



Patrice JAFFRÉZIC
Stagiaire développeur informatique

Compte-rendu de la période d'application en entreprise



28 rue François Arago
13005 Marseille

1 Remerciements

Un grand merci à l'équipe de km817 de m'avoir accueilli avec autant de simplicité et de gentillesse, et en particulier à Éric DEPIETROMARIA, mon maître de stage pour sa disponibilité, et Liz pour m'avoir fait partager son expérience.

Un grand merci également à Régine FISSEUX de l'A.F.P.A., notre responsable de formation pour son implication dans notre enseignement, à Dominique MULLER qui nous a dispensé la formation SQL, pour ses conseils avisés.

Merci aux membres de notre session pour leur franche camaraderie et leur solidarité qui ont contribué à nous faire vivre cette formation dans une ambiance studieuse mais néanmoins joyeuse, je ne vous oublierai pas.

Merci aussi à tous ceux qui de près ou de loin ont fait que cette formation soit possible.

Table des matières

1	Remerciements	2
2	Introduction.....	4
3	L'entreprise.....	5
3.1	km817.....	5
3.1.1	L'équipe	5
3.1.2	Historique	5
3.1.3	Les prestations.....	5
3.2	Déroulement du stage.....	8
3.2.1	L'accueil et l'intégration dans la société.....	8
3.2.2	La plateforme de développement.....	8
3.3	Le mode de travail au sein de km817.....	9
3.4	Les activités	10
4	Réalisations.....	11
4.1	AIRMES	11
4.2	OCTO Indonesian Furniture's	16
4.3	ERILIA.....	18
4.3.1	Fichiers XML.....	18
4.3.2	Pop-up	21
4.4	KM817	24
4.4.1	Programmation orientée objet	24
4.4.2	Analyse	32
4.4.3	Fichiers.....	34
5	Conclusion	36

2 Introduction

Pour conclure notre formation de développeur, nous devons effectuer un stage appelé « Période d'Application en Entreprise » : d'une durée de 12 semaines elle doit nous amener à confronter les connaissances acquises au cours de notre apprentissage au sein de l'A.F.P.A. et la réalité de notre métier.

J'ai souhaité effectuer cette étape importante de notre cursus chez km817 car le développement Web dynamique m'intéresse plus particulièrement : les tâches y sont variées et c'est un domaine en plein essor qui du fait de l'évolution des techniques, des exigences croissantes des clients et des internautes en matière de sécurité, est une activité en pleine professionnalisation.

3 L'entreprise

3.1 km817

km817 est une SARL au capital de 7622.50€, elle est gérée par 4 associés dont 2 ont le statut de salarié.

3.1.1 L'équipe

À Marseille :

- Un chef de projet, développeur (Éric DEPIETROMARIA – maître de stage et gérant de la société)
- Un technico-commercial (Benjamin LAYET)
- Un développeur informatique (Liz TRAMIER).

À Paris :

- Un chef de projet (Gaël MORIN).

3.1.2 Historique

Créée en 2002, km817 est composé d'expériences diverses:

Architectes, programmeur, graphiste.

Leur connaissance d'Internet depuis plus de dix ans ainsi que leur veille technologique leur ont permis de se positionner aujourd'hui en tant que partenaire privilégié dans le domaine de la publication de sites Internet et d'outils de gestion de données en ligne.

Ils ont développé un savoir-faire certain dans la chaîne de production, du cahier des charges à la réalisation en passant par le graphisme et l'hébergement.

Grâce à toutes ces compétences en installation et en maintenance ils peuvent aussi répondre aux questions spécifiques posées par les matériels et les logiciels.

3.1.3 Les prestations

3.1.3.1 Codage de sites avec interfaces d'administration:

Le codage est effectué de manière la plus efficace possible:

- Annotation systématique afin de pouvoir reprendre un projet et ce même plusieurs années après sa réalisation.
- Attention toute particulière à la qualité afin d'optimiser l'affichage des pages sur l'ensemble des navigateurs des systèmes d'exploitations courants.
- Logique de la chaîne de codage: la simplifier au maximum de manière à augmenter la vitesse d'affichage.
- Réalisation du code en vue de l'optimisation du référencement par les moteurs de recherche.

3.1.3.2 Plateforme Web

- Ayant développé et fait évoluer leur interface de gestion de contenu de sites, ils disposent aujourd'hui de nombreux modules satellites (module de news, module de liens, module de catalogue, module de vente en ligne, newsletter...) qui permet à leurs clients de mettre à jour leur site Internet comme bon leur semble.

- Chaque interface ou CMS est développée en fonction des besoins spécifiques avec les modules adaptés au type de profession et selon le type d'information à mettre en ligne. Cette interface est utilisable sans connaissances informatiques particulières.

3.1.3.3 Noms de domaines et hébergement:

km817 s'occupe également de l'enregistrement et du transfert de noms domaine dans toutes les extensions disponibles (fr, .eu, .com, .net, .org, .tv...).

La partie technique de la gestion serveur (mises à jour, gestion de la bande passante etc.) est assurée par un prestataire sous contrat et ce 24 h/24, 365 jours par an. Il réalise la partie administrative et l'appui technique de leurs prestations d'hébergement. Il n'y a donc qu'un unique interlocuteur.

Si les clients souhaitent gérer eux-mêmes leur espace d'hébergement, Km817 met à leur disposition une interface d'administration permettant de gérer l'ensemble de ses paramètres (boîtes mail, alias mail, redirections, groupe courriel, vérification de l'utilisation du disque et de la bande passante etc.).

3.1.3.4 Les fournisseurs

Depuis 2002 km817 a su s'entourer de prestataires de qualité. Ainsi disposent-ils aujourd'hui:

- D'un prestataire hébergement spécialisé sous contrat gérant le monitoring serveur et garantissant la bande passante ainsi que l'up time serveur (ou disponibilité du service, 99,95 %).
- De prestataires graphiques ayant une solide expérience dans la création Web mais aussi dans le Print.
- De fournisseurs de matériel de renom pour les installations de réseau et de parcs informatiques.
- D'un référenceur qualifié pour les positionnements de sites nécessitant un référencement très pointu ou très concurrentiel.
- De plusieurs partenaires pour la location des noms de domaines selon le domaine à enregistrer (zones Monde, Europe et France).

La chaîne d'édition du site Web est ainsi entièrement gérée par km817 : de la création de l'arborescence en passant par l'achat du nom de domaine jusqu'à l'installation du site et la gestion des courriers électroniques.

3.1.3.5 Références

(Liste non exhaustive)

L'ensemble de ses sites sont administrables par l'utilisateur final sans connaissance informatique particulière.

Pavillon de l'Arsenal

Centre d'information, de documentation et d'expositions d'architecture de la Ville de Paris.

Boutique avec paiement sécurisé.

<http://pavillon-arsenal.com>

Techniques employées

:php/html/mysql/javascript. Outil Massmailing*

Cité de l'Architecture

Cité de l'Architecture et du Patrimoine, Ministère de la Culture.

Documentation et expositions.

<http://www.citechailot.fr>

Techniques employées :

php/html/mysql/javascript/RSS.

