

*Tom Friang*

# DOSSIER DE SYNTHÈSE

---

LABORATOIRE ICUBE  
UNITÉ MIXTE DE RECHERCHE 7357



Laboratoire des sciences de l'ingénieur  
de l'informatique et de l'imagerie

# SOMMAIRE

---

<b>Présentations</b>	<b>3</b>
Tom, mais qui suis-je ?	3
Elan Formation, c'est quoi ?	4
Laboratoire ICube	5
Le Service Informatique du Laboratoire	5
<b>Le Projet</b>	<b>6</b>
Le projet : mais lequel ?	6
Mise en place du projet	6
La réalisation	8
Phase de test	9
Mise en oeuvre du projet	11
Le front-end ou l'interface utilisateur	14
Mise en prod	23
Difficultés rencontrées	23
Améliorations possibles du projet	24
Bilan du projet	25
<b>Projets annexes</b>	<b>27</b>
Listes invités	27
Traduction	28
DébUgage	28
Migration Symfony 2.8 vers Symfony 3.4	28
<b>Divers</b>	<b>29</b>
<b>Dictionnaire</b>	<b>30</b>



# PRÉSENTATIONS

---

## **TOM, MAIS QUI SUIS-JE ?**

Je m'appelle Tom Friang, j'ai 27 ans et je suis actuellement en reconversion professionnelle après avoir travaillé pendant 7 ans dans l'animation socio-culturelle.

J'ai choisi le développement web par affinité avec le monde de l'informatique et c'est pourquoi, aujourd'hui, je suis inscrit à une formation de développeur web à Elan Strasbourg afin d'acquérir d'une part les bases d'un langage de programmation et d'un framework (ici PHP et Symfony principalement), mais aussi de pouvoir obtenir un titre RNCP III qui me permettra de continuer en titre d'ingénieur en alternance.

En parallèle, je suis également inscrit à une formation à distance au CNAM Grand Est afin de valider deux unités d'études (NFP135 et NFP136) complémentaires à mon titre.

## ELAN FORMATION, C'EST QUOI ?

“La formation sur mesure !” telle est leur devise.

Elan est un centre de formation fort de 25 années d'expérience en intra ou inter dans les domaines de la bureautique, l'informatique, la PAO, le multimédia et techniques de secrétariat.

Il propose avant tout de la formation individualisée et sur mesure.

Il dispose de locaux à Strasbourg, Sélestat, Haguenau, Saverne, Colmar, Mulhouse, Metz et Nancy.

Leurs méthodes pédagogiques sont :

- Ecouter et comprendre la demande précise de l'entreprise ou du stagiaire ;
- Adapter une formation qui prenne en compte la singularité de l'apprenant ;
- Guider le stagiaire en permanence grâce à un formateur ;
- Anticiper ses attentes ;
- Suivre l'évolution des acquis tout en avançant à son rythme ;
- Valoriser la formation et certifier les compétences acquises ;
- Valider la pertinence de notre formation.



## LABORATOIRE ICUBE

*Le laboratoire des sciences de l'ingénieur, de l'informatique et de l'imagerie*

L'année 2013 voit la naissance du laboratoire ICube, un formidable projet sous l'égide du CNRS, de l'Université de Strasbourg, de l'ENGEEES et de l'INSA de Strasbourg.

Le laboratoire rassemble à parts égales deux communautés scientifiques à l'interface entre le monde numérique et le monde physique, lui donnant ainsi une configuration unique.

Avec près de 650 membres, dont 350 chercheurs et 250 doctorants, il est une force de recherche majeure du site de Strasbourg. Fédéré par l'imagerie, l'ICube a comme champs d'application privilégiés l'ingénierie pour la santé, l'environnement et le développement durable



Responsable actuel :  
Michel DE MATHELIN  
Laboratoire iCube  
300 Boulevard Sébastien  
Brant  
67412 ILLKIRCH

## LE SERVICE INFORMATIQUE DU LABORATOIRE

*là où s'est déroulé mon stage*

Mon tuteur a été Monsieur Jean-Marc MULLER, ingénieur systèmes et réseaux et responsable du service mais j'ai principalement travaillé avec Madame Aline GERARD, ingénieur d'étude, en charge du développement interne au laboratoire, responsable et développeuse de l'intranet du laboratoire.



# LE PROJET

---

## LE PROJET : MAIS LEQUEL ?

En effet, je n'ai pas eu à travailler sur un projet unique.

Mon principal projet est celui qui m'a pris 5 semaines sur les 8 de mon stage : développer et intégrer, à l'intranet actuel, un générateur automatique de listes de diffusion.

Ces listes sont basées sur les emails des membres du laboratoire, mais aussi sur les équipes, les statuts, grades, plateformes, sites, localisation.

Au final c'est environ 500 listes à automatiser et une centaine à maintenir pour un total de plus de 600 listes de diffusion concernant le laboratoire et ses membres.

Les consignes étaient claires : réaliser l'automatisation de ces listes à travers le Framework Symfony 2.8, qui est le framework utilisé pour l'intranet, afin que le robot de liste "Sympa" puisse les récupérer et générer des listing de mails.

## MISE EN PLACE DU PROJET

*ou "cahier des charges" pour les intimes*

Après une après-midi "découverte" du laboratoire en compagnie de mon tuteur, Jean-Marc, la réalisation de mon badge d'accès, la paperasse administrative, nous avons configuré un ordinateur sous Linux ainsi que l'accès réseau.

J'ai rencontré ma collègue Aline, responsable du développement web au service informatique de l'iCube.

C'est avec elle que j'ai collaboré sur la majeure partie de mon projet afin de réaliser le développement et la mise en production.

Jean-Marc m'a tout d'abord expliqué comment fonctionne le système actuel ainsi que les besoins du laboratoire vis-à-vis de ces listes de diffusion. Ces listes sont nécessaires

pour la communication interne au laboratoire. Les membres peuvent utiliser une liste pour contacter une partie d'une équipe, par exemple.

Pour exemple, une équipe "AVR" constituée de 20 membres, dont 10 doctorants, 9 techniciens et 1 stagiaire feront parti respectivement des listes :

