développement (ZedBoard Zynq-7000 ARM/FPGA SoC), enfin que les autres informations données dans l'appel d'offre sont véridiques.

1.3 Biens livrables du projet (Q2.3)

Le projet est divisé en deux phases de développement. À la fin de chacune d'elles, un livrable sera remis à une date déjà fixée et ce livrable contiendra, mise à part les fichiers sources, les fichiers nécessaires au déploiement du projet sur des cartes FPGA et des tablettes Android qui respectent les spécifications.

Le jeudi 29 mars 2018 à 8h30, le premier livrable sera délivré et contiendra les fonctionnalités demandées par Oronos, soit le transfert de tout le flot de données entre le serveur et la tablette avec tout le fonctionnement de l'affichage, les comptes des utilisateurs pour permettre la connexion entre le serveur et la tablette, le transfert des fichiers XML et l'implantation des commandes qui permettront la communication avec le serveurs en ligne de commande.

Le jeudi 12 avril 2018 à 8h30, le deuxième livrable qui sera remis permettra de compléter le premier livrable par des thèmes pour l'interface utilisateur de la tablette, une révision de son esthétique, le téléchargement de fichiers optionnels PDF, et l'interface graphique du serveur. Ce dernier livrable inclura également un rapport technique portant sur la réalisation du projet.

2. Organisation du projet

2.1 Structure d'organisation

L'équipe est organisée en trois sous-groupes de deux personnes afin de favoriser autant que possible une méthode de travail en binôme. L'équipe adopte également le réseau de gestion en étoile, i.e. réseau décentralisé pour faciliter la compréhension d'un point de vue globale du projet par tous les membres.

De plus, plusieurs rôles ont été établis au sein de l'équipe, mise à part le fait que tout le monde travaillera comme développeur ou analyste programmeur (certains plus que d'autres):

- Un gestionnaire ou coordonnateur de projets a émané naturellement compte tenu de ses expériences comme directeur (Anthony Abboud);
- Un rôle de support ou soutien technique (basé sur les compétences de chacun et changeant donc selon le sujet);
- Au moins un rôle de testeur, qui sera pris en charge par une personne n'ayant pas réalisé la tâche;
- Un rôle d'avocat du diable ou vigilant, afin de rester sur ses gardes jusqu'à l'échéance pour s'assurer de ne rien manquer ou de ne pas avoir mal compris ou interprété certains requis.

2.2 Entente contractuelle

Le type de contrat choisi pour la réalisation de ce projet est un contrat clé en main parce que nous avons une connaissance exacte de la demande et des spécifications demandées par le client. Ce type de contrat requiert un suivi minimal des travaux. Les coûts du projet seront déterminés dès le début et il n'y aura pas de changement.

Le paiement des prestations (dont le montant est déterminé plus loin dans ce document) sera fait lors des dates-clefs du projet : 40% au début du projet,

40% au premier livrable et finalement 20% au second livrable. Ceci reflète le besoin d'amortir nos coûts initiaux et la somme de travail nécessaire par livrable.