Ateliers Lion

Yves Lion Architecte.

Présentation de l'agence, des projets et des réalisations.

<http://ateliers-lion.com>

Techniques employées :

php/flash/xml/mysql/javascript.

Agence Brossy

Agence d'architecture Brossy et Associés.

Présentation de l'agence, des projets et des réalisations.

<http://brossy.com>

Techniques employées :

php/flash/xml/mysql/javascript.

BVA

Institut de sondage.

<http://www.bva.fr>

Techniques employées :

php/mysql/javascript/vcf.

TABAS

Boutique en ligne, objets de créateurs.

<http://bureaudetabas.com>

Techniques employées :

php/html/mysql/javascript.

Société Générale

Collection d'art contemporain de la Société Générale.

<http://collectionsocietegenerale.com>

Techniques employées

:php/flash/xml/mysql/javascript.

AIRES

Agent de joueurs de basket.

<http://aires-basket.com>

Techniques employées :

php/html/mysql/javascript.

Adjoint à ce site : module de statistiques pour joueurs de basket.

Ateliers Acanthe

Architectes paysagistes.

<http://mutabilis-paysage.co>

Techniques employées :

php/flash/xml/mysql/javascript

Pain de Sucre

Swimwear et bodywear haut de gamme.

<http://paindesucre.com>

Techniques employées :

php/flash/xml/mysql.

Nouvelle version en cours de réalisation.

***Outil Massmailing**

Outil d'expédition de mails lié à une liste de diffusion, création de templates HTML, rapport d'expédition, gestion de contact, abonnement et désabonnement automatiques.

WorldWideNest

Blog familial interactif

<http://worldidenest.com>

Techniques employées :

php/flash/xml/mysql.

3.2 Déroulement du stage

3.2.1 L'accueil et l'intégration dans la société

Se sont déroulées de manière très simple et très rapide: le poste que j'ai occupé étant déjà en place: j'ai remplacé une stagiaire A.F.P.A. qui se présentait au titre la semaine même ou j'ai commencé mon stage.

Km817 étant une structure légère (3 personnes), nous occupons tous le même espace (mais sur des bureaux différents): c'est pratique et convivial.

L'entreprise est située dans un centre d'affaires (une ancienne usine réaménagée en bureaux mutualisés) et est occupé principalement par des architectes, architectes d'intérieur, graphistes, l'équipe de rédaction du gratuit « Ventilo », etc.

Les espaces dédiés à chaque entreprise sont disposés en mode paysager seulement séparés par des étagères à mi-hauteur, dans un esprit très « loft », ce qui favorise les échanges entre les différentes sociétés : de fait tout le monde se connaît et le trafic généré est parfois apporteur d'affaires.

Le centre possède également un jardin arboré ce qui contribue un peu plus à rendre l'atmosphère agréable.

3.2.2 La plateforme de développement

3.2.2.1 Matériel

J'ai travaillé sur un PC sous Windows XP (SP2), mais km817 possède un parc d'ordinateurs mixte PC/Mac, ainsi qu'un serveur sous Mac OSX Server (10.4 – UNIX). Le « debugging » est donc effectué sur ces différents systèmes ce qui permet d'être compatible avec les principaux navigateurs existants, Internet Explorer bien entendu, mais également Firefox et Safari.

3.2.2.2 Langages et logiciels utilisés:

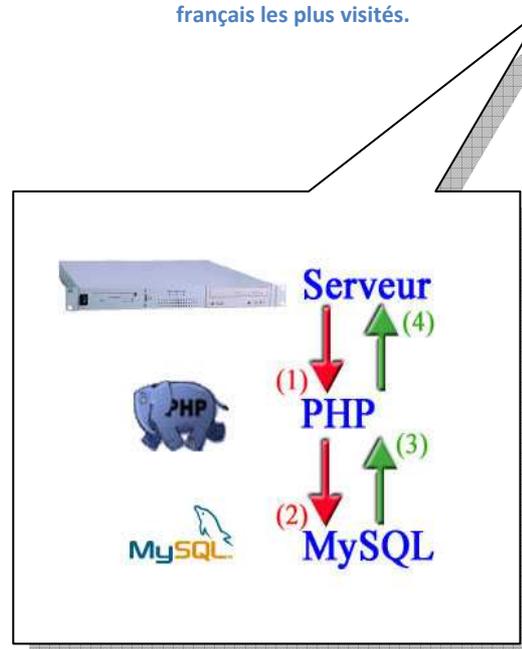
km817 utilise des langages Open Source: PHP, MySQL, JavaScript, HTML, XML, CSS, exception faite de l'Action Script d'Adobe FLASH : ceci afin de toujours posséder un langage mis à jour et ce de manière très réactive. En effet un langage Open Source est en perpétuel développement du fait d'une communauté de chercheurs et d'utilisateurs très active. Cela crée un cercle vertueux permettant de disposer de retours très rapides tant au niveau de l'évolution technologique qu'au niveau de la sécurité des programmes. Par ailleurs le problème d'acquisition de droits et de logiciels est ainsi résolu et n'est donc jamais répercuté sur la clientèle.

PHP, JavaScript, HTML, XML, CSS sont utilisés dans L'E.D.I. PHP Edit (2.8), Le SGBD utilisée est MySQL.

- **PHP**: est un langage de script orienté objet exécuté côté serveur. Les scripts sont écrits de manière simple puis intégrés au sein d'une page HTML. Il séduit aussi par son interface simplifiée avec les bases de données.
- **MySQL** est un Système de Gestion de Bases de Données (SGBD) fonctionnant sous Windows, Mac et Linux.



Figure 1 Le couple PHP/MySQL, associé à un serveur Web Apache, correspond à la configuration de 7/10ème des sites français les plus visités.



- **JavaScript** est un langage de script orienté objet exécuté côté client et qui s'intègre parfaitement aux pages HTML pour créer de petites animations ou interagir avec l'utilisateur
- Le **moteur PHP** installé sur le serveur et les Mac est MAMP.
- Le **moteur PHP** installé en local sur chaque PC est WAMP.
- **GIMP** est utilisé pour les petits travaux graphiques. GIMP (pour GNU'S Image Manipulation Program) est un logiciel Open Source considéré comme l'équivalent du célèbre Photoshop

3.3 Le mode de travail au sein de km817

Chacun travaille seul à son poste et en général sur un projet différent des autres.

Chaque poste est relié au serveur où les dossiers des différents projets sont stockés.

Pour être sûr de ne pas faire de dégâts irrémédiables sur le serveur en cas de mauvaise routine : la mésaventure est déjà arrivée...! (Avec une routine récursive), chacun à installé un serveur PHP en local sur son poste et ainsi teste ses scripts avant de les synchroniser avec le serveur d'exploitation.

Les instructions sur les tâches à réaliser me sont transmises oralement, je prends des notes et je demande un complément d'information ou des explications supplémentaires si besoin.