- avr
- avr-doctorant
- avr-technicien
- avr-stagiaire

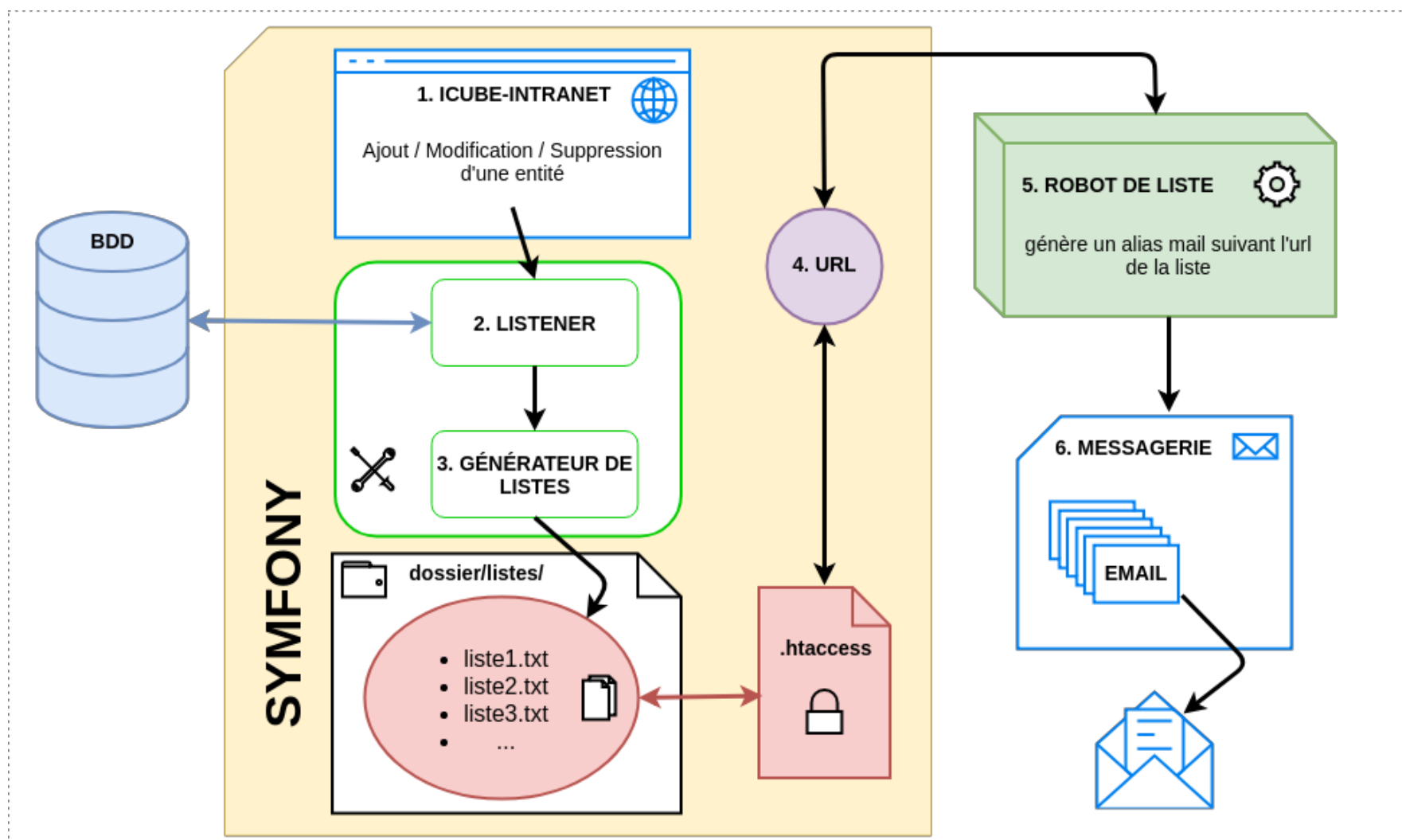
Donc :

1. pouvoir générer ces 4 listes sous forme de fichier texte (.txt) avec les emails de chaque membre ;
2. placer ces fichiers à l'intérieur d'un dossier dans le projet Symfony ;
3. pouvoir récupérer l'url de chaque liste afin de paramétrer le robot de liste sur chaque url. Ce robot générera ensuite un alias email en fonction du nom de la liste et des emails contenus à l'intérieur.

Par ailleurs, il faudra pouvoir également avoir accès à ces listes de manière plus conventionnelle. Pour cela, une interface graphique dans l'intranet devra être réalisée afin de :

1. pouvoir afficher l'intégralité des listes ;
2. pouvoir consulter le contenu d'une liste ;
3. rechercher un membre dans les listes ;
4. pouvoir imprimer le contenu d'une liste et la liste des listes.

La partie "front-end" viendra bien évidemment à la fin du développement du fait qu'on ne pourra afficher une liste que lorsqu'elle existera !



représentation globale du projet

## LA RÉALISATION

Pour ce projet, on ne m'a pas fixé de contrainte de temps. Si je le réalise en 1 semaine ou bien en 2 mois, peu importe, même si le caractère urgent de l'existence de ces listes de diffusion pour le laboratoire est important.

Cependant, j'ai quand même voulu réaliser ce projet dans une contrainte de temps, impossible pour moi de ne pas venir à bout de ce projet et encore moins de ne pouvoir concrétiser la demande, cela me permet de me mettre en situation de challenge également pour cette première "vraie" expérience professionnelle dans le développement web.

J'ai donc commencé par me documenter.

A mon sens, la base d'un bon développement est de savoir de quelle manière on va travailler sur le projet, quelles technologies on va devoir utiliser mais aussi sur quel environnement de travail on va pouvoir développer.

J'ai aussi échangé avec Aline sur le projet. Connaître son point de vue a été très bénéfique pour moi car cela m'a donné beaucoup d'idées de comment procéder quand j'entamerai la partie de développement. Elle m'a également fait découvrir le code source du projet, ainsi que la base de données.



Il faut également savoir que ce projet est le fruit d'un développement de près de 7 ans de la part d'Aline qui est la seule à développer au sein du service informatique.

## « Mais pourquoi ne pas utiliser une base de données ? » »

Le robot de listes aurait la possibilité d'alimenter le contenu de ses listes grâce à des requêtes sur la base de données. Cette solution dans le cadre des listes du laboratoire n'a pas été jugée optimale. La mise à jour des listes par le robot étant effectué toutes les cinq minutes, procéder par le biais de requêtes directes sur la BDD génèreraient 600 requêtes toutes les 5 minutes, soit 172800 requêtes par jour. La base de données du laboratoire étant utilisée par plusieurs applicatifs, procéder ainsi aurait dégradé de façon significative la performance des applicatifs. Il a donc été convenu d'alimenter le robot de listes à partir de fichiers textes générés uniquement quand une modification se produit dans la BDD. Ceci réduit ainsi, de manière significative, le nombre de requêtes sur la BDD.

## PHASE DE TEST

Avant de commencer mes phases de tests, j'ai posé mes idées sur des feuilles de papier que j'ai accroché dans mon bureau (je voulais utiliser du marbre et un burin pour plus d'effet, mais je n'en ai pas trouvé...)

J'ai essayé de visualiser à quoi les listes devront ressembler et comment les générer à travers Symfony. J'ai d'abord cherché des idées sur internet mais c'est surtout d'avoir échangé mes idées avec Aline qui m'a permis d'établir un plan raisonnable.

Pour commencer, il me fallait un environnement de travail. J'ai choisi d'utiliser l'IDE PhpStorm étant donné que la majorité du développement sera en php / Symfony et que c'est un IDE très utilisé dans le monde professionnel.

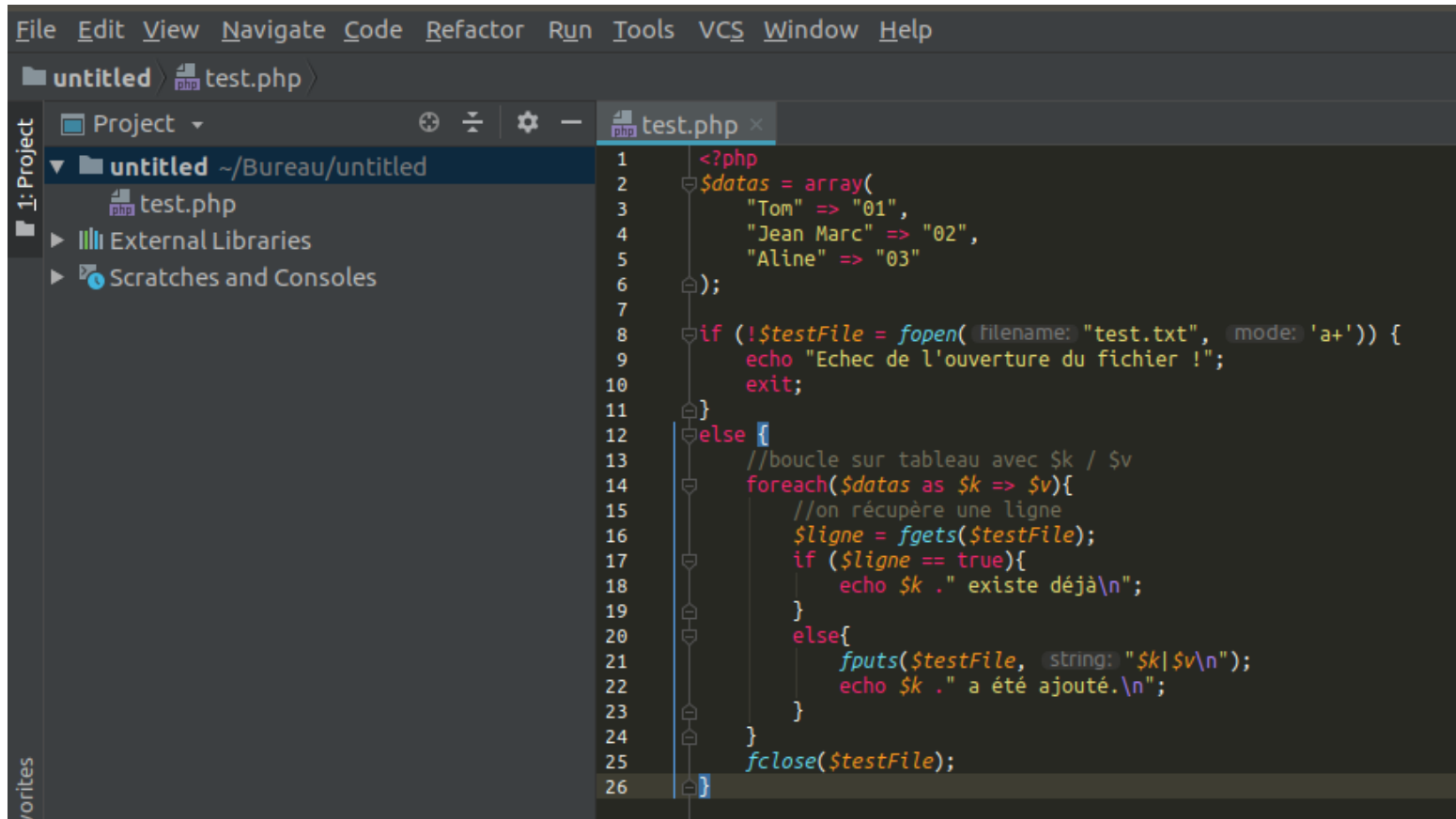
Ne connaissant pas l'utilisation de cet IDE, j'ai passé une bonne journée sur sa découverte et sa configuration .