3. Solution proposée

3.1 Architecture logicielle sur serveur (Q4.2)

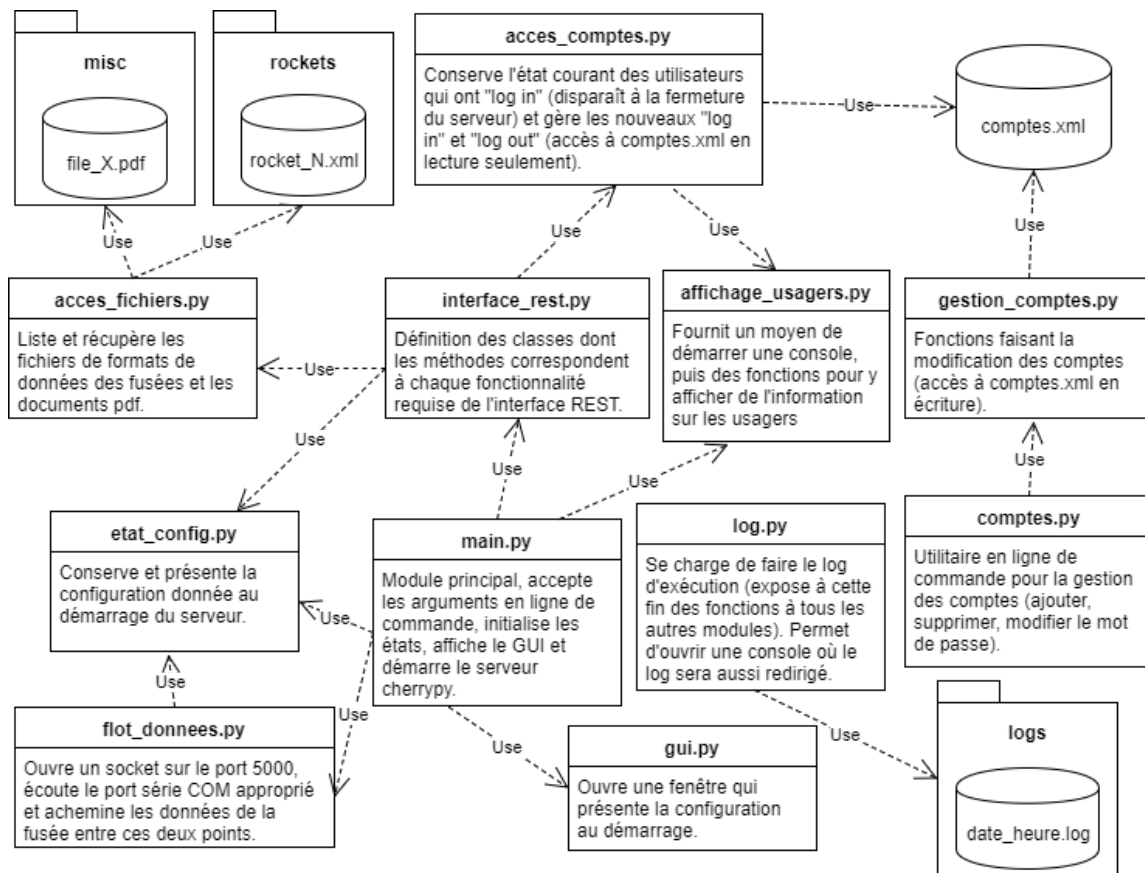


Figure 1 : diagramme de l'architecture du serveur projetée

Afin de concevoir un produit robuste, nous proposons un design hautement modulaire qui met l'emphasis sur la séparation des préoccupations. Au démarrage, le fichier `main.py` est invoqué; celui-ci se charge de tout ce qui est en lien avec l'initialisation. Celui-ci provoque aussi l'affichage du GUI, même si le module responsable de cette tâche est séparé dans le fichier `gui.py`. La

configuration de démarrage alors obtenue est ensuite entreposée dans le module `etat_config.py` pour la durée de l'exécution du serveur. Le travail d'initialisation de `main.py` permet aussi de démarrer le flot de données de la fusée (`flot_donnees.py`, travail effectué en parallèle avec le reste du serveur) et d'ouvrir une console pour l'affichage des usagers (`affichage_usagers.py`).

Enfin, `main.py` démarre CherryPy (bibliothèque choisie pour sa simplicité d'utilisation) avec les classes de `interface_rest.py`. Ce dernier fichier présente les méthodes de l'interface REST requises. Ces méthodes permettent par exemple d'accéder aux comptes (log in, log out). Les méthodes pour cette dernière fonctionnalité sont encapsulées dans `acces_comptes.py`, qui accède en lecture au fichier `comptes.xml` pour l'authentification. `acces_comptes.py` appelle aussi des fonctions de `affichage_usagers.py` pour permettre la mise à jour au fur et à mesure de l'affichage des usagers.

En outre, `interface_rest.py` va aussi aller chercher les fichiers PDF et de fusée via le module `acces_fichiers.py`. Ce module fournit des fonctions générales pour l'accès de fichiers autant applicables pour l'accès aux PDF que l'accès aux XML.

En plus, nous avons aussi un module `log.py` qui est accessible de partout et permet de prendre des logs (création de fichiers `.log` dans le répertoire logs, et affichage dans une console) et d'ouvrir une console (à partir de `main.py`).

