

Rapport de stage

Développement de fonctionnalités front-office et back-office d'un site internet de gestion de chantiers connectés

Tutrice : Hana ALOUAOUI
Exemplaire Personnel - **V 1.5**
Etudiant : Fallou SEYE
Rendu le 21 Juin 2018

Gestionnaire des versions

Version	Nom	Date	Commentaire
V 1.1	Fallou SEYE	18/06/2018	Mise en page, gestionnaire des versions, Entête pied et page
V 1.2	Fallou SEYE	19/06/2018	Remerciements, introductions, fiche techniques
V 1.3	Fallou Seye	20/06/2018	Partie 1, l'entreprise. Partie 2 Travail réalisé
V1.4	Fallou SEYE	22/06/2018	Partie 2 : Travail réalisé
V 1.5	Fallou SEYE	23/06/2018	Partie 3 : Conclusion + Annexes

Remerciements

Je souhaite remercier avant tout Benjamin BELLANTONIO, il n'est jamais facile de trouver un stage et c'est le seul parmi toutes les demandes que j'ai faites, à m'avoir fait confiance, à m'avoir donné cette chance de pouvoir exprimer mes qualités en développement. Son accueil, sa gentillesse et sa patience m'ont aidé à gagner en expérience, aussi court soit le stage.

En effet, débiter une nouvelle expérience avec un accueil plein de sourires, de la communication, des petites rigolades est beaucoup plus facile qu'autre chose, cela épargne une pression inutile.

Enfin, je ne terminerais pas sans remercier ma tutrice Hana AIOUAOUI ainsi que tous les professeurs et personnels du département Informatique de l'IUT qui m'ont donné cette chance et qui m'ont fait progresser durant ces trois années dans les domaines aussi bien techniques que secondaires (communication, droit...), ce qui a été essentiel pour réaliser ce stage, et qui le sera pour le reste de ma vie.

Fiche technique

<ul style="list-style-type: none"> • Etudiant : Fallou SEYE • 2018 • Raison sociale : Benjamin BELLANTONIO • Maître de stage : Benjamin BELLANTONIO • Enseignant tuteur : Hana ALOUAOUI
<ul style="list-style-type: none"> • Missions : établir un formulaire de changement de mot de passe ainsi que de fuseau horaire, la langue et l'heure. Concevoir et créer un système de notification par la suite en temps réel. Diffuser des messages sur le tableau de bord (front et back-end) en fonction de l'entreprise ou du groupe d'utilisateur auquel on fait partie. Concevoir et réaliser un système de discussion instantané en temps réel.
<ul style="list-style-type: none"> • Plateforme informatique : Unité centrale : Zalman, Ecran : Acer • Système d'exploitation : KDE Linux • IDE : PHPStorm, langages, librairies : PHP, JQuery, Node.js et Symfony 3.4, Twig, RabbitMQ
<ol style="list-style-type: none"> 1. Web 2. Développement 3. PHP 4. Symfony 5. Front-end 6. Back-end 7. Notifications 8. Chat 9. Temps réel

Sommaire

Gestionnaire des versions.....	2
Remerciements.....	3
Fiche technique.....	4
Sommaire.....	5
Introduction (Français).....	6
Introduction (Anglais).....	6
1.1/ L'entreprise, son histoire et sa culture :.....	7
1.2/ L'activité de l'entreprise :	7
1.3/ Ressources humaines et techniques :	7
1.4/ Le service	8
1.5/ Photographies des produits	8
2/ Travail réalisé	10
2.1/ Tutoriel	10
2.2/ Début dans le projet, mes premiers formulaires	10
2.3/ Notifications.....	11
Conception et implémentation	11
Ajout du temps réel	12
Notification sur les rapports de sécurité	13
2.4/ News dans le tableau de bord	13
2.5/ Mise à jour du statut du chantier	14
2.6/ Patterns pour la recherche de date	14
2.7/ Tchat en temps réel	15
3/ Bilan	19
3.1 L'expérience professionnelle en entreprise	19
3.2/ La formation.....	20
3.3/ Conclusion.....	21

Introduction (Français)

En commençant ce stage, j'ai rejoint Benjamin dans un gros projet qu'il a commencé en février dernier, qui doit normalement finir en août 2018 et qui consiste à réaliser un site internet nommé « APS (Angel Protect System) ». C'est logiciel de gestion de chantier et qui permet aux entreprises réalisant des chantiers de pouvoir suivre l'avancement, les informations et les problèmes liés grâce à des appareils photos connectés. Le client final est l'entreprise pétrolière TOTAL mais l'intermédiaire est DECADI, qui est son associé.

Arrivant dans ce contexte, j'ai commencé tout doucement car je ne maîtrisais pas bien les outils, puis de semaine en semaine, je me suis un peu plus impliqué dans le projet car je réalisais des tâches plus importantes et qui ont été validés et incluses dans le projet.