Le reste de la première semaine, je me suis consacré à la réalisation d'un script permettant de créer un fichier texte à partir d'un tableau à 2 dimensions en php puis d'une base de données "test".

L'idée était de d'abord bien comprendre quelles méthodes existent et ensuite comment les mettre en place dans un bout de script.

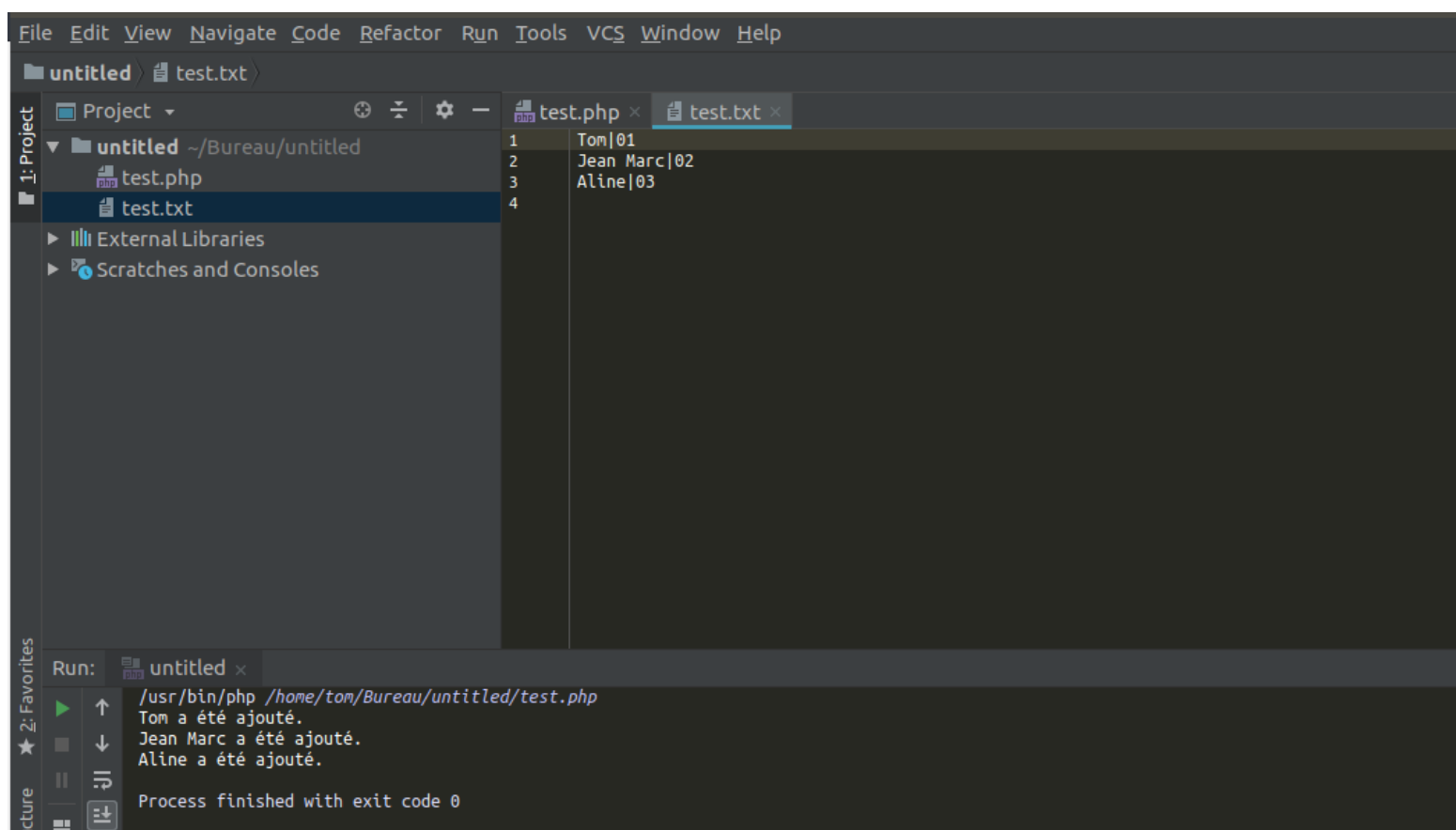
J'enregistrais les étapes de mon travail sur le système de versionning GitHub.

Par la suite j'ai mis en place un projet Symfony ainsi qu'une base de données "test" pour effectuer des premiers tests à travers Symfony. Les tests étant concluants, j'ai fait un point avec Aline pour savoir comment procéder ensuite sur le "vrai projet" et pouvoir commencer à développer le projet final.



```
1 <?php
2 $datas = array(
3     "Tom" => "01",
4     "Jean Marc" => "02",
5     "Aline" => "03"
6 );
7
8 if (!$testFile = fopen( filename: "test.txt", mode: 'a+' )) {
9     echo "Echec de l'ouverture du fichier !";
10    exit;
11 }
12 else {
13     //boucle sur tableau avec $k / $v
14     foreach($datas as $k => $v){
15         //on récupère une ligne
16         $ligne = fgets($testFile);
17         if ($ligne == true){
18             echo $k ." existe déjà\n";
19         }
20         else{
21             fputs($testFile, string: "$k|$v\n");
22             echo $k ." a été ajouté.\n";
23         }
24     }
25     fclose($testFile);
26 }
```

screenshot du générateur de listes originel



```
1 Tom|01
2 Jean Marc|02
3 Aline|03
4
Run:
/usr/bin/php /home/tom/Bureau/untitled/test.php
Tom a été ajouté.
Jean Marc a été ajouté.
Aline a été ajouté.
Process finished with exit code 0
```

screenshot du résultat du générateur de listes sous phase de test

## **MISE EN OEUVRE DU PROJET**

### *ou phase de développement*

J'ai commencé à développer sur le projet à la fin de la deuxième semaine de stage.

Je dois l'avouer, je suis peut-être parti un peu rapidement sur le développement car très vite, j'ai remarqué que je n'avais pas très bien compris ni le principe ni le fonctionnement de ces listes. Malgré cela, j'ai continué à développer sur le projet tout en comprenant au fur-et-à-mesure le fonctionnement de ces listes.

J'ai aussi continué à questionner Aline sur le fonctionnement de l'application et sur les besoins du service administratif vis-à-vis de l'intranet.