Enfin, nous avons un utilitaire `comptes.py` permettant à l'administrateur de gérer (à partir de la ligne de commande) les comptes de `comptes.xml` via les fonctions de `gestion_comptes.py`.

3.2 Architecture logicielle sur tablette (Q4.3)

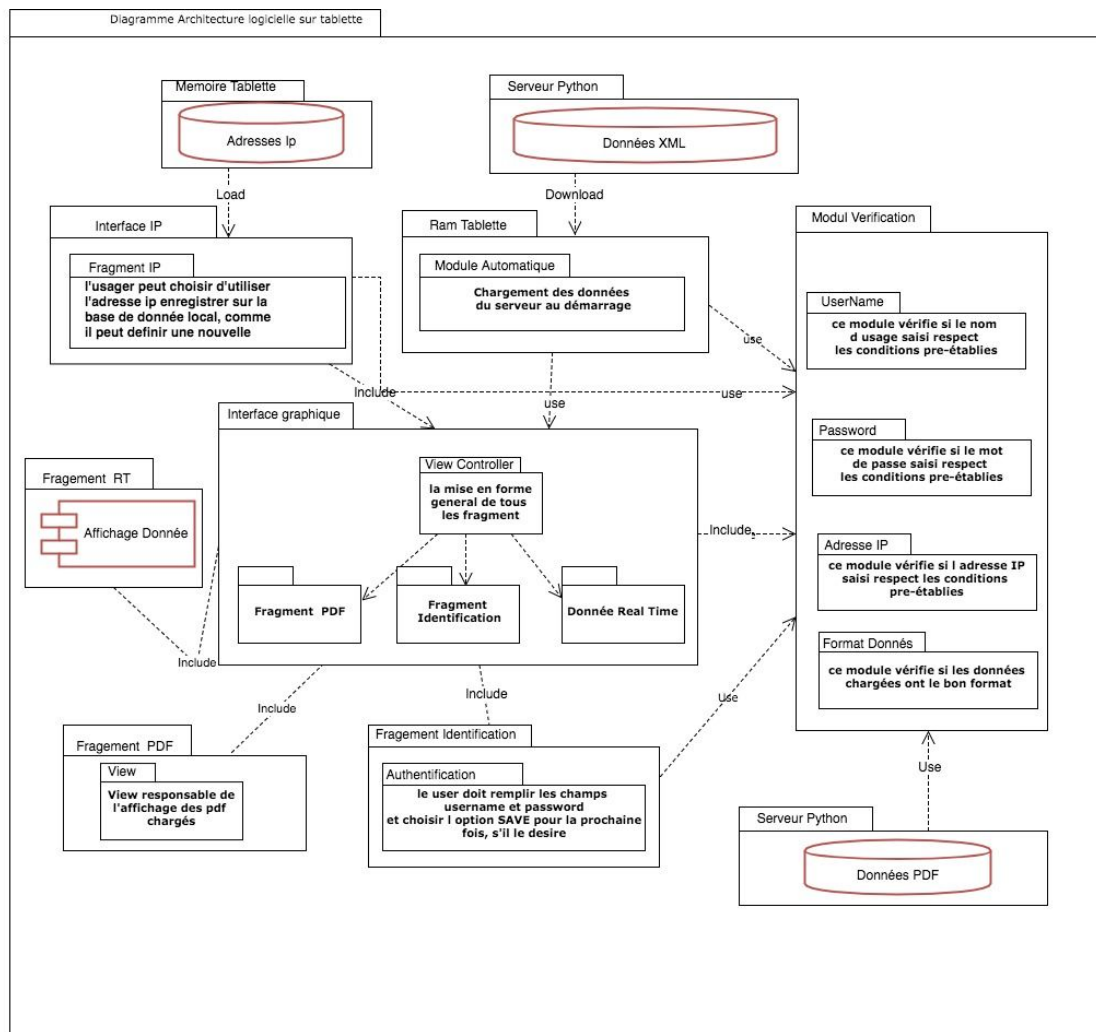


Figure 2.1: Diagramme architecture logicielle de la tablette.