3.4 Les activités

Durant ce stage je n'ai pas eu à mener un projet de bout en bout, mais plutôt effectuer divers travaux en fonction des attentes des clients : en effet la période estivale à laquelle s'est déroulé le stage est la basse saison pour les lancements de projets, mais au final, cela couvre bien l'éventail des différentes activités du développeur :

- Algorithmes
- Analyse
- Base de données
- Fichiers
- Programmation orientée objet

Ce stage peut être divisé en deux parties :

- Une première partie de familiarisation avec l'environnement de développement Web où j'ai pu mettre en pratique mes connaissances en SQL, programmation procédurale et fonctionnelle, algorithmes, fichiers.
- Une seconde partie consacrée à la programmation objet, l'analyse et la conception d'une base de données associée à cette classe.

4 Réalisations

4.1 AIRMES

Airmes est une société qui vend de l'air médicalisé sous pression aux hôpitaux, cliniques, etc.

La demande

Cette société possède déjà un site internet statique en html, mais souhaite en rajouter l'interface graphique : Ce travail est confié à un graphiste qui débute son stage en même temps que moi.

Mais surtout de le rendre interactif, afin de mieux communiquer avec ses clients. Bien entendu une base de données est accolée au site, déjà conçue par km817 et adaptée aux besoins particuliers de ce client.

Tâches à réaliser

Deux fonctionnalités doivent être ajoutées :

Faire un formulaire de connexion sécurisée pour que ses clients aient accès à des documents qui leurs sont destinés en particulier.

Ajouter une fonction dans une interface d'administration pour que l'administrateur du site ait un accès à des « retours d'expériences », en fait des commentaires + des photos et /ou de la vidéo uploadé par ses clients sur le serveur.

Conception

La première question que je me suis posée : quels vont être les identifiants ? Finalement j'ai opté pour l'entrée de l'email et d'un mot de passe codé en MD5, en effet l'email est par principe unique donc en théorie il n'y aura pas de problèmes de doublons.

A la validation du formulaire de connexion, une recherche est effectuée sur la base pour en contrôler la présence concrète. Les champs sont envoyés dans la requête SQL par l'intermédiaire du tableau PHP « \$_POST » qui est une solution plus élégante mais aussi plus sûre que le « \$_GET » qui apparaît dans l'URL du navigateur du client.

Si c'est le cas l'utilisateur reçoit un message de bienvenue avec son nom et son prénom extrait de la table. Ensuite les documents qui lui sont destinés sont affichés après vérification que les susdits documents soient effectivement en ligne.

Si le nombre de documents de la liste dépasse 10, un objet « compteur de page » appartenant aux bibliothèques de scripts de km817 est appelé et affiche les pages par leur numéro avec des flèches droite et gauche qui permettent de naviguer de page en page ou directement en cliquant sur le n° de la page correspondante.

Si l'utilisateur sélectionne un des fichiers, une nouvelle requête SQL est adressée à la base et va rechercher le fichier, son chemin est généré et un entête de forçage au téléchargement est envoyé au navigateur pour que le fichier s'enregistre automatiquement sur le disque du poste client.

Si les identifiants ne sont pas reconnus, un message lui indique d'essayer de se connecter à nouveau ou de contacter l'administrateur du site.

Un autre cas est possible : je l'ai découvert en testant : si l'utilisateur, tente de se connecter mais sans avoir rempli les champs, un message lui enjoint alors de le faire. Ce message est déclenché par le passage d'un \$_GET (dans l'URL).

Enfin l'utilisateur doit pouvoir modifier ses identifiants s'il le souhaite, un bouton le lui propose sur le formulaire, il doit alors saisir ses nouveaux identifiants et un update de ses données est fait sur la base. L'identité du client est conservée tout au long de sa navigation sur le site grâce au tableau PHP \$_SESSION, il ne faut pas oublier de déclarer le démarrage de la session par un session_start() au début de chaque script (avant tout autre chose) sans quoi cela ne fonctionnerait pas.

Connexion Sécurisée et affichage des documents de l'utilisateur

```
1 <?php
2 #####
3 # Vérification du login et pass utilisateur #
4 #####
5 if ($_SESSION['id_client'] == '') { //si il n'y a pas de session ouverte
6
7     ## Page login
8     if (isset($_SESSION['POST']['login']) && isset($_SESSION['POST']['password'])) { //le client se connecte
9         $sql = 'SELECT id, nom, prenom
10             FROM client_client.'. $table .'
11             WHERE email LIKE "'.$_SESSION['POST']['login'].'"
12             AND pass LIKE ".md5($_SESSION['POST']['password'])."';
13         $query = QueryError($sql);
14         //bug_log($sql);
15         //déclare une session ou l'id_client = l' id de la table client
16         if (mysql_num_rows($query) > 0) { //si le résultat est positif
17             $result = mysql_fetch_array($query); //mise du résultat de la requête dans un tableau
18             $_SESSION['id_client'] = $result[0]; //affectation de l'id de la table à l'id de session
19         } else { //si mauvais identifiant
20             $message = 'Votre email et/ou votre mot de passe n\'ont pas été reconnus,
21             essayez de vous authentifier à nouveau ou contactez l\'administrateur du site.';
22         } //fin
23     }
24     if ($_GET['error']=='login') { //si les champs n'ont pas été remplis
25         $message = 'Remplissez les champs pour vous connecter.';
26     } //fin
27     //formulaire si champs non remplis
28     $contenu = $message.'
29     <form name="form" method="post" action="redirect.php">
30     <input type="hidden" name="loc" value="rapport" />
31     Votre Mseacute;l:
32     <input name="login" type="text" length=80 class="input"><br />
33     Votre mot de passe:
34     <input name="password" type="password" length=30 class="input"/>
35     <br /><br />
36     <input type="submit" value="Connexion" class="submit"/>
37     </form>;
38 } //fin
39
40 if ($_SESSION['id_client'] > 1) { //si la session existe
41     ## Page liste
42     require('".$chemin."/classes/CompteurPage.php'); //chemin du fichier contenant l'objet
43     if ($_SESSION['CompteurPage_'.$obj] == '') { //si le compteur n'est pas déjà ouvert
44         # OUVERTURE DE LA CLASS START
45         $CompteurPage = new CompteurPage(); //nouveau compteur
46         $CompteurPage->NewLimit($);
47     }
48 }
```

Figure 2 Identification utilisateur

```

38 ▼ if ($SESSION['id_client']>1) { //si la session existe
39     ### Page liste
40     require ("$_SERVER['$_script_name']/classes/CompteurPage.php"); //chemin du fichier contenant l'objet
41     if ($SESSION['CompteurPage_'.$obj] == '') { //si le compteur n'est pas déjà ouvert
42         # OUVERTURE DE LA CLASS START
43         $CompteurPage = new CompteurPage(); //nouveau compteur
44         $CompteurPage->NewLimit(3);
45         $new = true;
46     } else { //si on désinitialise le
47         $CompteurPage = unserialize($SESSION['CompteurPage_'.$obj]);
48         $new = false;
49     }
50     $cond_admin = "AND online=1";
51     //selection de toutes les occurrences [d] probables pour le client connecté
52     $sql =
53         "SELECT C.id FROM rapport_rapport'. $table.' C
54         INNER JOIN client_relation'. $table.' B ON B.id_fiche = C.id
55         WHERE B.id_client = '$SESSION[id_client]'. '$cond_admin';
56     $query = QueryError($sql);
57     $array_aff = array();
58     while($result = mysql_fetch_array($query)) { //tant qu'il y a des lignes le résultat est un tableau
59         array_push($array_aff,$result[0]); //mise des résultat de la première colonne dans un autre tableau
60     } // while
61     #####
62     # TAB COMPTEUR
63     #####
64     $CompteurPage->MakeTab($array_aff); //à initialiser uniquement si $new == true
65     #####
66     # SET PAGE
67     #####
68     if (isset($_GET['page']))
69     {
70         $CompteurPage->Page($_GET['page']);
71     }
72     #####
73     # DONNEES PAGE
74     #####
75     $start=$CompteurPage->start;
76     $nb=$CompteurPage->nb;
77     #####
78     # Requête d'affichage des documents auquel a droit l'utilisateur #
79     #####
80     $sql = 'SELECT prenom , nom FROM client_client'. $table.' WHERE id = "'.$SESSION[id_client].'";
81     $query=QueryError($sql);

