



COMPTE RENDU DE REUNION N° 07



OBJET DE LA REUNION **Point sur l'avancement du projet**

DATE DE LA REUNION **10/12/2020**

REDACTEUR **Alice Gontard**

PROCHAINE REUNION **à définir**

Mel : gontard.e1902833@etud.univ-ubs.fr



ENTITE / SERVICE	PARTICIPANTS	TELEPHONE / TELECOPIE / E-MAIL	PRES.	ABS.	DIFF.	CONV.
	Matthieu Le Lain	lelain@univ-ubs.fr	✓			
	Christophe Garcia	garcia.e1901634@etud.univ-ubs.fr	✓			
	Sullivan Leboeuf	leboeuf.e1903169@etud.univ-ubs.fr	✓			
	Lucy Gastebois	gastebois.e1900989@etud.univ-ubs.fr	✓			
	Alice Gontard	gontard.e1902833@etud.univ-ubs.fr	✓			

DIFFUSION COMPLEMENTAIRE	DIFF.
--------------------------	-------

Table des matières

1. Abord de l'aspect sécurité	2
2. Matrice de risque et tableau de bord	2
3. Livraison et autres sujets	3
4. Prochaine réunion	3



COMPTE RENDU DE REUNION

N° 07

DATE	CONCERNE
D : DE- BUT F : FIN	A : POUR ACTION E : EMETTEUR I : POUR INFORMATION

1. Abord de l'aspect sécurité

Pour commencer, nous nous sommes demandé si nous devons mettre sur git les divers identifiants et mots de passe que nous avons. La réponse a été que non, **on ne met pas les identifiants et mots de passe sur git car c'est facilement récupérable par des tiers.**

Cependant, il est possible d'avoir un fichier de configuration dans lequel on retrouve ses identifiants et mots de passe.

Ce sera l'API qui viendra lire ce fichier de configuration.

Enfin, nous avons abordé la sécurité des mots de passe.

Dans la table connexion, on pourrait avoir une fonction qui chiffre les mots de passe et qui fait des vérifications avec un mot de passe chiffré.

Pour cela on retrouve différents niveaux de sécurité :

- **Niveau de sécurité de base (+)** : on passe le mot de passe dans une fonction de hachage (avec un md5). Au moment de la connexion l'utilisateur saisit son mot de passe, puis on le hache et on le compare aux haches stockées.
- **Niveau de sécurité plus poussé (++)** : on utilise une clé de chiffrement secrète --> il faut chiffrer le mot de passe avec la clé
- **Niveau de sécurité encore plus poussé (+++)** : on rajoute par-dessus le mot de passe chiffré un txt aléatoire par-dessus le code. Nous pouvons rechercher sur internet la procédure de chiffrement et salage.

Pour le projet Rosetta, la sécurité avec la fonction de hachage devrait largement suffire.

Personne ne doit pouvoir récupérer le mot de passe.

Si jamais le mot de passe est perdu, il faudrait que ça renvoie à l'utilisateur une page où il puisse taper son email. Puis que ça vérifie dans base de données le mail et enfin soit on génère un mot de passe au hasard ou soit on envoie sur une url qui permet à l'utilisateur de saisir un nouveau mot de passe.

La fonction de hachage se fait dans l'API et elle n'est pas une urgence de premier sprint.

Rajouter une sécurité sera un plus à présenter lors de la soutenance et pour la livraison du projet au client.

Nous sommes le seul groupe qui a posé cette question de sécurité.

2. Matrice de risque et tableau de bord

Notre matrice de risque est très bien réalisée, c'est exactement ce qui est attendu.

Nous avons bien listé les risques, nous avons eu de très bonnes idées, et nous avons bien utilisé l'outils.

Le nombre de risques est limité car le projet est cadré d'un point de vue pédagogique par l'IUT.

Par exemple : Il n'y a pas de grand risque que l'entreprise de Monsieur HOURDIER ferme.

Ensuite, concernant l'aspect juridique du projet ne doit pas être évoqué dans la matrice de risque.

De plus, nous ne sommes pas obligés de mettre les tableaux de bord en PDF sur gitlab.



COMPTE RENDU DE REUNION

N° 07

DATE	CONCERNE
D : DE- BUT F : FIN	A : POUR ACTION E : EMETTEUR I : POUR INFORMATION

3. Livraison et autres sujets

Concernant la documentation SQL, il n'en existe pas d'officielle ou se sont des librairies annexes donc **nous pouvons juste faire des commentaires.**

Il existe une fonction sur git, le git ignore dont les fichiers mit dedans seront ignoré durant les commandes pull, push, ...

Par exemple on y retrouvera le dossier Node modules.

Pour le livrable au client, on peut faire un livrable de luxe qui sera en zip.

→ Il faudra faire un **export du git, écrire un readMe de l'installation et nous pouvons créer une instance en ligne avec une URL** pour que le client puisse tester ce que nous avons développé.

→ Nous pouvons aussi réaliser **une vidéo qui présentera ce que nous avons développé pour le moment**. Nous pourrons montrer comment on ajoute, on supprime, etc.

Cette solution est très bien pour un client qui n'a pas le temps, on lui prémâche le travail.

4. Prochaine réunion

Avant la réunion :

Prochain rendez-vous

Être avancé dans le code.

Pour le moment, nous n'avons pas de prochain rendez-vous de prévu mais si on a besoin il n'a aucun problème.