A ce stade du projet, j'ai également consulté Madame Eglantine HIRSCH, qui fait parti de l'équipe d'administration du laboratoire et qui utilise régulièrement ces listes. Cela m'a permis de mieux comprendre les besoins finaux de l'utilisateur.

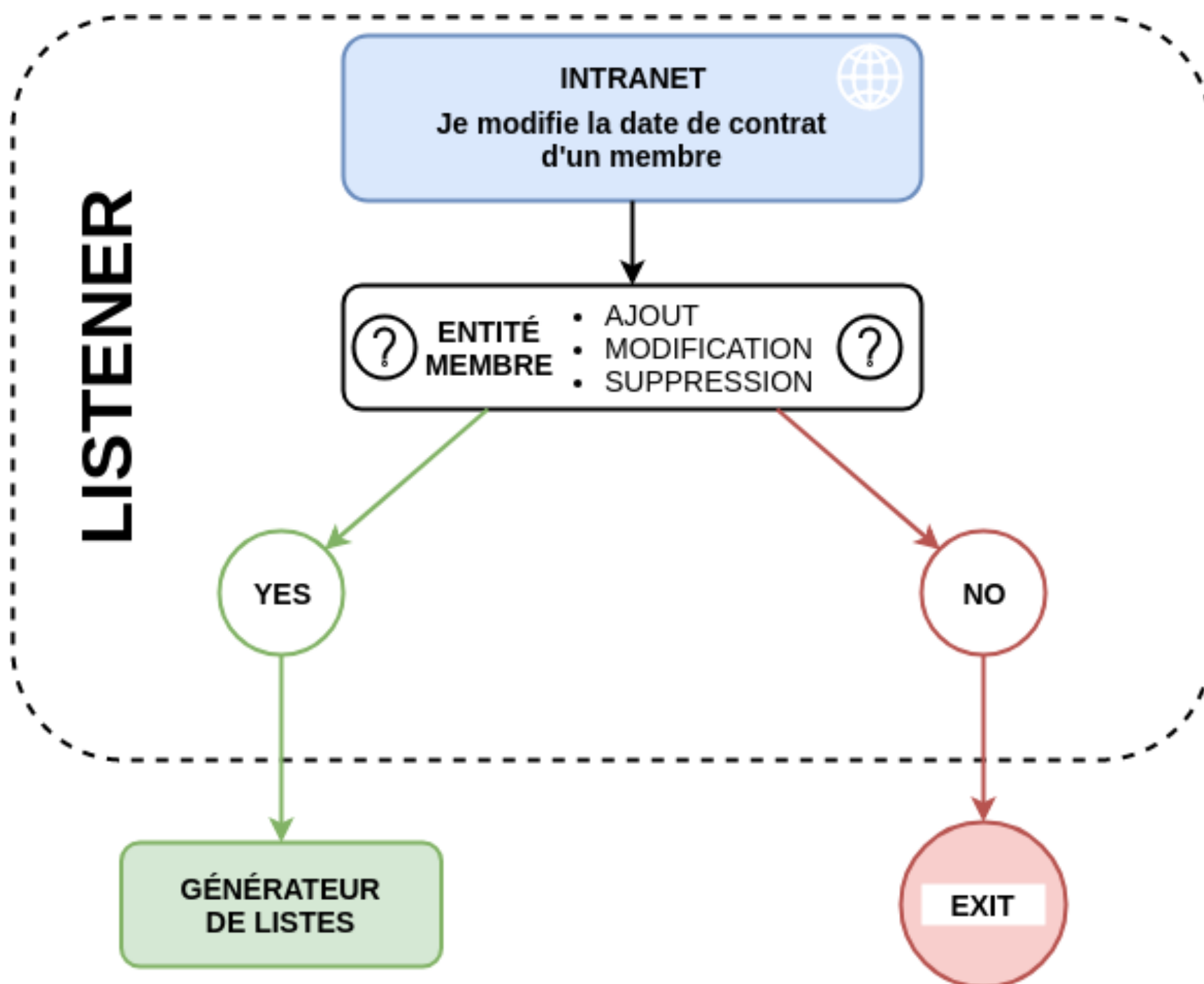
### *Event Listeners*

Dans Symfony il existe des "Event Listeners".

Lors de l'exécution d'une application Symfony, de nombreuses notifications d'événements sont déclenchées. L'application peut "écouter" ces notifications et y répondre en exécutant n'importe quel code.

On peut donc paramétrer un Event Listener à notre sauce afin qu'il effectue l'action nécessaire lors de l'une de ces notifications.

Ici l'intérêt d'un Listener est de mettre à jour une liste si une entité de la base de données est modifiée. Par exemple, si la date de contrat d'un membre est modifiée, nous allons donc modifier un champ dans l'entité "Membre" ce qui va nous permettre de mettre à jour la liste correspondante, ici la liste "all" qui est celle de tous les membres du laboratoire.



*action d'un Listener dans le framework Symfony*

Cela étant, il a fallu déterminer quelles listes sont le plus régulièrement mises à jour.

L'idée n'est pas de mettre toutes les listes à jour lorsqu'une entité est modifiée mais uniquement celles qui sont nécessaires ou qui sont en rapport avec l'entité.

C'est pourquoi j'ai développé un Listener en fonction d'une création d'un membre ou d'une mise à jour de celui-ci.

Évidemment, l'appartenance ou non d'un membre dans une liste est définie par plusieurs critères. Les principaux critères sont un accès ou non de l'utilisateur à l'intranet, la date de fin de son contrat à l'iCube ou si cette date est nulle.

Cela peut ensuite être défini par équipe, par grade, statut, école doctorale, ...

Pour ce faire, j'ai recherché, au moment de l'action sur une entité, les champs à tester pour pouvoir inscrire un membre dans la liste correspondante.

```
$status = $member->getLastStatus();  
$typeStatus = $status->getTypeStatus();
```

J'ai du effectuer de nombreux tests tout en gardant une lecture claire du code pour en faciliter le développement mais également la maintenance..

Lors de ce travail, je me suis rendu compte que plus les listes étaient "sous-catégorisées" (ex : la liste "image-irts-cdd" qui est la liste regroupant tous les CDD (ITA et BIATSS) de l'équipe "image-irts", où ita sont les membres ita du laboratoire et biatss sont les membres qui font partis des organismes Universités de Strasbourg, Engees, Ecam, Hôpitaux universitaires de Strasbourg et INSA), plus il s'avérait difficile de produire la liste en question.

En expliquant cette problématique à Aline, nous avons convenu que je ferai toutes les listes liées à une équipe (c'est-à-dire que si une équipe possède des membres dans des catégories différentes, cela générera la liste principale de l'équipe ainsi que toutes celles qui en découlent).

Cela a simplifié grandement la recherche des bons éléments à tester pour établir une liste et par conséquent gagner du temps sur le développement.

« *Il va de soit que les listes peuplées avec des emails de membres seront mises à jour, mais qu'en est-il des listes vides ?* » »

Car en effet, le laboratoire possède aussi des listes d'équipes vides pour prévoir le jour où une personne devra être inscrite dans cette liste mais avant tout parce que le robot doit être paramétré sur l'url d'une liste qui existe bel et bien.

Nous avons réfléchi avec Jean-Marc et Aline à cette éventualité surtout avec le Listener qui ne met à jour que l'entité modifiée.

De ce constat, j'ai proposé de faire un générateur de toutes les listes, en plus du Listener, ce qui permettra, d'une part, de mettre à jour toutes les listes au même moment, ensuite de mettre à jour les listes qui ne sont pas impactées par le Listener et enfin de générer les listes vides.

## LE FRONT-END OU L'INTERFACE UTILISATEUR

C'est lorsque j'ai fini de développer le Listener que je me suis attaqué à une partie moins amusante mais toutefois faisant parti de mes compétences : le front-end.

Cela m'a permis de visualiser le travail réalisé jusqu'à présent et de mettre un peu de couleur en dehors de la coloration syntaxique de l'IDE !

La partie graphique est majoritairement réalisée grâce au framework Bootstrap ainsi que EasyAdmin (un bundle graphique de Symfony).

J'ai réalisé l'interface que les utilisateurs pourront voir concernant les listes de diffusions.