La mission n'était pas la plus urgente donc elle m'a occupé durant la totalité du stage et j'ai réussi à l'achever à temps, les explications ainsi que la pédagogie de Benjamin m'ayant beaucoup aidées car il y avait beaucoup de choses à apprendre et à faire, le projet avait bien avancé et donc j'avais raté beaucoup de choses.

L'environnement de travail me correspond exactement car nous étions que deux (j'aime le travail en petit nombre), on a réussi à communiquer quand il le fallait, à coder ensemble sans problème ni conflits et sans casser la structure des programmes.

Introduction (Anglais)

Starting this internship, I joined Benjamin in a big project that he started in the last February and which should be done in August. « APS (Angel Protect System) » is a site management software that allows a real time trace for companies which have sites and connected it to the website. We can see all problems, informations and progressions reported to the site thanks to cameras. The final client is the oil company Total but we work for DECADI, its partner.

By coming in this context, I had to start slowly because I did not control all tools that I should have to, but weeks to weeks, I got myself more involved in the project because I made more important tasks which were validated and included to the project.

The mission was not so hurry so I took my time, I got busy during the whole internship and I finished in time, Benjamin helped me a lot, he knew how to make me understand because the project already started so I missed a lot of things.

The professional environment was very good for me because we were only two (I prefer to work with few people) we talked when we had to and coded together without problems or major conflicts.

1.1/ L'entreprise, son histoire et sa culture :

Benjamin BELLANTONIO est développeur Freelance depuis 2014. Il a réalisé de nombreux projets dont APS est le plus important.

Il a commencé le développement par une simple formation basique (HTML, CSS, PHP, MySQL) qui lui a permis d'enchaîner sur un stage qui a finalement duré sept mois. Il a ensuite été embauché et a travaillé dans deux entreprises pendant trois ans jusqu'en Janvier 2014, où il décide de continuer seul avec un ami graphiste. C'est ainsi qu'il est en individuel jusqu'à maintenant. Il a donc un bon nombre de clients et d'associés avec qui il collabore.

Benjamin est un développeur très sérieux qui se consacre à fond dans ce qu'il fait. Il prône une « hygiène du développement » c'est-à-dire produire un code de qualité qui sera totalement fonctionnel, validé par le client et qui permettra une maintenance facile et rapide. C'est un principe fondamental pour lui car cela inclut la satisfaction du client mais aussi individuelle car cela donne un sentiment de fierté du travail accompli.

La satisfaction du client est aussi une des valeurs qu'il veut respecter au maximum car rendre la confiance qu'on lui a donnée est toujours quelque chose de positif et d'encourageant. Cela aide à se rassurer quant à son honnêteté, car on a réalisé le travail que l'on devait faire.

Le fait de rester soi-même, de faire normalement son travail permet d'être exempt de tout reproche et d'avoir la conscience tranquille. Il permet aussi de se faire un nom dans le secteur informatique qui est en constante recherche de développeurs ayant ce profil.

1.2/ L'activité de l'entreprise :

L'activité de Benjamin est essentiellement du développement web. Il s'occupe de la conception et de la réalisation du projet (front et back-office). Le design est délégué à un graphiste.

1.3/ Ressources humaines et techniques :

Etant Freelancer, il est seul, mais il collabore beaucoup avec des associés tel que le designer, d'autres associés du projet ou le responsable informatique de l'entreprise cliente. Il arrive aussi qu'il prenne un stagiaire pour l'aider dans des tâches qui ne sont pas les plus importantes.

Le travail est effectué sous linux et avec l'IDE PHPStorm.