Au démarrage, le fragment *Interface IP* est invoqué, celui-ci se charge de tout ce qui est en lien avec l'initialisation de l'adresse ip, il récupère la dernière adresse ip avec laquelle l'utilisateur du support (tablette) a pu se connecter. Après que l'utilisateur eut fait son choix, le fragment *RAM Tablette* prend le relais, il télécharge toutes les données XML nécessaires du serveur et les charge en mémoire vive (temporaire). Alors le fragment *Authentification* est invoqué. Celui-ci provoque aussi l'affichage du GUI. À partir de ce fragment, le contrôle

des champs à remplir est assuré par le module de vérification. Ce dernier contient toutes les règles relatives aux données saisies par l'utilisateur du système. Après une authentification réussie, le fragment *données en temps réel* prend le relais et il peut afficher toutes les données qu'il reçoit du serveur.

Enfin le module *contrôle d'affichage* roule en permanence en arrière plan pour détecter toute demande de changement d'environnement (mode jour/nuit).

4. Processus de gestion

4.1 Estimations des coûts du projet

Selons nos estimations de temps, de ressources humaines ainsi que de ressources matérielles, nos estimations de coûts¹ se résument à:

- Salaire des développeurs-analystes²: 44 850\$
- Salaire du coordonnateur de projet³: 10 295\$
- Coût de la tablette Android Nexus 9: 500\$
- Coût de 3 Zedboard⁴: 1 865\$
- Coût de 3 licences du logiciel Vivado 2017⁵: 13 563\$
- Autres coûts (transports, électricité, salles, imprévus etc.): environ 9 000\$

Le prix total hors-taxes demandé pour ce projet est donc de 80 000\$.

¹ Tout ces coûts sont en CAD.

² Pour 69 heures dans le projet par employé tel qu'estimé plus loin.

³ idem.

⁴ Coût canadien selon le site <https://www.xilinx.com/products/boards-and-kits/1-elhabt.html>

⁵ Coût canadien selon le site <https://www.xilinx.com/products/design-tools/vivado.html#buy>

4.2 Planification des tâches (Q2.2)

Voici le diagramme de Gantt de nos tâches. Celui-ci permet de bien visualiser dans le temps l'ordonnancement du projet. À droite de chaque barre se retrouve le pourcentage de la tâche réalisée. On observe bien que les échéances sont fonction des tâches plus prioritaires aux tâches moins prioritaires.