```

Figure 3 Vérification des documents en ligne de l'utilisateur avec appel de l'objet CompteurPage

```

77     #####
78     # Requête d'affichage des documents auquel a droit l'utilisateur #
79     #####
80     $sql = 'SELECT prenom , nom FROM client_client'. $table.' WHERE id = "'.$SESSION[id_client].'";
81     $query=QueryError($sql);
82     $result=mysql_fetch_array($query);
83     $accueil = 'Bienvenue M. '.ucfirst($result[0]).' '.strtoupper($result[1]).', voici vos documents: '.<br>';
84     //exécution de la requête et affichage des docs auquel a droit le client
85     $sql =
86         "SELECT A.titre,A.presentation,A.description,D.url,C.id FROM client_relation'. $table.' B
87         INNER JOIN rapport_rapport'. $table.' C ON B.id_fiche = C.id
88         INNER JOIN rapport_rapport_fr'. $table.' A ON A.id_rapport = C.id
89         INNER JOIN rapport_rapport_fichier'. $table.' D ON D.id_rapport = A.id_rapport
90         WHERE C.online >0 AND B.id_client = '$SESSION[id_client]'. ' LIMIT '$CompteurPage->start.', '$CompteurPage->limit;
91     $query=QueryError($sql);
92     //bug_or(on.$sql.$table);
93     while($affichedoc = mysql_fetch_array($query)){
94         $produits .= '
95         <div align="left">
96             <form style="float:right;" action="download.php" name="form_'.$affichedoc[4].'" method="post" target="_self" align="left">
97                 <input type="hidden" name="id" value="'.$affichedoc[4].'" />
98                 <input type="submit" value="Téléchargez" id="download" class="submit"/>
99             </form>
100             >>'. $affichedoc[0]. '
101             <br />
102             '. $affichedoc[1]. '
103             '. $affichedoc[2]. '
104         </div>
105         <hr style="border:1px solid black"/>
106         ';
107     }
108     $navigation2 = '
109     <a href="#" onclick="document.getElementById(\'modifier\').style.display=\'block\';">
110         <input type="hidden" name="modidentif">Modifier vos identifiants ?</a>
111     <br><br>
112     <form name="form" id="modifier" method="post" action="redirect.php" style="display:none;>
113         <input type="hidden" name="loc" value="rapport" />
114     Votre Mseacute;l:<br />
115     <input name="login" type="text" length=30 class="input"><br /><br />
116     Votre mot de passe:<br />
117     <input name="password" type="password" length="30" class="input"/>
118     <br /><br />
119     <input type="submit" value="Connexion" class="submit"/>
120     <input type="hidden" name="action" value="modifier_user" />

```

Figure 4 SQL et affichage des documents de l'utilisateur

```

94     $produits .= '
95     <div align="left">
96         <form style="float:right;" action="download.php" name="form_'.$affichedoc[4].'" method="post" target="_self" align="left">
97             <input type="hidden" name="id" value="'.$affichedoc[4].'" />
98             <input type="submit" value="Téléchargez" id="download" class="submit"/>
99         </form>
100         >>'.$affichedoc[0].'
101         <br />
102         '.$affichedoc[1].'
103         '.$affichedoc[2].'
104     </div>
105     <hr style="border:1px solid black"/>
106     ';
107
108     $navigation2 = '
109     <a href="#" onclick="document.getElementById(\'modifier\').style.display=\'block\';">
110     <input type="hidden" name="modidentif">Modifier vos identifiants ?</a>
111     <br><br>
112     <form name="form" id="modifier" method="post" action="redirect.php" style="display:none;">
113     <input type="hidden" name="loc" value="rapport" />
114     Votre Mseacute;l:<br />
115     <input name="login" type="text" length=30 class="input"><br /><br />
116     Votre mot de passe:<br />
117     <input name="password" type="password" length="30" class="input"/>
118     <br /><br />
119     <input type="submit" value="Connexion" class="submit"/>
120     <input type="hidden" name="action" value="modifier_user" />
121     <input type="hidden" name="loc" value="rapport" />
122     <input type="submit" value="Modifier" class="submit" />
123     </form>;
124
125     if ($CompteurPage->nb_page > 1){
126     $navigation = "<p style='clear:both;border-top:1px solid #666699;border-bottom:1px solid #666699;padding:10px;text-align:right;'>".$CompteurPage
127     ->index().$CompteurPage->precedent().$CompteurPage->suivant()."<p>";
128     }
129     $contenu = $navigation.$produits;
130     # FERMETURE DE LA CLASSE
131     $SESSION['CompteurPage_'.$obj] = $s;
132     $s = serialize($CompteurPage);
133     } // fin de session client ouverte
134     # remise à 0 du formulaire #
135     unset($SESSION['POST']);
136     ?>

```

Figure 5 Sérialisation de l'objet CompteurPage pour pouvoir l'utiliser dans un autre script, le tableau \$_POST est effacé.

Forçage au téléchargement

```

4  * @copyright 19/06/2007
5  */
6  session_start();
7  require('conf.inc.php');
8  require(ROOT.'tek/tek.inc.php');
9  nb_connect('');
10 $obj = 'rapport';
11 if ($SESSION['id_client']>1) { //session client existe
12 //vérification id_client/id_rapport
13 $sql = 'SELECT id_client, id_rapport FROM client_relation'.$table.', rapport_rapport_fichier'.$table.'
14 WHERE id_client = '.$SESSION['id_client'].'
15 AND id_rapport = '.$POST['id'];
16 $query = QueryError($sql);
17 if (mysql_num_rows($query)>0) { //si il y a plus d'un résultat
18 //requete sur le fichier à télécharger avec l'id du $_POST
19 if (!empty($POST['id'])) { //si l'id n'est pas vide
20 $sql = 'SELECT C.id, D.url FROM rapport_rapport'.$table.' C
21 INNER JOIN rapport_rapport_fichier'.$table.' D ON D.id_rapport = C.id
22 WHERE C.id = '.$POST['id'].' AND C.online=1';
23 $query1 = QueryError($sql);
24 if (mysql_num_rows($query1)>0) { //si il y a plus d'un résultat
25 $affichedoc = mysql_fetch_array($query1); // mise du résultat de la requête en tableau
26 //chemin du fichier
27 $chemin = 'data/'.$obj.'/'.$obj.'_'.$obj.'/'.$affichedoc[0].'/fichier/'.$affichedoc[1];
28 header("Content-type: application/force-download");
29 header("Content-Length: ".filesize($chemin));
30 header("Content-Disposition: attachment; filename=".basename($chemin));
31 readfile($chemin);
32 } // if
33 } else {
34 header("Location: ".$SERVER['HTTP_REFERER']);
35 } // if else
36 } else {
37 header("Location: ".$SERVER['HTTP_REFERER']);
38 } // if else
39 } else {
40 header("Location: ".$SERVER['HTTP_REFERER']);
41 } // if else
42 ?>