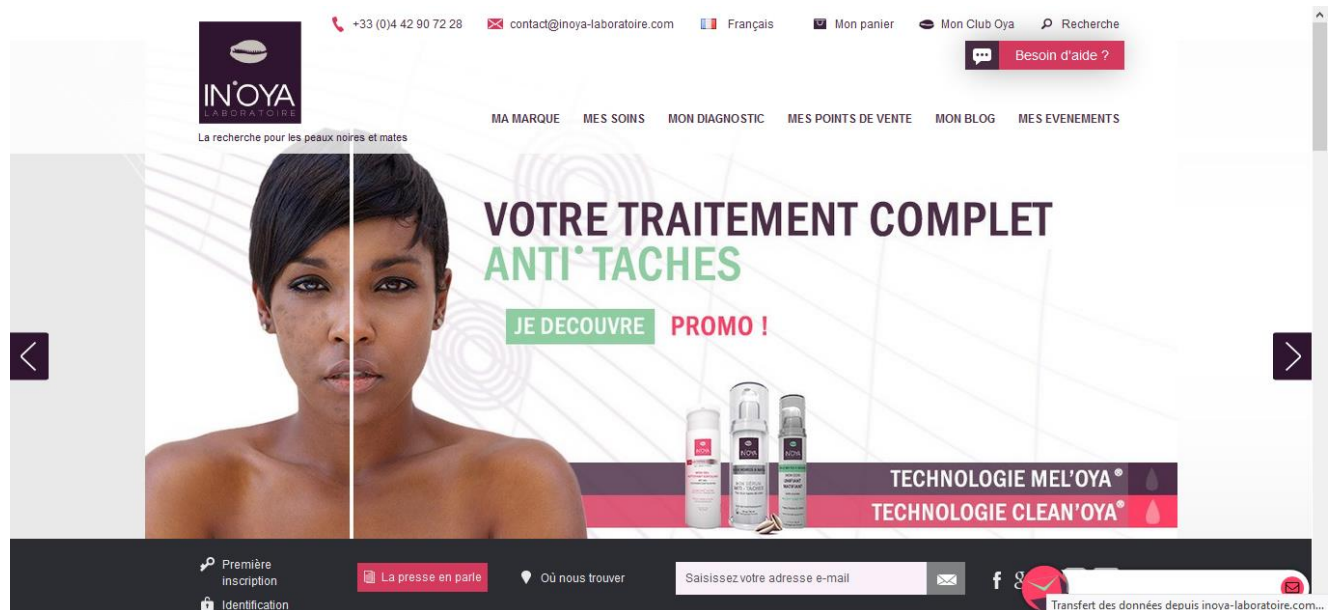
Les projets sont développés avec le langage de programmation PHP et en utilisant le Framework Symfony avec tout ce qu'il englobe (doctrine, Twig...). Le langage Javascript est aussi utilisé (bibliothèque JQuery) et les incontournables HTML, CSS, MySQL.

Certaines librairies, API ou protocoles sont aussi présents de temps en temps tel que, GuzzleHttp, RabbitMQ, Node.js qui sont utilisés pour des traitements non courants.

1.4/ Le service

Etant associé en binôme avec Benjamin, je travaillais sur le projet courant avec lui. J'étais donc directement impliqué sur ce dernier, il fallait donc faire très attention à ce que je fais, pour ne pas créer de confusions, tout tester avec d'envoyer.

1.5/ Photographies des produits



IN'OYA

JE DONNE MON AVIS SOLUTIONS RÉFÉRENCES CLIENTS SECTEURS D'ACTIVITÉS NEWS GUIDE LOCAL CONTACT RESERVEZ VOTRE DEMO

SOLUTIONS FEEDBACK CLIENT AVEC APPLICATIONS MOBILES

SOLUTION DE RECUEIL D'AVIS POUR LES ENTREPRISES

Je Donne Mon Avis s'inscrit totalement dans le par... analyses et une communication des avis qui deviennent...

[Lire plus](#)

Utilisation des cookies
En naviguant sur notre site vous acceptez l'installation et l'utilisation des cookies sur votre ordinateur. [En savoir plus](#)

X

Je donne mon avis

Vous êtes connecté en tant qu'administrateur

aps Rechercher dans les PVS, CS, Chantiers, Utilisateurs... Fallou Seye

Carte **Chantier**

Appareils photo
Planning
Plan de formation
Timelapse
Documents sécurisés
Messagerie
Statistiques

PVS

93.5% Situation à risque
6.5% Bonnes Pratiques

NB RÈGLES TRANSGRESSÉES PAR PVS (1)

PVS sur la durée du chantier

TOP 5 RÈGLES TRANSGRESSÉES

efgdxgdxg

Chantier - Chantier complet
Aix-en-Provence, France
0102030104

Total Panorama 4

MÉTÉO

11/05/2018 02:29 PM
Situation à risque
xy xwv

En ce moment
87° / 30°C

Jeu. 21/06/2018 Vent 16km/h Précipitations 0%

Information
Attention chaleur extrême

DERNIERS CONTACTS

A
Admin Admin
admin
Abeba Adis
ok

Angel Protect System

2/ Travail réalisé

2.1/ Tutoriel

Maîtriser les fondamentaux de Symfony d'ici la fin de semaine pour intégrer le projet la semaine suivante.

La première semaine était une semaine de découverte, je devais prendre mes repères par rapport aux outils informatiques à utiliser. Je connaissais bien le langage PHP ainsi que Javascript et autres mais les utiliser et les adapter à un contexte précis est un tout autre travail.

J'ai commencé par créer un petit projet tutoriel pour appréhender Symfony, les débuts étaient difficiles mais une fois les concepts compris, c'est plus facile.

Je travaillais en autonomie, j'ai du tout rechercher par moi-même, essentiellement en utilisant la documentation officielle et si je ne comprenais pas, j'allais voir d'autres forums.

Cependant, si le problème persistait, Benjamin venait m'aider à comprendre ou à faire debugger le code que je faisais.

J'ai dû apprendre comment faire un contrôleur, ses différentes routes, comment une action avec toutes les méthodes qui vont avec, obtenir l'utilisateur de session, avoir accès aux services, récupérer les paramètres d'URL, créer, traiter un formulaire, comment envoyer un email, retourner des Template...comment interagir avec la base de données grâce à classes qui ne faisaient que persister les données (Managers) et des classes qui s'occupaient seulement des requêtes (Repository).