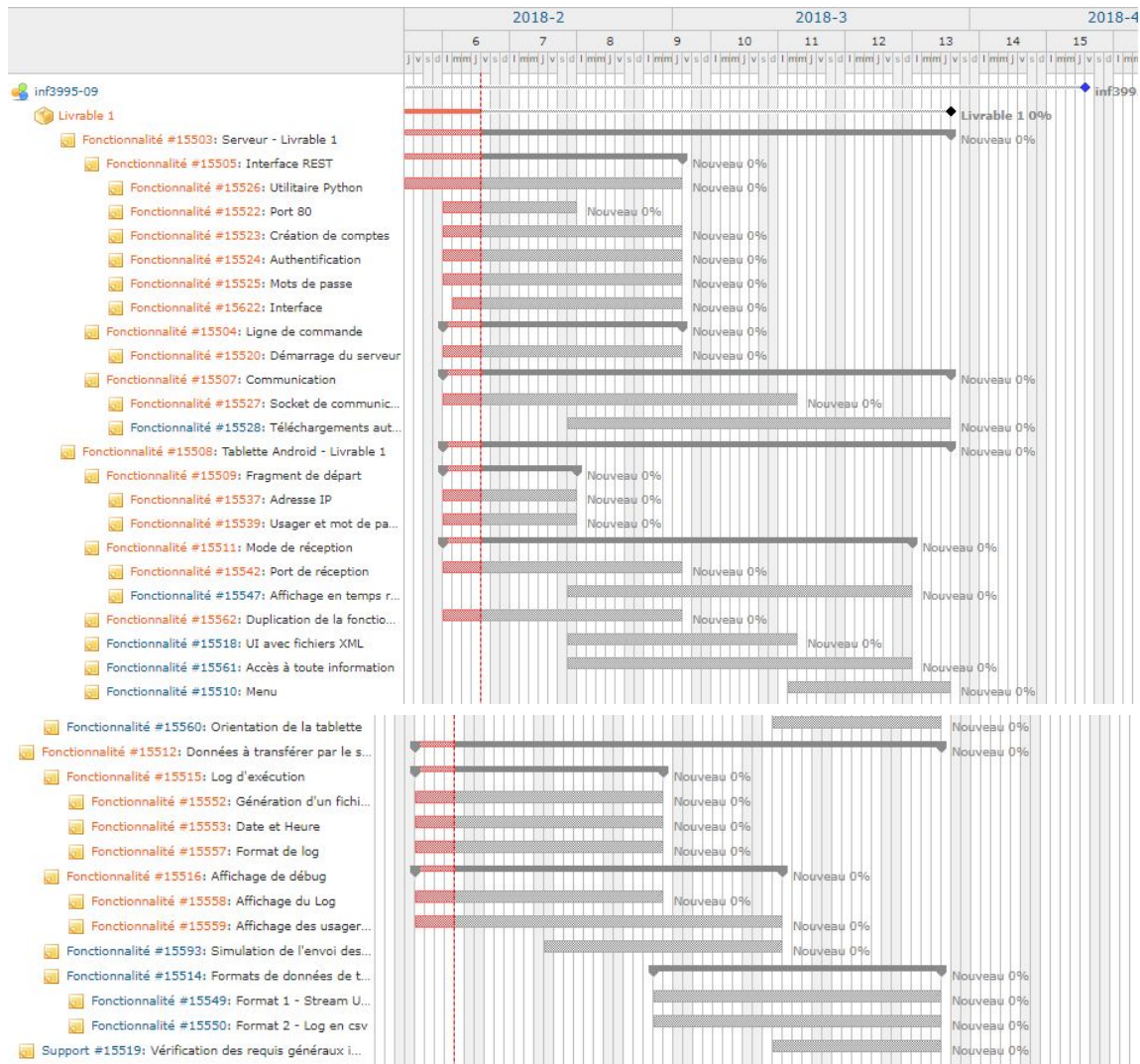


Figure 3: diagramme de Gantt

Un résumé de nos tâches du projet sur Redmine, nommé le "Roadmap" est présenté ci-dessous. Cela nous permet d'évaluer la progression globale du projet.



Figure 4: le Roadmap

Enfin, nous avons également assigné chacune de ces tâches à un responsable, afin de s'assurer qu'elles soient tous en développement. Exemple:

<input type="checkbox"/>	# ▾	Tracker	Statut	Priorité	Version cible	Sujet	Assigné à	% réalisé	Temps passé total	Echéance
<input type="checkbox"/>	15622	Fonctionnalité	Nouveau	Normal	Livable 1	Interface	Gabriel Demers		0.00	01/03/2018
<input type="checkbox"/>	15593	Fonctionnalité	Nouveau	Normal	Livable 1	Simulation de l'envoi des données	Gabriel Demers		0.00	13/03/2018
<input type="checkbox"/>	15562	Fonctionnalité	Nouveau	Normal	Livable 1	Duplication de la fonctionnalité 15515	Khalil Bennani		0.00	01/03/2018
<input type="checkbox"/>	15561	Fonctionnalité	Nouveau	Normal	Livable 1	Accès à toute information	Anthony Abboud		0.00	25/03/2018
<input type="checkbox"/>	15560	Fonctionnalité	Nouveau	Normal	Livable 1	Orientation de la tablette	Anthony Abboud		0.00	29/03/2018
<input type="checkbox"/>	15559	Fonctionnalité	Nouveau	Normal	Livable 1	Affichage des usagers connectés	Anthony Abboud		0.00	13/03/2018
<input type="checkbox"/>	15558	Fonctionnalité	Nouveau	Normal	Livable 1	Affichage du Log	Luc Courbariaux		0.00	01/03/2018
<input type="checkbox"/>	15557	Fonctionnalité	Nouveau	Normal	Livable 1	Format de log	Luc Courbariaux		0.00	01/03/2018
<input type="checkbox"/>	15553	Fonctionnalité	Nouveau	Normal	Livable 1	Date et Heure	Luc Courbariaux		0.00	01/03/2018
<input type="checkbox"/>	15552	Fonctionnalité	Nouveau	Normal	Livable 1	Génération d'un fichier	Luc Courbariaux		0.00	01/03/2018
<input type="checkbox"/>	15550	Fonctionnalité	Nouveau	Normal	Livable 1	Format 2 - Log en csv	Reph-Dauphin Mombrun		0.00	29/03/2018
<input type="checkbox"/>	15549	Fonctionnalité	Nouveau	Normal	Livable 1	Format 1 - Stream USB	Reph-Dauphin Mombrun		0.00	29/03/2018
<input type="checkbox"/>	15547	Fonctionnalité	Nouveau	Normal	Livable 1	Affichage en temps réel	Reph-Dauphin Mombrun		0.00	25/03/2018