```

Figure 6 Forcer un document à se télécharger sur le poste client

Gestion des retours d'expérience par l'administrateur

```

1 <?php
2 ## Page liste
3 require("../classes/CompteurPage.php");
4 if ($SESSION['CompteurPage_'.$obj] == ''){//s'il n'y a pas compteur page dans la session
5 #OUVERTURE DE LA CLASS START
6 $CompteurPage = new CompteurPage(); //instanciation d'un nouvel objet compteur
7 $CompteurPage->NewLimit(2); //nb d'affichage d'occurrences dans le compteur
8 $new = true; //variable $new passée à true
9 }else{
10 $CompteurPage = unserialize($SESSION['CompteurPage_'.$obj]); //sion désérialisation du compteur dans la mémoire
11 $new = false; //variable $new passée à false
12 }
13 //selection de toutes les occurrences en ligne
14 $sql = 'SELECT A.id, B.titre, B.presentation, B.description, A.photo, C.url FROM retour_retour_fr'. $table.' B
15 INNER JOIN retour_retour'. $table.' A ON A.id = B.id_retour
16 LEFT JOIN retour_retour_photo'. $table.' C ON B.id_retour = C.id_retour
17 WHERE A.online = 1';
18 $query = QueryError($sql); //envoi de la requête sql au serveur
19 $array_aff = array(); //nouveau tableau
20 while($result = mysql_fetch_array($query)){ //mise des résultat dans un tableau
21 array_push($array_aff, $result[0]); //mise des '0' dans un tableau
22 } //while
23 # TAB COMPTEUR
24 #####
25 $CompteurPage->MakeTab($array_aff); //a initialiser uniquement si $new == true
26 #####
27 # SET PAGE
28 #####
29 if (isset($_GET['page'])){//s'il y a des pages
30 {
31 $CompteurPage->Page($_GET['page']); //affectation de la page dans l'objet compteur
32 }
33 #####
34 # DONNEES PAGE
35 #####
36 $start=$CompteurPage->start; //affectation de la première page au compteur
37 $nb=$CompteurPage->nb; //affectation du nombre de page dans le compteur
38 //Affichage des résultats de la requête
39 $sql = 'SELECT A.id, B.titre, B.presentation, B.description, A.photo, C.url FROM retour_retour_fr'. $table.' B
40 INNER JOIN retour_retour'. $table.' A ON A.id = B.id_retour
41 LEFT JOIN retour_retour_photo'. $table.' C ON B.id_retour = C.id_retour
42 WHERE A.online = 1
43 LIMIT '.$CompteurPage->start.', '.$CompteurPage->limit;
44 $query = QueryError($sql);

```

Figure 7 Appel du « CompteurPage », SQL des retours en ligne, SQL des retours

```

29 if (isset($_GET['page'])){//s'il y a des pages
30 {
31 $CompteurPage->Page($_GET['page']); //affectation de la page dans l'objet compteur
32 }
33 #####
34 # DONNEES PAGE
35 #####
36 $start=$CompteurPage->start; //affectation de la première page au compteur
37 $nb=$CompteurPage->nb; //affectation du nombre de page dans le compteur
38 //Affichage des résultats de la requête
39 $sql = 'SELECT A.id, B.titre, B.presentation, B.description, A.photo, C.url FROM retour_retour_fr'. $table.' B
40 INNER JOIN retour_retour'. $table.' A ON A.id = B.id_retour
41 LEFT JOIN retour_retour_photo'. $table.' C ON B.id_retour = C.id_retour
42 WHERE A.online = 1
43 LIMIT '.$CompteurPage->start.', '.$CompteurPage->limit;
44 $query = QueryError($sql);
45 while($affichedoc = mysql_fetch_array($query)){
46 $produits.= '<p align="left"><UL type="circle">
47 <LI>'. $affichedoc[1].'
48 <h6 align="left">'. $affichedoc[2]. '</h6>
49 <h6 align="right">'. $affichedoc[3]. '</h6>
50 <form action="download.php" name="form_'.$affichedoc[4].'" method="post" target="_self" align="left">
51 .DATAROOT.'.$obj.'/'.'.$obj.'/'.'.$obj.'/'.'.$affichedoc[4].'/fichier/'.$affichedoc[5].'.'.'
52 <input type="hidden" name="id" value="'.$affichedoc[8].'" />
53 <input type="submit" value="Voir les photos?" id="voirgalerie" />
54 </form>
55 </LI></UL>
56 </p><hr>';
57 } //while
58 if ($CompteurPage->nb_page > 1){ //s'il y a plus d'une page dans le compteur affiche toi comme ça
59 $navigation = "<p style='clear:both;border-top:1px solid #666699;border-bottom:1px solid #666699;padding:10px;text-align:right;'>".
60 $CompteurPage->index().$CompteurPage->precedent().$CompteurPage->suivant()."<p>";
61 }
62 $contenu = '<div id="Pres" style="">
63 <div style="width:460px;">
64 . $accueil.'
65 . $navigation.'
66 . $produits.'
67 </div>
68 </div>';
69 # FERMETURE DE LA CLASS
70 $SESSION['CompteurPage_'.$obj] = $s;
71 $s = serialize($CompteurPage); //srialization du compteur
72 ?>

```

Figure 8 Affichage des retour et sérialisation de l'objet « CompteurPage »

Résultat

<http://www.airmes.fr/>

offline

4.2 OCTO Indonesian Furniture's

Est un des tous premiers clients de km817, c'est une société qui fabrique des meubles en Indonésie et les vend en Europe.

La demande

Le formulaire de contact du site avait besoin d'un petit toilettage : une image qui sert à valider le formulaire afin qu'un robot ne puisse pas s'en servir pour envoyer des emails portait à confusion, il fallait la changer.

Ensuite le formulaire devait être vérifié avant l'envoi.

Tâches à réaliser

Une nouvelle image

Faire une vérification du formulaire

Conception

J'ai donc fait une nouvelle image à l'aide du logiciel GIMP.

Pour la vérification du formulaire le JavaScript s'est imposé de lui-même puisqu'il ne doit être vérifié que côté client.

Une fonction pour vérifier la véracité de l'email à été nécessaire car les gens d'OCTO doivent absolument pouvoir recontacter leurs prospects, il est donc indispensable que l'email laissé dans le formulaire soit valide.