J'ai réussi tout cela pendant la semaine sans trop de problèmes sauf au début. J'ai compris tous les concepts, et, même si je venais de commencer, j'avais assez pour continuer et apprendre sur le tas.

2.2/ Début dans le projet, mes premiers formulaires

Durant cette semaine, j'ai commencé par deux formulaires :

- Un formulaire de changement de mot de passe
- Un formulaire pour changer ses paramètres tel que la langue, l'heure, l'abonnement aux emails, cela m'a pris un jour

Les vues des formulaires étaient déjà créées mais il ne manquait plus qu'à implémenter le traitement pour ces derniers.

Le contrôleur était déjà fait, le seul devoir qui me revenait était d'écrire la fonction qui traite les deux actions.

Comme les deux formulaires sont sur la même page, quand on envoie un, les deux sont envoyés, je devais donc traiter les deux formulaires sur la même fonction du contrôleur.

Pour chaque formulaire, je devais les récupérer, vérifier lequel a été validé, puis faire le traitement, pour ne toucher que celui qu'on a envoyé.

Pour le changement de langue, si elle a été changée, le formulaire devait changer l'URL pour mettre la nouvelle URL et changer la traduction de la page. On devait renvoyer la nouvelle URL avec la langue qu'il fallait.

J'ai eu des difficultés à faire ça car je n'avais pas vu cette facette de Symfony pendant mon petit tutoriel, mais j'ai regardé la documentation et réussi à trouver ce qu'il fallait.

Toutefois, j'ai pris du temps à faire fonctionner cela mais Benjamin m'a aidé et ça a bien marché.

2.3/ Notifications

L'objectif était de réaliser un système de notification en temps réel qui permet de tenir informé l'utilisateur de tout ce qui se passe dans le chantier et/ou dans le site internet.

Conception et implémentation

La conception s'est faite sur une demi-journée. J'ai utilisé un modèle de conception Observateur.

Lorsqu'une action se passe, la notification est automatique et non manuelle, juste en prenant connaissance des utilisateurs concernés et du type de notification, il est aussi possible d'envoyer des données supplémentaires en option. Cela a été intégré au projet et je devrais continuer à étendre cela la semaine suivante.

Premièrement, j'ai fait tout le travail pour lister créer, supprimer une notification côté administrateur. Le traitement pour passer en vu/ pas vu se fait en ajax (asynchrone) donc dans un contrôleur ajax.

L'essentiel du travail était le contrôleur, le côté dynamique en ajax et appeler le service qui envoie les notifications au bon moment.

Le système de notification a été utilisé une première fois pour un type de notification de connexion FTP refusée. Toutes les dix minutes, on teste les galeries qui récupèrent les photos en tentant une connexion FTP, si celle-ci échoue, elle est désactivée (galerie) et une notification est envoyée (on a le choix, email, sms ou les deux).

J'ai mis cet algorithme dans une commande Symfony, que j'ai elle-même mise dans un cron qui est exécuté toutes les 10 minutes. La fonctionnalité a été ajoutée au projet.

Pour alléger tout ce code, j'ai utilisé RabbitMQ, un protocole permettant l'échange de données entre un consommateur et un producteur, le tout suivant une queue de message.

Quand le producteur publie un message, celui-ci entre dans la queue (FIFO) et le consommateur la récupère.

La commande se permet juste de publier l'identifiant des galeries qui sont encore actives et le consommateur récupère, fait le teste, désactive si besoin et envoie la notification.

Ensuite, je me suis penché sur la vue utilisateur, j'ai fait une petite vue cliquable du genre Facebook pour afficher les X dernières notifications. J'ai implémenté un composant Twig (fonction qui renvoie une vue HTML en prenant des paramètres) en lui donnant l'identifiant de l'utilisateur et le nombre de notifications à retourner.

Cependant, il ne fallait pas oublier la partie sécurité, J'ai du tester avant chaque traitement, si l'utilisateur était connecté, s'il était administrateur.

Ajout du temps réel

Pour pouvoir mettre tout cela en temps réel, un petit tutoriel s'imposait pour connaître Node.js et socket.io. Socket.io est une librairie permettant les web sockets et donc la réception de données sans rechargement ni de requêtes pour en récupérer.

J'ai écrit le serveur Node.js qui réceptionne l'identifiant de la notification, l'envoi à l'utilisateur concerné s'il est connecté. Je suis resté bloqué une demi-journée à faire marcher la requête http avec Guzzle alors que l'erreur était simple : J'envoyais la requête avec la méthode POST alors que le serveur que j'ai écrit attendait des requêtes GET.

J'ai aussi fait marcher la déconnexion de la socket.

Lorsque l'utilisateur n'était pas connecté, mais qu'il arrive sur le site, il y a un problème car on envoie l'identifiant de l'utilisateur dans la socket (d'envoi de notification) donc je devais vérifier que l'utilisateur est bien connecté, pour cela, quand il se connecte, on garde sa socket.