*En voici quelques exemples :*

On peut accéder à cette page à travers les menus de navigation de l'intranet.

The screenshot displays the ICube Intranet interface. The top navigation bar includes the 'ICube Intranet' logo, a hamburger menu icon, and user information for 'Aline User'. The left sidebar contains various navigation items: 'Tableau de bord', 'Statistiques', 'Membres', 'Trombinoscope', 'Export', 'SST', 'Gestion des éléments', 'Wikis', and 'Listes de diffusion'. The main content area shows the 'Listes de diffusion' menu, which is expanded to reveal a list of 484 lists. The table lists the following items:

#	Liste	
1.	acr	
2.	acr-hus	
3.	all	
4.	avr	
5.	avr-acb	
6.	avr-acr	
7.	avr-adt	
8.	avr-ai	ajouter description Modifier
9.	avr-cdd	ajouter description Modifier
10.	avr-cr	ajouter description Modifier
11.	avr-doctorant	ajouter description Modifier
12.	avr-dr	ajouter description Modifier
13.	avr-hdr	ajouter description Modifier

The right sidebar contains a search bar labeled 'rechercher une liste ...' and a 'Liste' button. The bottom of the interface shows the URL 'localhost:8000/[redacted]list/' and a Symfony logo.

exemple de menu "Listes de diffusion"



ICUBE Intranet

0

Aline User

Tableau de bord

Statistiques

Membres

Statistiques

Membres

Trombinoscope

Export

SST

Gestion des éléments

Wikis

Listes de diffusion

Listes de diffusion

Créer une liste

Rechercher

Liste

Imprimer

484 listes

Rechercher une liste ...

#	Liste	Description
1.	acr	ajouter description Modifier Détails
2.	acr-hus	ajouter description Modifier Détails
3.	all	ajouter description Modifier Détails
4.	avr	ajouter description Modifier Détails
5.	avr-acb	ajouter description Modifier Détails
6.	avr-acr	ajouter description Modifier Détails
7.	avr-adt	ajouter description Modifier Détails
8.	avr-ai	ajouter description Modifier Détails
9.	avr-cdd	ajouter description Modifier Détails
10.	avr-cr	ajouter description Modifier Détails
11.	avr-doctorant	ajouter description Modifier Détails
12.	avr-dr	ajouter description Modifier Détails
13.	avr-hdr	ajouter description Modifier Détails

interface représentant la liste des listes

Sur ce second screenshot de l'application, on peut voir la "liste des listes", comme on l'appelle dans le milieu. C'est un tableau affichant toutes les listes du laboratoire. On peut accéder aux détails d'une liste en cliquant sur le bouton "détails".

ICUBE Intranet

Retour aux listes

Détails de la liste : **ACR-HUS**  
Total de : 6 emails

Rechercher une liste ...

Imprimer cette liste

#	eMail membre
1.	[redacted]@chru-strasbourg.fr
2.	[redacted]@chru-strasbourg.fr
3.	[redacted]@unistra.fr
4.	[redacted]@chru-strasbourg.fr
5.	[redacted]@chru-strasbourg.fr
6.	[redacted]@chru-strasbourg.fr

Copyright © 2011-2018 ICUBE All rights reserved.

détails d'une liste

Une fonction de recherche réalisée en JavaScript (JS) permet de rechercher une liste par son nom.

ICUBE Intranet

Listes de diffusion

Créer une liste Rechercher Liste Imprimer

484 listes

inet

#	Liste	Description
245.	inetlab	ajouter description Modifier Détails
246.	inetlab-ai	ajouter description Modifier Détails
247.	inetlab-cdd	ajouter description Modifier Détails
248.	inetlab-contact	ajouter description Modifier Détails

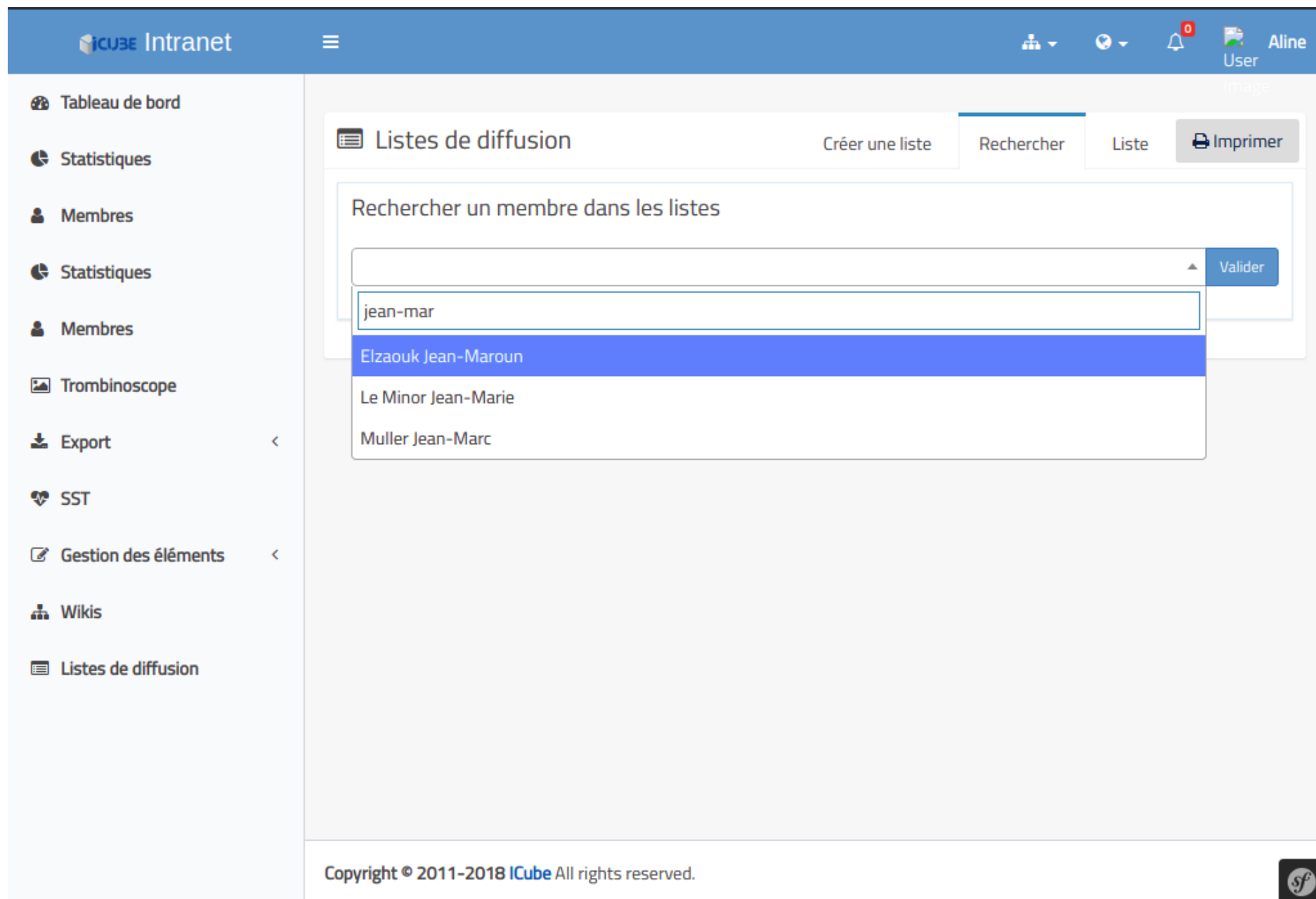
fonctionnalité de recherche d'une liste par son intitulé

Sur la même page, mais dans un autre onglet, j'ai ajouté une autre fonction de recherche qui permet de rechercher un membre dans les listes. J'ai utilisé Select2, un

bundle JQuery, qui permet de créer un tableau JSon avec tous les membres du laboratoire pour proposer une sélection à la frappe.

On peut ensuite sélectionner le membre et lancer la recherche.