Figure 5: Les demandes de redmine

4.3 Calendrier de projet (Q3.3)

<u>Jalons</u>	<u>Échéance prévue</u>
Fragment de départ de la tablette Android	Dimanche 18 février 2018
Interface REST du serveur	Jeudi 1 mars 2018
Logs d'exécution	
Ligne de commande du serveur	
Interface utilisateur via les fichiers XML	Mardi 13 mars 2018
Simulation de l'envoi des données	
Affichage des données en temps réel	Dimanche 25 mars 2018
Fonctionnalités mineures restantes	Mercredi 28 mars 2018
Remise de l'ensemble du livrable 1	Jeudi 29 mars 2018
GUI du serveur	Mercredi 11 avril 2018
Esthétisme de l'application Android	
Envoi de fichiers PDF (serveur)	
Réception de fichier PDF (application)	
Remise de l'ensemble du livrable 2	Jeudi 12 avril 2018

Tableau 1: Calendrier de projet

4.4 Ressources humaines du projet

Ce projet nécessite l'implication de cinq ou six personnes en moyenne. L'équipe est constituée de six personnes. Chacune d'elles dispose d'un éventail de qualifications et d'expériences.

Un des membres est assez versé du côté matériel et a déjà travaillé avec une carte FPGA et des outils Xilinx lors d'un stage. Un autre membre a réussi un stage en base de données. Donc, nous n'avons aucune crainte pour tout ce qui a trait aux bases de données. Nous avons aussi la chance d'avoir plusieurs membres qui connaissent le langage Python.

De plus, deux des membres suivent un cours d'informatique mobile en parallèle avec ce projet et un des membres avait déjà découvert le monde du développement Android depuis avant d'entrer à l'école, en raison d'un grand intérêt pour ce domaine.

Enfin, un membre a pu réaliser une des tâches les plus complexes qui soit avec brio, il s'agit de la physique, détection et réponse aux collisions lors d'un précédent projet de logiciel graphique interactif en 3D à lui seul. Par conséquent, l'équipe est tout à fait à la hauteur de ce projet et possède les compétences nécessaires à sa réalisation.

5. Suivi de projet et contrôle

5.1 Contrôle de la qualité

Tout au long de la réalisation de ce projet, il y aura un suivi continu par les membres de l'équipe par rapport à chaque nouvelle fonctionnalité ajoutée. Nos développeurs travaillent la plupart du temps par paire pour éviter l'émergence de bogues, assurer la lisibilité et le respect des requis.

À la fin du développement d'une fonctionnalité, le membre en charge en fait la démonstration aux membres de l'équipe. Cela permettra de vérifier que

chacun comprend et adhère à l'implémentation de la fonctionnalité et de classer la tâche comme achevée en toute confiance.

Nous désignerons également un membre de l'équipe qui se chargera de vérifier cette nouvelle fonctionnalité par rapport au requis technique.