Voici le schéma :

- Quand une notification est envoyée, on envoie au serveur temps réel l'identifiant de l'utilisateur en question et l'identifiant de la notification
- Le serveur temps réel vérifie qu'il est connecté si oui, il lui envoie un message.
- Le client reçoit le message du serveur temps réel avec les deux identifiants et grâce à cela peut récupérer l'objet en question en AJAX.

J'ai fait les deux premiers points je devais ensuite faire la requête en ajax qui va récupérer la notification, en fait, la requête renvoie directement la vue avec la notification que l'on ajoute en JQuery dans la partie HTML qui garde les notifications tout en prenant soin de détacher la moins récente.

Difficulté : quand une notification apparait, le bouton lu/non lu ne marche pas et marche que lorsqu'on recharge. J'ai réglé ce problème, j'ai remplacé l'évènement sur le document en le mettant sur body.

Notification sur les rapports de sécurité

Les rapports de sécurité permettent l'analyse des photos prises sur les chantiers (analyser une photo, constater une erreur ou une bonne pratique). Il doit pouvoir envoyer des notifications aux administrateurs lorsque le rapport est établi.

J'ai créé la notification. On a d'ailleurs modifié l'entité pour mettre un lien si besoin, en l'occurrence oui car le rapport est un fichier PDF. S'il y a un lien, on l'affiche à la place du contenu.

On pouvait joindre des utilisateurs externes au site dans le rapport de sécurité. Il était évident qu'ils ne sont pas dans la base donc on devait créer des utilisateurs temporaires pour pouvoir les affecter aux envois d'emails.

J'avais un autre problème, je ne recevais plus d'emails, je ne savais pas pourquoi...j'avais en fait juste décoché le bouton « Recevoir les emails ».

Finalement, tout a bien marché.

2.4/ News dans le tableau de bord

Je devais travailler sur la diffusion de messages informatifs dans le tableau de bord utilisateur, des chantiers et des groupes. Une news a une date de début, de fin, elle est aussi reliée à un chantier et un groupe d'utilisateurs.

L'administrateur crée l'information selon le type souhaité : Info, Warning, Danger. Comme on en affiche une à la fois, il faut trier selon la date, afficher la plus ancienne et qui n'est pas fini. Cela passé, on doit prendre en compte le chantier et le groupe, s'il y a un chantier passé en paramètre, on ne ressort que les news reliés, pareil pour les groupes, mais s'il y a un chantier, on ne tient pas compte du groupe.

Toutefois, j'ai dû changer, nettoyer mon code avec Benjamin car certaines choses n'allaient pas :

- J'ai copié/collé un autre formulaire sans bien changer ce qu'il fallait.
- Une erreur de jointure dans l'entité.
- Nettoyage du contrôleur, Ajout d'un formulaire pour l'édition création, suppression et lister avec filtres.

J'ai par la suite modifié la fonction de tri : car quand il y a un chantier en paramètre mais qu'il n'est associé à aucune news, elle ne renvoie rien, alors que par défaut, il faut que l'on renvoie les news triés. J'ai ajouté une condition avant d'appeler la fonction, s'il y a un chantier est qu'il est associé à au moins une news, on fait le traitement pour chantier, sinon on fait le traitement pour les groupe, sinon rien.

En ce qui concerne les groupes, je vais chercher les groupes ou l'utilisateur est membre, les passer dans la requête.

On avait une erreur lors de la suppression, un comportement qui n'était pas prévu, on a ajouté une condition pour éviter ce cas-là.

2.5/ Mise à jour du statut du chantier

Les chantiers ont un statut : « Processing », « Finished », « Waiting », « None ».

Il y avait plusieurs choses à faire :

- Lister les chantiers en cours, s'il s'est passé 15 jours depuis la date de fin, mettre le statut en « finished »
- Lister tous les chantiers en attente ou sans statut, si la date de début est passée, on passe le statut en « Process »
- Ajouter tout cela à un cron.

C'est très court mais j'ai pris du temps à réaliser, travailler avec les dates (pas assez de fonctions, comment savoir l'intervalle...) est pénible.

2.6/ Patterns pour la recherche de date

Afin de pouvoir dater les photos, la date est directement dans le nom du fichier. On utilise alors un pattern (fait maison) pour pouvoir le retrouver dans le nom du fichier.

L'administrateur doit pouvoir ajouter lui-même des patterns s'il décide comment nommer les noms de fichiers photos. J'ai donc créé une entité

La fonction qui détecte la date est déjà faite par Benjamin.

J'ai fait les fonctions de traitement (contrôleur) avec les vues associées pour lister, ajouter, modifier ou supprimer un pattern.

J'ai aussi remplacé les champs pattern (texte) par une liste déroulante qui affiche tous les patterns dans la base.