Note : Si le membre que l'on recherche n'apparaît pas dans la liste, c'est que ce membre n'existe pas.



recherche d'un membre

Sur ce screenshot, on peut voir le résultat de la recherche d'un membre. Le résultat de la recherche affiche les listes dans lequel il est inscrit.

Pour ce faire, j'ai créé une méthode dans mon controller qui permet de rechercher l'email du membre dans toutes les listes. Je parcours chaque fichier de liste pour les lire ligne par ligne. Si je trouve l'email du membre dans une ligne, j'inscris le nom du fichier (donc de la liste) dans un tableau à 2 dimensions php pour ensuite le transmettre à ma vue afin d'afficher le résultat.



Enfin, dans le profil de la personne connectée, j'ai placé un cadre pour afficher l'appartenance du membre dans des listes.

The screenshot shows the ICUBE Intranet interface. At the top, the user is identified as 'John User'. The main profile section for 'M. John TRAVOLTA' includes a 'Modifier les informations' link, a profile picture placeholder, and contact details like email and website. To the right, the 'Informations personnelles' section shows a birth date of 18/02/1954 and a location of DE. Below this, the 'Appartenance' section lists 'Service Commun - Services centraux - Informatique'. The 'Évolution' section contains a table of career progression:

Ingénieur d'étude	Date	Quotité	Taux d'affectation	Université de Strasbourg
Ingénieur d'étude	10/04/2010 - 23/04/2016	100%	100%	Université de Strasbourg
Ingénieur d'étude	24/04/2016 - 12/11/2017	100%	100%	Université de Strasbourg
Ingénieur d'étude	13/11/2017	100%	100%	Université de Strasbourg

On the left, the 'Liste de diffusion' section shows a list of distribution lists: biatss, all, ita-biatss, ita, and biatss-unistra. A green button 'Voir toutes les listes' is at the bottom. Below this, there is a section for 'Inscription à la liste d'annonces : communication libre à destination du personnel du laboratoire ICube'.

cadre des listes du membre

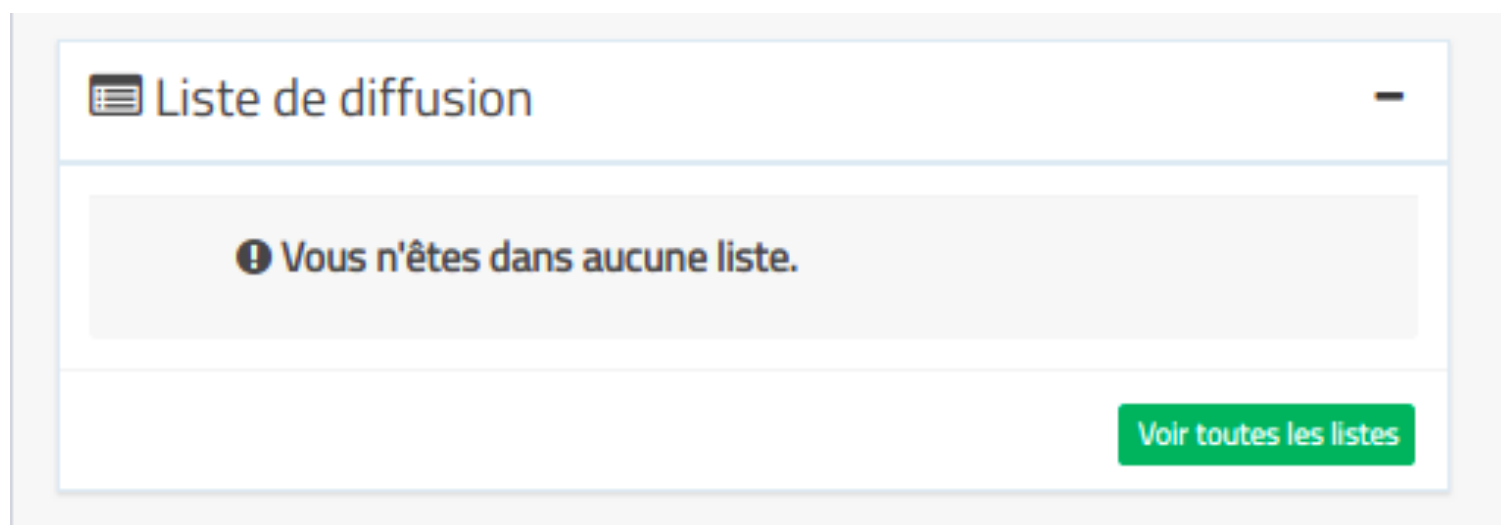
« Mais encore une fois, qu'en est-il des listes vides ? Comment apparaissent-elles sur l'interface ? »

Oui, il fallait le prévoir. Le détail d'une liste vide apparaît comme cela :

The screenshot shows the ICUBE Intranet interface with a sidebar menu on the left containing 'Tableau de bord', 'Statistiques', 'Membres', and 'Statistiques'. The main content area shows the details of a distribution list named 'AVR-ACB'. At the top, there is a green button 'Retour aux listes'. Below it, the text reads 'Détails de la liste : AVR-ACB' followed by a warning icon and the message 'Aucun membre ne fait parti de cette liste.' To the right of this message is a search bar with the placeholder text 'Rechercher une liste ...' and a magnifying glass icon.

affichage du détail d'une liste étant vide

◀ Et si le membre n'appartient à aucune liste ? ▶



*affichage profil membre appartenant à aucune liste ]*



## Command Symfony

Il est possible à travers le framework de créer une commande de terminal. Là aussi, le principe est simple : créer une méthode destinée à être appelée par ligne de commande.

Une fois créée, cette commande sera disponible comme commande à part entière dans l'application Symfony. On peut même y ajouter de l'aide et une description mais aussi un peu de style pour en faire une commande attirante et pro.

J'ai choisi de faire le générateur de toutes les listes grâce à "Symfony Command" du fait que ce soit très performant pour exécuter un script rapidement. Il permettra aussi d'appeler cette commande à travers un "cron", ce qui peut s'avérer utile pour la suite.

Pour développer la commande, je suis parti d'une feuille de papier et d'un stylo (la base) pour établir toutes les "grandes catégories" de listes (les équipes, les grades, les plateformes, etc) ainsi que leurs contraintes concernant un membre (la période du contrat par exemple). J'ai checké avec Jean-Marc et Aline pour savoir si tout était cohérent.

Mais n'avais-tu pas déjà fait cela avec le Listener ?

Effectivement, j'avais déjà réalisé une grande partie de ces tests à travers le Listener cependant le travail à travers une Command est différent et donc j'ai préféré "refaire" afin de tester proprement le développement de cette Command.

Une fois les idées à plat et après avoir développé le listener, le travail fut assez rapide. Je me suis aidé du listener pour tester la commande, à vrai dire j'ai développé l'intégralité du code à l'intérieur du listener afin d'avoir accès rapidement aux informations de mes dump.

Au fur-et-à-mesure du développement des grandes catégories, j'ai testé ma commande afin de vérifier que les bonnes listes se créent.

```
list
list:generate          Create or regenerate all mail lists files.
```

*la commande terminale dans le projet Symfony*

C'est aussi à travers cette commande que j'ai généré les fichiers vides car comme je l'ai précisé précédemment, le robot de liste a besoin d'une url existante et donc d'un fichier

existant pour fonctionner correctement. Dans son fonctionnement interne, le laboratoire a besoin de connaître l'existence, vide ou non, d'une liste.

