

## Comptes-Rendus des réunions de travail

Réunion n°1 :		
Lieu et Date	Université de Rouen, le 14/10/2016	
<b>Membres présents</b> <ul style="list-style-type: none"><li>Baudin Jordan</li><li>Caron Franck</li><li>Gille Nicolas</li><li>Leplumey Camille</li><li>Pommier Grégoire</li><li>Rabou Khalid (1h)</li></ul>	<b>Intervenants</b> <ul style="list-style-type: none"><li>Mr. Karim Abdellah-Godard</li></ul>	
Résumé par	Horaire de la réunion	Référence
Caron Franck	09h00 → 12h00	R001
Récapitulatif		
<p>Première réunion avec l'équipe du projet SmartLogger.</p> <p>Durant la première moitié de la séance, nous avons eu nos premières discussions sur la marche à suivre et sur l'organisation au sein de l'équipe :</p> <ul style="list-style-type: none"><li>Sélection du chef d'équipe : Franck.</li><li>Sélection de technologies de contact, pour améliorer les communications en interne (Slack, Discord).</li><li>Mise en commun des coordonnées des membres de l'équipe.</li><li>Création d'un dépôt GIT, associé au projet.</li></ul> <p>Sur la seconde moitié de la séance, nous avons démarré l'écriture de la STB. Nous avons étudié le sujet afin d'en ressortir les différents aspects, et effectué un brainstorming pour identifier le besoin client, les fonctions principales devant être réalisées par le produit final, etc.</p>		

## Réunion n°2 :

Lieu et Date	Université de Rouen, le 21/10/2016	
<b>Membres présents</b> <ul style="list-style-type: none"><li>• Caron Franck</li><li>• Gille Nicolas</li><li>• Leplumey Camille</li><li>• Pommier Grégoire</li><li>• Rabou Khalid</li></ul>	<b>Intervenants</b> <ul style="list-style-type: none"><li>• Mr. Jonathan Germond</li></ul>	
Résumé par	Horaire de la réunion	Référence
<TO ADD>	10h00 → 11h30	R002

## Récapitulatif

Cette réunion permet de réaliser la première rencontre entre l'équipe de projet et le client.

Nous avons démarré par des formalités concernant l'organisation établie (rôle de chacun, etc.), et les différents moyens mis en place, afin de faciliter la communication entre les acteurs du projet (notamment, via les applications Slack et Git).

Dès lors, nous avons commencé par demander une explication plus approfondie du travail à réaliser : son contexte d'exécution, la façon dont il sera utilisé, les critères à respecter... Nous avons ainsi évoqué divers sujets en rapport avec la définition du besoin, ce qui nous a permis par la suite, de pouvoir commencer la rédaction du document de Spécification Technique du Besoin :

- Tout d'abord, l'application finale sera utilisée par l'entreprise cliente en interne, afin de pouvoir surveiller l'activité de leurs autres applications. De ce fait, l'application sera amenée à gérer des flux de données de taille raisonnable (de l'ordre du Mo).
- Ces flux de données ne seront pas standardisés en une norme unique, il faudra donc s'adapter à différents formats d'entrée, et faciliter les futures adaptations.
- De même, les algorithmes d'apprentissage du système pourront être remplacés, au cours de la vie du système. Il faudra donc opter pour une programmation très modulaire dans le but de faciliter toute modification « à chaud ».
- Le logiciel devra conserver TOUTES les données dont il aura effectué le traitement, il faudra donc prévoir des solutions de stockage efficaces. Suite aux recommandations, nous nous orienterons vers l'emploi d'une base de données NoSQL.
- Ce dernier n'effectuera des alertes que si les analyses des données révèle une défaillance (équivalent à un Warning, ou une Erreur). L'ampleur de l'alerte dépendra du niveau de criticité de cette défaillance (Échelle à définir ultérieurement par nos soins).
- Le système fonctionnera sur un serveur Linux (de type Apache Mesos) de l'entreprise. Il devra être conçu de manière à pouvoir fonctionner sur de (très) longues périodes de fonctionnement (dans l'idéal, ce dernier ne devrait pas s'arrêter). Ses réponses seront à envoyer en temps-réel (ou presque, en notifiant de manière périodique toutes les n secondes).

Le client nous a également fait part des ses recommandations, à propos des technologies à utiliser :

- Pour l'interface utilisateur : Angular 2 (Réalisation), L.E.S.S (Affichage).
- Pour le mécanisme d'apprentissage du système : Bibliothèque Spark + Langage Scala
- Pour toute partie applicative : Utilisation du framework Spring Boot.

Du fait que ses technologies sont connues par l'entreprise, et qu'il s'agisse de technologies à la fois récentes et efficaces, nous allons certainement opter pour ces technologies, lors de la future phase de réalisation.

### Réunion n°3 :

Lieu et Date	Université de Rouen, le 04/11/2016	
<b>Membres présents</b> <ul style="list-style-type: none"> <li>Baudin Jordan</li> <li>Caron Franck</li> <li>Gille Nicolas</li> <li>Leplumey Camille</li> <li>Pommier Grégoire</li> </ul>	<b>Intervenants</b> <ul style="list-style-type: none"> <li>Mr. Karim Abdellah-Godard</li> </ul>	
Résumé par	Horaire de la réunion	Référence
Caron Franck	9h00 → 12h00	R003

### Récapitulatif

Deuxième réunion dans le cadre du TP de Gestion de Projets. Afin de préparer la séance, nous avons rédigé la première version de notre document de Spécification Technique des Besoins.

Ainsi, Mr. Abdellah-Godard nous a fait part de ses conseils et critiques à propos du document, dans le but de pouvoir faciliter son amélioration future :

- Améliorer la définition des exigences en vérifiant que chacune est mesurable.
- Décrire de manière encore plus précise les cas d'utilisation.

Nous avons ensuite démarré la rédaction de notre Document d'Architecture Logicielle.

A cet effet, nous avons commencé par schématiser le fonctionnement du logiciel selon les cas d'utilisation définies au préalable par la STB.

Nous avons alors dégagé des cas systèmes alternatifs (e.g : gestion des données invalides) qui augmentent la documentation pré-établie.

Dès lors, nous avons compartimenté les différentes fonctionnalités au sein de modules afin de respecter au maximum les différents critères de modularité de notre projet :

- Input** : Module d'entrée de notre applicatif. Il se charge de la récupération des données envoyés à notre système, les adapte en un format unique et vérifie leur validité.
- Output** : Module de sortie. Il se charge d'effectuer les opérations d'alerte ou de redirection nécessaire, à l'issue du traitement d'un jeu de données.
- Machine Learning** : Module comprenant la partie analytique de notre système. Ce module prendra en entrée des données valide.
- Database** : Module de gestion de la base de données. Contient l'intégralité des données ayant été manipulées par le systèmes (données traitées, données non traitées et données invalides).

Ces modules correspondent à un schéma d'architecture de premier niveau, définissant les composants majeurs de notre système.

Le travail futur à réaliser sera donc de préciser le contenu de ces modules. Puis de définir les principales fonctions de ces dernières. Nous pourrons alors finaliser la rédaction du DAL.