C'est à cette occasion que j'ai découvert les expressions régulières très pratique et très puissante pour ce type de cas où l'on doit retrouver une chaîne de caractère avec différentes séquences.

Les autres vérifications ne portaient que sur le fait que les champs ne soient pas vides pour ne pas être trop contraignant vis-à-vis des clients potentiels contactant OCTO, mis à part bien entendu le champ où doit être reproduite la chaîne de caractères inscrite sur l'image.

Remarque : Le site OCTO étant bilingue, la modification à été faite dans les deux langues.

```

1  function isMail(txtMail) {
2      var regMail=new RegExp("[0-9a-z, _-]+@[0-9a-z, -]{2,}[.]{1}[a-z]{2,5}?", "i");
3      return regMail.test(txtMail);
4  }
5  function validerFormulaire(formulaire) {
6      if (formulaire.nom.value==""){
7          alert("Please, enter your first name in this field");
8          formulaire.nom.focus();
9          formulaire.nom.style.backgroundColor='#FFC77E';
10         return false;
11     }
12     if (formulaire.societe.value==""){
13         alert("Please, enter the name of your company in this field");
14         formulaire.societe.focus();
15         formulaire.societe.style.backgroundColor='#FFC77E';
16         return false;
17     }
18     if (!isMail(formulaire.email.value)){
19         alert("Please, enter a valid email adress if you want to receive an answer");
20         formulaire.email.focus();
21         formulaire.email.style.backgroundColor='#FFC77E';
22         return false;
23     }
24     if (formulaire.tel.value==""){
25         alert("Please, enter your phone number in this field");
26         formulaire.tel.focus();
27         formulaire.tel.style.backgroundColor='#FFC77E';
28         return false;
29     }
30     if (formulaire.code.value!="BBASTJ"){
31         alert("Wrong code, try again!");
32         formulaire.code.focus();
33         formulaire.code.style.backgroundColor='#FFC77E';
34         return false;
35     } else {
36         alert("The form is validate");
37         return true;
38     }
39 }
40

```

Figure 6 La vérification JavaScript(en anglais)

Résultat

<http://www.indonesiafurnitures.com/fr/contacts.html>

online

<http://www.indonesiafurnitures.com/uk/contacts.html>

online

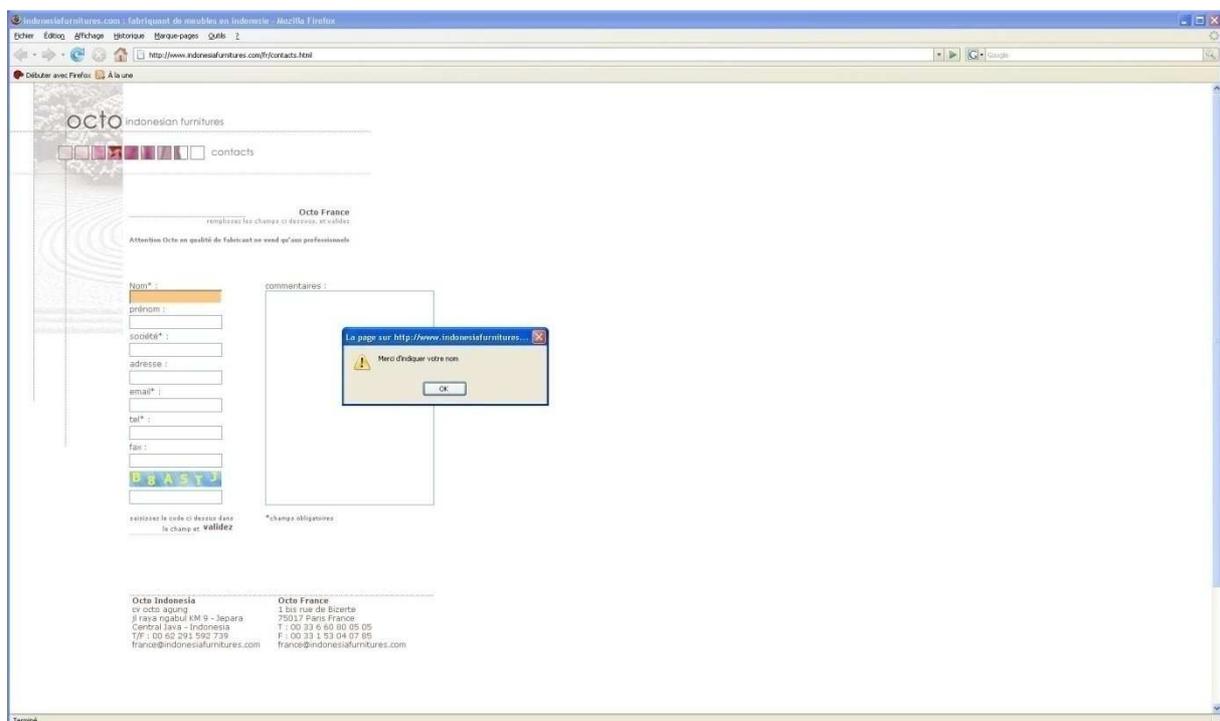


Figure 7 Le formulaire en ligne (en français)

4.3 ERILIA

Est une société de gestion d'offices et d'HLM

4.3.1 Fichiers XML

La demande

Il à été demandé à km817 de réaliser une carte de France dynamique dans laquelle on puisse zoomer et avoir accès à un certain nombre de villes à l'intérieur desquelles on accède à des points qui contiennent eux-mêmes un certain nombre d'informations et notamment des photos si elles existent. Il fallait donc parser des fichiers XML disposant de ces informations dans l'Action Script

Tâche à réaliser

Un fichier XML par carte :

France, Marseille, Marseille Michelet, Nice, Nîmes, St Étienne, Vallauris.

Conception

Cette carte à été réalisée en Action Script par Éric DEPIETROMARIA

Pour ma part je me suis occupé de générer les fichiers XML, non sans avoir auparavant entré pour chaque point ses coordonnées en pixel. La tâche était énorme puisque cette carte comprenait plusieurs sous-cartes et plusieurs centaines de points, tâche dont je me suis acquitté une nouvelle fois grâce à GIMP. Long et fastidieux mais indispensable.

Ces informations sont ensuite exportées sous forme de « .CSV. »

J'ai donc eu à coder l'algorithme qui inclut ses informations dans un fichier XML. L'algorithme à été codé sous PHP, car contrairement à ce que l'on pourrait penser PHP n'est pas qu'un langage pour le Web puisque qu'il peut aussi lire et écrire dans des fichiers.