Bien évidemment, le test final permet de générer un peu moins de 500 listes grâce à cette commande. La centaine de listes restantes sont des listes dites "manuelles" qu'on ne peut générer avec l'aide du listener ou de la command.

```
Terminal
+ 454/468 [=====>] 97% ~ biatss-insa.txt ~ successfully generated!
x 455/468 [=====>] 97% ~ dr.txt ~ successfully generated!
  456/468 [=====>] 97% ~ acr-hus.txt ~ successfully generated!
  457/468 [=====>] 97% ~ SERTIT-pr.txt ~ successfully generated!
  458/468 [=====>] 97% ~ SERTIT-PERMA.txt ~ successfully generated!
  459/468 [=====>] 98% ~ Hors équipe-PERMA.txt ~ successfully generated!
  460/468 [=====>] 98% ~ postdoc.txt ~ successfully generated!
  461/468 [=====>] 98% ~ postdoc-msii.txt ~ successfully generated!
  462/468 [=====>] 98% ~ cdd-insa.txt ~ successfully generated!
  463/468 [=====>] 98% ~ enseignant-chercheur-ensiee.txt ~ successfully generated!
  464/468 [=====>] 99% ~ perma-ensiee.txt ~ successfully generated!
  465/468 [=====>] 99% ~ cdd-cnrs.txt ~ successfully generated!
  466/468 [=====>] 99% ~ ita-inria.txt ~ successfully generated!
  467/468 [=====>] 99% ~ cdd-inria.txt ~ successfully generated!
  468/468 [=====] 100% ...
... PROCESS COMPLETED!
... 468 files were generated!
... end
tom@tom-HP-ProBook-650-G1:~/Bureau/ARHI-master$
```

*résultat des listes générées par la commande*

## MISE EN PROD

Le moment fatidique, mais qui révèle si le développement a été cohérent et bien pensé, est la mise en production !

Pour cette partie, c'est principalement Aline qui s'en est occupé du fait qu'elle reste responsable du développement et de la bonne tenue de l'application.

Cela a pris une bonne journée afin de merge ma branch sur le master du fait, qu'au final, j'ai tout de même modifié quelques fichiers critiques et bien évidemment ajouté de nouveaux fichiers.

En raison d'un petit bug sur le merge, nous avons dû vérifier, à ce moment là, les fichiers les uns après les autres.

La réelle mise en prod n'a eu lieu que le lendemain où un oubli de ma part a été constaté sur un test d'une entité vide. L'oubli a été corrigé rapidement.

Un retour utilisateur nous est parvenu le lendemain : impossible d'ajouter une personne au sein d'une liste.

Une raison simple car lors de la création des fichiers de listes à travers mon générateur, les droits utilisateurs définissent le propriétaire du ou des fichiers. En l'occurrence Aline est devenue la propriétaire lorsqu'elle a lancé la commande directement après la mise en prod. Donc aucun autre membre ne pouvait les modifier.

Problème résolu en changeant les droits de création et de modification par le serveur Apache à travers Symfony.

## DIFFICULTÉS RENCONTRÉES

Absolument tout était une galère sans nom.

Plus sérieusement, je suis arrivé avec beaucoup d'enthousiasme dans mon stage, sans forcément connaître le projet sur lequel j'allais intervenir. En découvrant le projet je n'avais pas imaginé à quel point il pouvait être critique pour le laboratoire et encore moins à quel point il était complexe. 600 listes, vous imaginez ?

Tout au long du développement j'ai été confronté à des "blocages" de part mon manque de connaissances dans le domaine mais j'ai aussi et surtout fait de nombreuses découvertes. J'ai découvert beaucoup de principes, de méthodes, de possibilités à travers le framework Symfony et php en général et c'est en gardant ma curiosité et ma volonté de

faire (jeu de mots) que je suis parvenu à développer le projet en répondant au cahier des charges.

La plus grande difficulté pour moi, dans le développement, c'est de remettre en question un code qui fonctionne. Le premier pas n'est pas évident du fait que ça fonctionne alors pourquoi irait-on le modifier ?

Le générateur de listes fonctionne ? Et bien on va le faire autrement.

C'est fait ? On va rajouter ça.

Et ainsi de suite !

Cette remise en question perpétuelle est peut-être difficile mais je trouve qu'elle en vaut le détour. On découvre de nouvelles choses, on pense autrement, on trouve de nouvelles idées, on rafraîchit le code, on échange avec d'autres développeurs et on peut même se faire de nouveaux amis sur les forums !

## **AMÉLIORATIONS POSSIBLES DU PROJET**

### *ou mises à jour*

- Modifier l'affichage de la liste des listes (on pourrait éventuellement griser les listes qui sont vides tout en gardant leur affichage afin de savoir qu'elles existent tout de même) ;
- Traduire en anglais le descriptif des listes dans l'interface (a été développé avant la fin du stage);
- Permettre à un responsable d'équipe de créer une liste personnalisée à travers l'interface utilisateur (a été développé avant la fin du stage).

## BILAN DU PROJET

Evidemment, le café faisait partie de cette réussite(!) mais pouvoir compter sur le soutien de mes tuteurs, sur une documentation assez fournie sur internet ainsi que sur ma simple curiosité permettent, selon moi, de débloquent les portes et de continuer à progresser.

Durant ce stage j'ai su mettre en application mes compétences techniques apprises lors de ma formation comme le développement de pages web à travers le framework Symfony, la manipulation d'une BDD MySQL avec l'ORM Doctrine et l'utilisation d'un IDE avec phpStorm. J'ai aussi développé des pages web dynamiques et des interfaces utilisateurs grâce à des langages comme JavaScript et le framework Bootstrap. J'ai également utilisé l'anglais en créant une interface de traduction des descriptions des listes.

J'ai su mettre en oeuvre mes compétences collaboratives en travaillant avec un autre développeur et partager mon développement avec le système de versioning Git. Nous avons pu développer de manière plus efficace corriger les bugs rapidement, fusionner nos travaux rapidement et être autonome sur nos développements respectifs. J'ai réussi, grâce à mes capacités d'écoute et de compréhension, partager et échanger mes idées avec un autre développeur. J'ai beaucoup appris de mes échanges avec Aline et Jean-Marc qui ont été deux personnes ressources durant mon stage mais qui le reste pour moi aujourd'hui. J'ai découvert de nombreux concepts de développement qui me permettent aujourd'hui d'être plus serein dans la suite de mon apprentissage.

À travers l'élaboration du cahier des charges, j'ai su mettre en oeuvre mes compétences de gestion de projet. J'ai choisi les outils ainsi que la méthodologie de travail collaborative avec les conseils de mes collaborateurs. Venait ensuite le développement, le débogage puis la mise en prod qui ont été des phases très riches pour moi notamment par les différents langages utilisés.

Mon savoir être m'a permis de mettre en oeuvre mes compétences humaines à travers l'échange, l'écoute et le partage auprès de mes collègues. Cela m'a rapidement permis de m'intégrer à l'équipe informatique du Laboratoire de prendre des cafés avec eux et de me faire entendre comme développeur à part entière dans mes choix et mes idées. La bonne entente dans l'équipe était un véritable moteur lors des phases compliquées du développement où la remise en question est fréquente.

Après seulement 2 mois je remarque que je suis capable de m'insérer dans une équipe de développement, de faire mes preuves et ainsi de créer ma place au sein de cette équipe.

Malgré les difficultés rencontrées, qui, je pense, font partie intégrante d'un quotidien de développeur, cela ne fait pas de doute pour moi que mon avenir est dans cette voie.

Je suis très satisfait de mon stage et de la réussite du projet. J'ai découvert un milieu dans lequel je me plais, où j'ai envie de continuer à apprendre, découvrir et m'éclater (parce que si on rigole pas un peu ça n'a aucun sens).

Mon objectif est de persévérer et je reste curieux de voir où je pourrai bien arriver un jour.

Au final, je me suis vraiment éclaté à faire ce développement !