5.2 Gestion de risque (Q2.6)

Comme tous les projets, nous devons faire face à plusieurs risques. Par exemple:

- L'absence d'un ou plusieurs membres de l'équipe pourrait être un problème et retarder les échéances. Pour régler ce problème, nous redistribuons des tâches et nous adoptons une réseau de communication en étoile.
- Des bogues pourraient surgir. Il faut analyser les parties faites avec plus attention et nous minimisons le temps alloué à une tâche pour ne pas changer les dates de livraison.
- Les problèmes d'intégration sont omniprésents dans les projets. Pour les réduire ou potentiellement les prévenir, nous travaillerons sur une branche *développeur*. Ainsi, la branche de développement sera fusionnée à la branche master seulement quand elle (develop) sera assez stable. Donc, la branche master est destinée aux remises de livrables. Enfin, une branche de *backup* sera utilisée en mesure de prévention du pire.
- Le matériel utilisé lors de ce projet peut présenter un risque car il peut tomber en panne, se briser ou des accidents peuvent survenir. Toutefois, en cas de problèmes avec des postes ou cartes FPGA, il suffit de le signaler aux responsables, sans mentionner que l'équipe dispose de tout un laboratoire dont les postes sont dotés de cartes FPGA. En ce qui concerne des problèmes avec la tablette Android, tels que perte, vol, etc. ces problèmes ne représentent guère un obstacle à la réalisation du projet

puisque la majorité de l'équipe dispose d'appareils mobiles Android et d'émulateur sous Android Studio.

5.3 Tests (Q4.4)

Les mesures prises pour l'assurance de la qualité (programmation par paire, validation en équipe de la complétion des tâches) contribue fortement à la prévention des bogues. Toutefois, des tests exhaustifs et de stress sont toujours nécessaires.

D'un côté, on trouve le sous-système application pour tablette Android (Client) pour lequel nous allons adopter des tests manuels pour le bon fonctionnement de l'interface de l'application. Par exemple, vérifier si la navigation se fait correctement, i.e. mène au fragment ou à l'activité approprié et des tests de performance afin de s'assurer que l'affichage se fait en temps réel, comme spécifié dans les requis.

D'un autre côté, le sous-système du logiciel serveur nécessitera des tests manuels pour le bon fonctionnement du GUI de l'application. Par exemple, on peut vérifier si le programme s'exécute comme attendu en fonction des paramètres (types de fusées, de terrains, etc.) entrés initialement via le GUI.

Du point de vue matériel, le Zedboard est quant à lui géré par le système d'exploitation linux (une distribution fourni par Xilinx) : on s'attend à ce que l'on puisse donc abstraire la plupart des détails quant au fonctionnement du système embarqué autre que la taille de la mémoire sur la puce.

Enfin, au niveau du système global, des tests d'intégration s'imposent car de multiples problèmes risquent de survenir si les systèmes développés séparément ne prévoyaient pas des interfaces de communication suffisamment isolées et satisfaisantes. Il ne faut pas oublier les tests de régression car certaines fonctionnalités risquent de ne plus passer des tests qu'elles avaient déjà passés au préalable lors d'une version antérieure.

5.4 Gestion de configuration

La gestion de version sera réalisée au moyen de git. C'est un outil très pratique, à l'heure du jour et de popularité croissante.

Pour structurer le projet, on subdivise son répertoire en *Livrable 1* et *Livrable 2*. À l'intérieur de chacun des dossiers de livrables se trouvent encore deux sous-dossiers, à savoir Client et Serveur.

Le premier sous-dossier, i.e. Client est destiné au projet Android et le dossier Serveur, à l'application serveur. À la racine du projet Android, nous avons placé un fichier *.gitignore* afin d'éviter les conflits causés par des fichiers que l'EDI Android Studio génère automatiquement et à chaque fois, de même que certains fichiers JSON qu'il modifie ne serait-ce que lors d'un simple lancement. L'architecture de base des projets Android est respectée, incluant les répertoires pour les tests Android et les tests en général et des paquetages Java.

Pour ce qui est du deuxième sous-dossier, i.e. Serveur, il y aura un dossier pour les tests, séparé du code et des fichiers de données également.

Tout le code (client et serveur) sera documenté de façon à ce que un outil tel que Doxygen puisse passer à son travers, récupérer les commentaires de documentation et générer la documentation en pdf ou format web (HTML). Enfin, dans le projet, on y trouvera des fichiers readme et de guides de configurations et d'installation.

6. Références (Q3.2)

Prix Vivado : <https://www.xilinx.com/products/design-tools/vivado.html#buy>

Prix Zedboard : <https://www.xilinx.com/products/boards-and-kits/1-elhbt.html>