Je profite de l'occasion qui m'est donnée pour dire que PHP sait aussi générer des images, mais je n'ai pas eu l'occasion de le tester durant ce stage.

```

1 432;mon;1736;1957;7:30;Aigues-Mortes;aigues-mortes;Logis du Bosquet;36;1985;PLA;;0
2 208;mar;1968;1948;6:13;Aix-en-Provence;aix-en-provence;La Constance;6;2001;Autres;0
3 206;mar;1968;1948;6:13;Aix-en-Provence;aix-en-provence;Le Botticelli;58;2000;PLI/Autres;1
4 205;mar;1968;1948;6:13;Aix-en-Provence;aix-en-provence;Le Sénéque;144;1996;PLA;PLI/Autres;1
5 207;mar;1968;1948;6:13;Aix-en-Provence;aix-en-provence;Le Jules Verne;172;2000;Autres;0
6 563;mar;1968;1948;6:13;Aix-en-Provence;aix-en-provence;James Chasserlaud - Résidence Etudiants;136;1990;;;0
7 20;aja;2371;2300;5:2a;Ajaccio;ajaccio;A Mandarina 1, 2 & 3;330;1986/89;PLA;;1
8 18;aja;2371;2300;5:2a;Ajaccio;ajaccio;Livrelli;30;1977;;HLMO;0
9 17;aja;2371;2300;5:2a;Ajaccio;ajaccio;Les Salines 1 & 2;810;1966/70;;HLMO;0
10 23;aja;2371;2300;5:2a;Ajaccio;ajaccio;Les Hauts de Petra di Mare;148;2004/2005;PLUS;;0
11 21;aja;2371;2300;5:2a;Ajaccio;ajaccio;Monte e Mare;91;1987;PLA;;0
12 22;aja;2371;2300;5:2a;Ajaccio;ajaccio;Petra di Mare 1, 2 & 3;342;1996/2000;PLA;;1
13 25;aja;2371;2300;5:2a;Ajaccio;ajaccio;A Seta;71;2004;PLUS;;0
14 24;aja;2371;2300;5:2a;Ajaccio;ajaccio;A Licciona 1 & 2;98;2002/2007;PLA/ PLUS;;1
15 552;aja;2371;2300;5:2a;Ajaccio;ajaccio;La Miséricorde - Foyer Infirmer;37;1992;;;0
16 19;aja;2371;2300;5:2a;Ajaccio;ajaccio;Les Hauts du Vazzio;84;1980;PLA;;1
17 30;bas;2492;2272;5:2b;Aléria;aleria;De Diane;36;1987;PLA ;;0
18 536;ri1;2089;1279;3:74;Annenasse;annemasse;Le Tetras 1 & 2;117;2002;PLUS;PLS;1
19 537;ri1;2089;1279;3:74;Annenasse;annemasse;Résidence Marie Laurencin;50;2007;PLUS;0
20 100;nic;2303;1884;6:6;Antibes;antibes;L'Amarante;84;2001;PLA;Autres/ PLI;0
21 97;nic;2303;1884;6:6;Antibes;antibes;Les Nymphéas;69;1995;PLA;;1
22 98;nic;2303;1884;6:6;Antibes;antibes;Les Stréitizias;93;1996;PLA;;1
23 96;nic;2303;1884;6:6;Antibes;antibes;Les Bleuets;45;1989;PLA;;0
24 99;nic;2303;1884;6:6;Antibes;antibes;Les Albizzias;77;1997;PLA;;0
25 95;nic;2303;1884;6:6;Antibes;antibes;Le Logis de Saint-Claude;231;1976;;HLMO;0
26 101;nic;2303;1884;6:6;Antibes;antibes;Square du Lys;37;2002;PLUS;;0
27 180;ist;1826;1895;6:13;Arles;arles;Souleiado;170;1968/89;;HLMO/Autres;0
28 181;ist;1826;1895;6:13;Arles;arles;Les Peupliers;519;1974;;HLMO/.L.M.;0
29 182;ist;1826;1895;6:13;Arles;arles;Les Flamants Roses;70;1994;PLA;;0
30 544;mar;1991;2002;7:13;Aubagne;aubagne;Les Ciboulettes;18;1983;;;0
31 213;mar;1991;2002;7:13;Aubagne;aubagne;Les Pernettes;46;1995;PLA;;0
32 212;mar;1991;2002;7:13;Aubagne;aubagne;La Farandole;74;1986;PLA;;1
33 215;mar;1991;2002;7:13;Aubagne;aubagne;Le Savoie;132;2004;PLUS;;0
34 211;mar;1991;2002;7:13;Aubagne;aubagne;Les Logis des Bartavelles ;120;1983;PLA;;0
35 214;mar;1991;2002;7:13;Aubagne;aubagne;Les Romarins 2;20;1995;PLA;;0
36 209;mar;1991;2002;7:13;Aubagne;aubagne;Le Charrel;992;1973;PLR;1
37 210;mar;1991;2002;7:13;Aubagne;aubagne;Les Ciboulettes;6;1983;Autres;0
38 405;avi;1871;1846;6:84;Avignon;avignon;Le Corail;32;2005;PLA;;0
39 421;avi;1871;1846;6:84;Avignon;avignon;Les Treilles Blanches;156;2005;PLA;;0
40 406;avi;1871;1846;6:84;Avignon;avignon;L'Emeraude;34;2005;PLA;;0
41 419;avi;1871;1846;6:84;Avignon;avignon;Le Saint-Ruf 2;18;2005;PLA;;0
42 582;avi;1871;1846;6:84;Avignon;avignon;Réa. Etud. Laugier - Foyer Etudiants;86;2005;;;0
43 409;avi;1871;1846;6:84;Avignon;avignon;Les Grands Caméas;536;2005;PLA;;0
    
```

Figure 8 Fichier original en csv

```

1 <?php
2 /**
3  * Génération du fichier xml pour la carte de france
4  * @version $Id$
5  * @copyright 2007
6  */
7 $fichierLecture = "France2XML.csv" ; //attribution des noms des fichier à des variables
8 $fichierEcriture = "best_xml_france.xml";
9 $headerxml = '<?xml version="1.0" encoding="UTF-8" ?>' . "\n" . '<main titre="France" height="1570" width="1688">' . "\n";//entete
10 $footerxml = '</main>');//ped
11 file_put_contents($fichierEcriture, $headerxml, FILE_APPEND);
12 $dataFile = fopen($fichierLecture, "r"); //ouverture du fichier en lecture
13
14 if ($dataFile) { //si il y a des lignes
15     unset($donnees2xml, $ville_prec); //initialisation des variables
16     $villes = array(); //tableau des villes
17     while (!feof($dataFile)) { // tant que l'on est pas à la fin des lignes
18         $buffer = fgets($dataFile, 4096); //mise des résultats dans un buffer
19         $resultat = explode(";", $buffer); //mise du buffer dans le tableau
20         if ($resultat[6] != $ville_prec) { //si la ville est différente de la ville de la boucle précédente.
21             if ($ville_prec != '') { //si la ville précédente existe (à la boucle)
22                 $donnees2xml .= '</point>' . "\n"; //on ferme le point ouvert pour la ville précédente
23             }
24             $donnees2xml .= '<point Xx="' . ($resultat[3]-45) . '" Yy="' . ($resultat[3]-65) . '" priorite="0" nom="' . $resultat[6] . '" Type="point" og="' .
25             $resultat[1] . '" dpt="' . $resultat[5] . '" commune="' . $resultat[6] . '">' . "\n";//écriture des points
26             $villes[$resultat[6]] = array(); //la donnée "nom de la ville" est un tableau dans le tableau ville
27             //if
28             $ville_prec = $resultat[6] //affectation de la ville précédente
29             $donnees2xml .= '<item id="' . $resultat[0] . '" nom="' . $resultat[8] . '" nb_logement="' . $resultat[9] . '" annee="' . $resultat[10] . '" photo="' .
30             trim($resultat[13]) . '"></item>' . "\n";//écriture des attribut de chaque point si ils existent
31             array_push($villes[$ville_prec], $resultat[8]); //dans les cases de villes on rajoute les immeubles de la ville
32         } //while
33     } //après la dernière boucle on ferme aussi le dernier point
34     $donnees2xml .= '</point>' . "\n"; //après la dernière boucle on ferme aussi le dernier point
35     //echo $headerxml.$donnees2xml.$footerxml; // et on affiche
36     file_put_contents($fichierEcriture, $donnees2xml, FILE_APPEND); //ouverture, écriture, fermeture du fichier en écriture (données)
37     file_put_contents($fichierEcriture, $footerxml, FILE_APPEND); //ouverture, écriture, fermeture du fichier en écriture (le pied)
38
39     fclose($dataFile); //fermeture du fichier
40 } else { //si le fichier n'a pu être ouvert
41     die("fopen failed for $filename");
42 } //else
43 ?>
    
```