Dans le formulaire de création ou d'édition, j'ai fait une fonction qui, en ajax, appelle la fonction qui renvoie la date (celle faite par Benjamin), juste pour tester avant d'envoyer le formulaire. Si le teste est valide, on autorise l'ajout. Il y a une petite boîte de dialogue qui explique facilement le fonctionnement du pattern.

2.7/ Tchat en temps réel

L'objectif est de réaliser un tchat qui ressemble à celui de Facebook pour permettre la communication aux utilisateurs. J'ai commencé en faisant un chat côté administrateur, avec le contrôleur qui peut lister, créer, modifier supprimer une discussion. Cela à eu l'avantage de m'aider par la suite.

Dans un premier temps, j'ai fait la conception avec seulement deux entités Discussion et Message. Ensuite, j'ai compris qu'à long terme, je devais anticiper plusieurs choses. J'ai donc rajouté une entité UserMessage qui est le détail d'un message pour un utilisateur.

Un message n'est qu'une seule entité, mais pour chaque utilisateur il y aura le détail de ce message et cet utilisateur : a-t-il vu le message, quand ? S'il veut supprimer le message, il supprime cette entité car avant d'afficher un message, on regarde s'il y a le détail pour cet utilisateur.

J'ai mis à jour la date, J'ai fait le contrôleur pour lister, ajouter, continuer ou supprimer des utilisateurs. J'ai rencontré un problème sur la suppression en cascade que j'ai corrigé

Par la suite, j'ai commencé la partie utilisateur, avec une fenêtre qui s'ouvre sur le côté, affichant les discussions et liste des contacts.

Pendant une journée, j'ai juste réussi à changer la disposition, pour afficher en premier toutes les discussions et sur le bouton +, la liste des contacts. Cela utilise une librairie que je ne connais pas donc c'était difficile à manier.

Etape par étape, j'ai tout d'abord affiché le message à gauche ou à droite en fonction de celui qui envoie et fait la méthode ajax qui envoie le message ainsi que l'affichage.

Ensuite, pour chaque message, si on appui sur le bouton plus (deux points en haut à droite) on affiche les détails de chaque messages, vu, quand ainsi qu'une croix de suppression.

Mon code s'est mis à ne plus marché, Benjamin a trouvé que cela venait de mon code socket.io (4, 5 semaines en arrière), je n'avais pas ajouté de condition pour tester si la socket existait et par conséquent, ça bloquait tout.

J'ai rajouté un effet bold si le message est non lu (dernier message dans liste discussions).

On a changé de plan, on remplace la liste des contacts directement par une nouvelle discussion avec la possibilité d'ajouter des utilisateurs. La création de discussion marche.

J'ai fait une mise au point de ce qu'il manque, beaucoup de choses néanmoins :

- Mettre le bloc discussion dans un fichier à part qui pourra être appelé en ajax.
- Choisir les utilisateurs à la nouvelle discussion sauf moi.
- Enlever la vue dès qu'on a créé la discussion et revenir sur la liste des discussions.
- Si tous les utilisateurs ont supprimé leur détail de message, supprimer le message original
- Supprimer une discussion s'il n'y a plus de messages.

- Fermer la discussion dès qu'il y a plus de message...
- Vérifier qu'il y a des messages pour une discussion pour ne pas l'afficher vide (on supprime).
- Rafraichir la liste des discussions.
- Tout passer en vu lorsque l'on clique sur une discussion.
- Mettre les vu/ pas vu sur chacun de mes messages envoyés

J'ai mis le bloc html de discussion dans un fichier. J'ai mis la condition pour ne pas s'ajouter soi-même ou ne pas parler à soi-même dans une discussion. J'ai aussi réussi à afficher la discussion dans la liste des discussions mais ça n'affiche pas le dernier message.

J'ai par la suite voulu optimiser mon code. J'ai changé du code pour mettre sur la même fonction le traitement d'un message et le traitement d'une nouvelle discussion car c'est sur le même formulaire, si la discussion n'est pas fournie on en crée une. Le dernier message s'affiche bien. La création de discussion marche ainsi que l'envoi de message

Petit problème : Lorsque je crée une discussion, la page de la discussion apparait mais le retour en arrière dans la liste des discussions ne fonctionne pas et l'évènement 'enter' n'est plus pris en compte. On a réglé ce problème, il fallait recharger le plug-in.

J'ai réussi à passer en vu chaque message que j'écris moi-même et à l'ajouter en dernier message de la liste des discussions : je devrais faire pareil en temps réel quand je reçois un message.

J'ai beaucoup de problème avec l'envoi la suppression, le rafraîchissement de la liste en ajax, car des requêtes étaient effectuées plein de fois donc ça semait le trouble.

J'ai réglé tous ces problèmes et j'ai refait une mise au point :

J'ai fait les discussions en cascade, c'est à dire :

Lorsqu'un utilisateur supprime un message (en réalité un userMessage), on vérifie qu'il reste encore des userMessage s'il n'en reste pas, on supprime le message de base pour économiser la mémoire. Quand on supprime un message, on vérifie que la discussion contient encore des messages, si non, on supprime la discussion. J'ai par la suite, ajouté sur l'interface un bouton pour supprimer une discussion.