# PROJETS ANNEXES

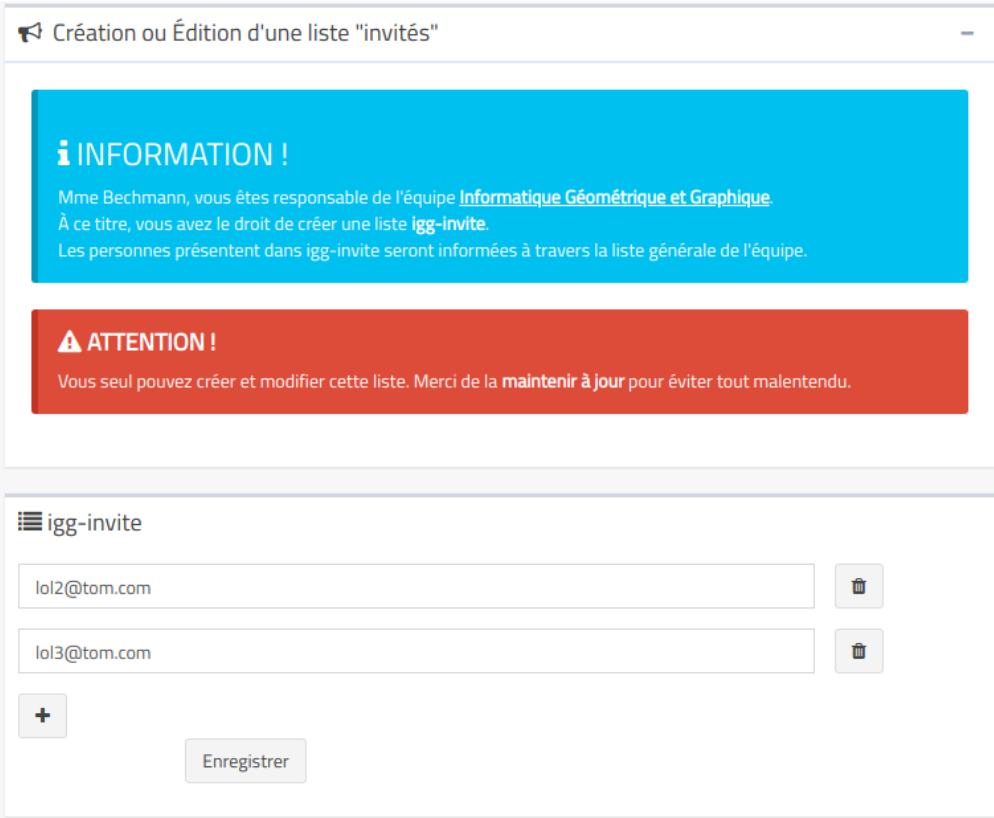
## LISTES INVITÉS

Une autre grande attente au sein de l'ICube concernant les listes, vient des responsables d'équipes qui ont besoin de pouvoir ajouter des personnes invitées à leur équipe. J'ai donc réalisé une interface seulement pour les personnes connectées en tant que responsable d'équipe leur permettant de créer une liste "invite" comme par exemple "igg-invite". L'interface leur permettra de renseigner l'adresse email des personnes, de pouvoir en ajouter autant qu'ils le souhaitent et bien évidemment de supprimer l'invité de la liste une fois sa mission accomplie.

Pour ce faire j'ai utilisé un petit bundle de Symfony "collection" qui permet de réaliser des tableaux JavaScript pour moduler facilement les saisis du formulaire.

A travers Sympa (souvenez-vous, c'est le robot de liste), Jean-Marc a fait en sorte que la liste "invite" soit dans le même alias mail que la liste générale de l'équipe. Cela permettra de contacter l'intégralité de l'équipe à travers le même alias.

Enfin, la liste "invite" sera évidemment consultable de la même façon que pour les autres listes à travers l'application.



Création ou Édition d'une liste "invités"

**INFORMATION !**  
Mme Bechmann, vous êtes responsable de l'équipe [Informatique Géométrique et Graphique](#).  
À ce titre, vous avez le droit de créer une liste **igg-invite**.  
Les personnes présentes dans igg-invite seront informées à travers la liste générale de l'équipe.

**ATTENTION !**  
Vous seul pouvez créer et modifier cette liste. Merci de la **maintenir à jour** pour éviter tout malentendu.

igg-invite

lol2@tom.com

lol3@tom.com

+

Enregistrer

Légende : interface de création / modification d'une liste « invité »

## TRADUCTION

Une description accompagne chaque liste pour que les membres du laboratoire sachent de quoi il s'agit véritablement. Il m'a suffi d'ajouter une colonne dans mon tableau de la liste des listes pour afficher les descriptions. Puis à travers un script Bash, j'ai auto-complété les descriptions pour que cela n'apparaisse pas vide dans l'interface. J'ai fait de même pour la version anglophone. Il suffira donc de renseigner manuellement les descriptions une à une pour que l'interface soit terminée.

## DÉBUGAGE

Evidemment, lorsqu'on se balade sur une appli web en mode dev, on rencontre parfois des bugs, d'où le fait de les corriger tout de suite pour ne pas les oublier.

## MIGRATION SYMFONY 2.8 VERS SYMFONY 3.4

Dans le but de garder le projet à jour en terme de technologie et de sécurité, nous avons commencé la migration du projet vers une version plus récente de Symfony.

La version 2.8 arrivant en fin de mise à jour, Aline est aussi obligé de migrer le projet. A savoir que migrer petit à petit vers une version ultérieure aide grandement le développeur à maintenir son projet à jour. Aline avait, depuis quelques mois, procédé à des modifications au sein du projet pour que la migration se fasse en douceur. L'idée était d'arriver sur la version actuelle du framework.

Cependant, elle souhaite attendre une version LTS (Long Term Support) de Symfony 4 afin d'être tranquille pendant un moment. Effectivement, migrer un projet d'une version à une autre demande beaucoup de temps et de débogage, ce qui est donc difficile pour une personne seule de développer et maintenir à côté de cela.

# DIVERS

---

Rendez-vous, conférences ou autre auxquels j'ai participé pendant mon stage :

- Rencontre avec les collègues du service informatique de l'université
- Mise en place d'une relation entre les services informatiques de l'ICube et de l'université avec l'ISIS de Strasbourg
- Entretien dans la start-up "Vazee" dans le cadre d'un éventuel projet de stage ou professionnel
- Mise en place d'un nouveau pare-feu dans l'entreprise Viwametal
- Visite des portes ouvertes de l'entreprise Alcatel
- Participation à une conférence sur le "blockchain"
- Entretien avec l'entreprise "ProcessMaker" dans le cadre d'un futur contrat professionnel
- Inscription aux unités d'études du CNAM (NFP135 et NFP136) dans le cadre de mon projet professionnel

# DICTIONNAIRE

Langage développeur	Langage humain
Apache	Serveur HTTP le plus populaire du web
Back-end	Les coulisses d'un programme
Base de données	Permet de stocker et retrouver l'intégralité d'informations brutes
Bootstrap	Bibliothèque graphique permettant de réaliser des sites web dynamiques
Branche	Projet de développement
Bundle	Paquet de contenus
Controller	Permet de gérer les demandes venant de l'utilisateur
Cron	Table de planification sous Linux
CSS	« Cascading Style Sheet », fichier où l'on configure le graphisme d'une page web
Dump	Permet d'exposer visuellement « l'intérieur » d'une donnée ou d'une fonction
Framework	Ensemble cohérent de composants logiciels permettant un développement plus efficace
Front-end	La partie visuelle d'un programme ou application
Git	Logiciel de gestion de versions décentralisées
Github	Site web d'hébergement et de gestion de développement de logiciels, utilise le logiciel Git
IDE	Ensemble d'outils qui permet d'augmenter la productivité des programmeurs
JavaScript	Langage de programmation de scripts principalement employé pour les sites web dynamiques
Linux	Système d'exploitation, logiciel libre créé par Linus Torvalds en 1991
Merge	Fusion de branches sur git
Méthode	Fonction membre d'une classe (en programmation orienté objet)
Mise en prod	Livraison de l'application aux utilisateurs finaux
MySQL	Système de gestion de bases de données relationnelles
PHP	« PHP Hypertext Preprocessor », langage de script open source utilisé pour les pages web
PhpStorm	IDE spécialement conçu pour le développement d'application en PHP
Screenshot	Capture l'image d'un écran
Script	Programme en langage interprété (ex. script PHP)
Serveur	Dispositif informatique qui offre des services à 1 ou plusieurs clients (ex. web)
Symfony	Framework MVC libre écrit en PHP
Sympa (robot de liste)	Programme permettant de générer des alias mail
Terminal	Point d'accès de communication entre l'homme et la machine
Versionning	Consiste à garder une version d'une entité (ex. version d'un logiciel)