J'avais un problème avec l'évènement clic, lorsque je clique sur une discussion pour afficher le détail, il y a un évènement clic qui va passer les messages en vu (ajax) mais l'évènement ne marchait pas sur toute la div en question, après avoir essayé plusieurs choses, je me suis dit si je clique sur une discussion, la vue de détail (affichant toute les discussion) étant inclut et plus grande que la vue de la liste qui affiche le dernier message, elle s'affiche et la souris sera toujours dedans la div donc mouseover, et ça a très bien marché, j'ai optimisé, mis des booléens pour éviter l'appel ajax à chaque mouvement de souris.

Un moment donné, l'utilisateur simple n'avait plus la liste déroulante des utilisateurs à ajouter à une discussion lorsqu'il en crée une...c'est un problème du filtre administrateur car on utilise une fonction qui laissait passer seulement les administrateurs. Comme c'est Benjamin qui a codé ça, il a dit qu'il verra ça plus tard.

Il a réglé le problème pendant que je finissais la partie vu/pas vu, quand je clique sur les deux points, je peux voir à quel heure l'interlocuteur a vu mon message et à quelle heure j'ai vu les siens. Cela ne marchait pas complètement.

Ensuite il m'a expliqué le problème, on a ajouté la petite pastille verte si l'utilisateur est connecté.

J'ai continué en ajoutant un petit bouton plus permettant d'afficher les croix pour chaque discussion afin de les supprimer avec une confirmation.

J'ai fait une dernière mise au point :

- Bloquer l'ajout d'utilisateur à un.
- Vérifier qu'une discussion existe déjà.
- Retourner à la ligne en changeant en textarea et en mettant un événement de type input
- Toute la partie temps réel
- La partie détail message était laissée de côté car beaucoup d'erreurs, je m'y remets

J'ai passé de 5h de temps pour faire l'algorithme qui vérifie qu'une discussion existe ou pas, un algorithme qui n'était pas bon, donc j'ai dû repenser.

J'ai donc fait un algo.

- Je prends les discussions de l'utilisateur en question (celui qui en a le moins si j'ai le temps plus tard)
- Je parcours chaque discussion
- Je parcours tous les identifiants utilisateur que je veux rajouter
- Je vérifie qu'il y a une association discussion_id user_id
- Si oui l'utilisateur a déjà une discussion avec moi
- Et dès qu'un renvoie false c'est que la discussion n'existe pas car il manque celui ci

J'ai fait une heure voire deux de débogage car j'ai fait une erreur dans la requête qui compte en fonction de la discussion et de l'utilisateur sur le booléen à retourner, je renvoie maintenant directement le nombre et si c'est 0 c'est qu'il n'y a pas de discussion qu'on partage.

Ensuite j'ai bougé le code au bon endroit.

Afficher la discussion qui existe déjà si elle existe déjà. Je n'ai pas fini cette partie.

J'ai continué et fait ce qu'il me restait à faire, afficher la discussion existante si elle y est. L'essentiel du travail était de modifier mon code Javascript car je n'avais pris en compte que les cas de figures suivants.

- Ajouter un message, ou ajouter une discussion.

J'ai réglé le problème de l'utilisateur qui supprime la discussion...

Après cela, Benjamin est venu voir mon code, et là, on a constaté plusieurs bugs car on ne supprime plus un utilisateur d'une discussion, et tout mon code se basait sur ça.

On a passé toute une matinée à régler quelques bugs et à optimiser le code.

On a aussi ajouté la sécurité sur les requêtes ajax

J'ai commencé la partie temps réel le dernier Jeudi (je dois tous finir avant vendredi soir)

On a tout fait en 2h30 alors que je croyais que cela allait me prendre des jours et ne pas avoir le temps d'ici la fin du stage.

Une fois le code optimisé, on a fait les derniers tests et on a ajouté cela au projet.

Je me suis enfin remis sur la partie détail, on a compris d'où venait les problèmes, quand on supprime le détail d'un message et que quelqu'un d'autre affiche la discussion, certains détails vont être à vide car les propriétaires les ont supprimés, et cela lançait une exception. On devait donc vérifier que le détail existe avant de vouloir afficher les informations.

3/ Bilan

3.1 L'expérience professionnelle en entreprise

Le stage m'a apporté beaucoup d'expérience aussi bien au niveau technique que relationnelle.

Cela m'a apporté une nouvelle façon vision dans la façon de réaliser des applidations, et m'a aidé à me remettre en question sur beaucoup de mes défauts en tant que développeur. La première chose dont je peux conclure est de toujours chercher la solution optimale mais qui ne consomme pas trop de ressources car il faut penser au fait que l'application pourra être utilisée par des centaines, voire des milliers de personnes.

Le travail que j'ai réalisé, hormis quelques petits points, répondait aux attentes de Benjamin

Cependant j'ai remarqué quelques points sur ma façon de coder qui ne va pas avec un vrai travail de projet :

- Il y a des moments où je réfléchis trop alors qu'il y a une solution très minime ou déjà prête
- Trouver une solution avec des outils ou langages que je ne maîtrise pas est difficile pour moi, même si cela est arrivé tout au long du projet, j'insiste surtout sur le temps que cela me prend

Malgré ces points cités, j'ai remarqué une grande amélioration dans mon autonomie, il était facile pour nous d'appeler un professeur lorsque nous rencontrions un problème à l'IUT, Or, là, nous avions affaire avec des collègues de travail, on ne pouvait pas les appeler tout le temps, j'étais dans l'obligation de trouver mes réponses seul, et cela est meilleur.

J'ai, bien sûr acquis des connaissances techniques de programmation : Symfony, Node.js. Mais le plus important pour moi, est d'avoir progressé dans l'algorithmique, dans la vision des choses, savoir ce que l'on veut, comment le faire, anticiper de futurs cas ou erreurs, suivre les choses étape par étape, ne pas s'accélérer. Peu importe soit le langage utilisé, on peut s'adapter, mais la vision de ce que l'on cherche à réaliser reste inchangée à partir du moment qu'on l'a.

Ce qu'il me manque est plus est un bagage d'expérience. Il est clair qu'une personne ayant une carrière plus remplie que la mienne aurait fait mon travail en très peu de temps, car elle connaît déjà tous les vices et a déjà eu affaire et trouve commun les erreurs que je venais de découvrir.

Au niveau relationnel, je m'entendais très bien avec tout, le monde, on a passé de bons moments, allait quelque fois au restaurant. Avec Benjamin, on a très bien communiqué pour pouvoir produire du code sans avoir aucun problème. La communication a été très importante car elle a permis de ne pas me perdre durant le projet.

3.2/ La formation

La formation au Département informatique est très intéressante car elle permet une insertion professionnelle rapide et directe. Le contenu de la formation est fait de telle sorte que l'on ait assez pour continuer à évoluer seul tout en étant productif.

L'ensemble des connaissances acquis pendant mon séjour à l'IUT m'ont été utile lors de ce stage, aussi bien dans la base de données que dans la programmation et l'algorithmique.

J'ai su réutiliser les compétences acquises pendant la formation sans trop de problèmes car nous étions bien formés. Cependant, utiliser des Framework et librairies était moins facile, mais tout à fait faisable, car un point essentiel qui est très utile est de comprendre des programmes et algorithmes déjà fait, de plus nous avons aussi utilisés quelques librairies donc je connaissais la marche à suivre.

Je pense avoir les aptitudes nécessaires en tant que développeur grâce à cette formation, notamment dans la recherche et la réalisation du besoin. Il est maintenant plus dur d'intégrer et d'utiliser ces connaissances sur un tout nouveau projet l'on vient de commencer.

3.3/ Conclusion

Durant tout ce stage, j'ai réalisé le travail que je devais faire, en partant des deux petits formulaires, les notifications, les patterns en passant par les news et finalement le plus gros travail qui était le tchat.

Je me suis vite mis dans le bain lors de la deuxième semaine après avoir vu les concepts de Symfony puis j'ai enchaîné sans trop aller vite sans aussi perdre énormément de temps par la suite. Je n'ai aussi pas eu d'énormes problèmes de compréhension, je devais juste faire des recherches sur les librairies, les documentations, mais j'avais une vision de ce que je devais faire et je demandais à Benjamin dans le cas contraire.

Je dirais que le monde de l'entreprise est un monde totalement nouveau, très différent du monde scolaire, mais c'est un monde sans secret particulier, si on est compétent, on réussit les missions données. Ce stage a été très riche en expérience pour moi car cela m'ouvre les yeux sur beaucoup de choses : Sur la façon de coder, comment rechercher.

Cela m'a aussi appris à faire une mission du début à sa fin, accomplir un travail, sans relâcher car cela n'est pas possible, on est payé pour faire ce que l'on doit faire.

Ce fût une expérience très enrichissante humainement, socialement, et surtout techniquement.

Concernant mon projet professionnel :

Sur le court terme, j'envisage de terminer mes études (avoir un master puis ensuite faire une thèse) tout en continuant mon site internet (ce stage m'a beaucoup aidé pour pouvoir le continuer).

Sur le moyen terme, je pense travailler en entreprise pendant quelques années (5 ans au plus) afin d'acquérir de l'expérience tout en continuant à gérer mon site internet (si cela marche). J' imagine travailler plus tard aux Etats-Unis, au Rwanda ou rester en France (Paris si possible).

Enfin, sur le long terme, je veux travailler pour moi-même, être indépendant, créer mon entreprise à Marseille ou au Sénégal, ou bien les deux si possible. J'hésite encore sur ces points là, mais j'ai beaucoup de temps devant moi et de conditions à remplir avant de prendre la décision.