Figure 9 Algorithme codé en PHP

Résultat

<http://www.erilia.fr>

offline

```
1 </main> <?xml version="1.0" encoding="UTF-8" ?>
2 <main titre="France" height="1570" width="1688">
3 <point Xx="1691" Yy="1892" priorite="0" nom="Aigues-Mortes" Type="point" cg="mon" dpt="30" commune="Aigues-Mortes">
4 <item id="432" nom="Logis du Bosquet" nb_logement="36" annee="1985" photo="0"></item>
5 </point>
6 <point Xx="1923" Yy="1883" priorite="0" nom="Aix-en-Provence" Type="point" cg="mar" dpt="13" commune="Aix-en-Provence">
7 <item id="208" nom="La Constance" nb_logement="6" annee="2001" photo="0"></item>
8 <item id="206" nom="Le Botticelli" nb_logement="58" annee="2000" photo="1"></item>
9 <item id="205" nom="Le Senéque" nb_logement="144" annee="1996" photo="1"></item>
10 <item id="207" nom="Le Jules Verne" nb_logement="172" annee="2000" photo="0"></item>
11 <item id="563" nom="James Chasseriaud - Résidence Etudiants" nb_logement="136" annee="1990" photo="0"></item>
12 </point>
13 <point Xx="2326" Yy="2235" priorite="0" nom="Ajaccio" Type="point" cg="aja" dpt="2a" commune="Ajaccio">
14 <item id="20" nom="A Mandarina 1, 2 & 3" nb_logement="330" annee="1986/89" photo="1"></item>
15 <item id="18" nom="Livrelli" nb_logement="30" annee="1977" photo="0"></item>
16 <item id="17" nom="Les Salines 1 & 2" nb_logement="810" annee="1966/70" photo="0"></item>
17 <item id="23" nom="Les Hauts de Petra di Mare" nb_logement="148" annee="2004/2005" photo="0"></item>
18 <item id="21" nom="Monte e Mare" nb_logement="91" annee="1987" photo="0"></item>
19 <item id="22" nom="Petra di Mare 1, 2 & 3" nb_logement="342" annee="1996/2000" photo="1"></item>
20 <item id="25" nom="A Sêta" nb_logement="71" annee="2004" photo="0"></item>
21 <item id="24" nom="A Licciona 1 & 2" nb_logement="98" annee="2002/2007" photo="1"></item>
22 <item id="22" nom="La Misericorde - foyer infirmier" nb_logement="37" annee="1992" photo="0"></item>
23 <item id="19" nom="Les Hauts du Vazzio" nb_logement="84" annee="1980" photo="1"></item>
24 </point>
25 <point Xx="2447" Yy="2207" priorite="0" nom="Aléria" Type="point" cg="bas" dpt="2b" commune="Aléria">
26 <item id="30" nom="De Diane" nb_logement="36" annee="1987" photo="0"></item>
27 </point>
28 <point Xx="2044" Yy="1214" priorite="0" nom="Annemasse" Type="point" cg="ril" dpt="74" commune="Annemasse">
29 <item id="536" nom="Le Tétris 1 & 2" nb_logement="117" annee="2002" photo="1"></item>
30 <item id="537" nom="Résidence Marie Laurencin" nb_logement="50" annee="2007" photo="0"></item>
31 </point>
32 <point Xx="2258" Yy="1819" priorite="0" nom="Antibes" Type="point" cg="nic" dpt="6" commune="Antibes">
33 <item id="100" nom="L'Amarante" nb_logement="84" annee="2001" photo="0"></item>
34 <item id="97" nom="Les Nymphéas" nb_logement="69" annee="1995" photo="1"></item>
35 <item id="98" nom="Les Strélitziass" nb_logement="93" annee="1996" photo="1"></item>
36 <item id="96" nom="Les Bleuets" nb_logement="45" annee="1989" photo="0"></item>
37 <item id="99" nom="Les Albizzias" nb_logement="77" annee="1997" photo="0"></item>
38 <item id="95" nom="Le Logis de Saint-Claude" nb_logement="231" annee="1976" photo="0"></item>
39 <item id="101" nom="Square du Lys" nb_logement="37" annee="2002" photo="0"></item>
40 </point>
41 <point Xx="1781" Yy="1830" priorite="0" nom="Arles" Type="point" cg="ist" dpt="13" commune="Arles">
42 <item id="180" nom="Soulelado" nb_logement="170" annee="1968/89" photo="0"></item>
43 <item id="181" nom="Les Peuliers" nb_logement="519" annee="1974" photo="0"></item>
```

Figure 10 le fichier XML

4.3.2 Pop-up

La demande

ERILIA a demandé à ce que des photos contenues dans un dossier s'affichent dans une fenêtre pop-up, avec en légende : la situation géographique, le nom du programme immobilier, le nombre de logements qui le compose, et l'année de livraison.

Tache à réaliser

Une fenêtre pop-up

Conception

La contrainte sur ce travail était que si le poste client devait recharger les photos à partir du serveur à chaque nouvelle demande, cela aurait été d'une lenteur affligeante pour l'internaute.

Pour palier à ce problème la solution a été de passer les photos côté poste client dans un script JavaScript afin de les faire afficher rapidement.

Les serveurs UNIX créent des fichiers invisibles (dont le nom est précédé d'un point) pour chaque fichier contenu dans le dossier et un autre fichier invisible dont le nom est précédé de 2 points qui désigne lui le dossier parent du dossier dans lequel il est contenu, il convenait donc d'éliminer ces fichiers inutiles avant de les charger sur le poste client.

La fenêtre ne devait pas s'ouvrir pour chaque photo, mais les afficher une par une dans le pop-up, il fallait donc aussi prévoir des petites flèches pour la navigation.

Pour concevoir la fenêtre pop-up elle-même j'ai disposé d'un modèle fourni par le client. J'ai dû procéder à un découpage des parties qu'il était possible de récupérer et de construire les autres en particulier le petit tableau qui affiche les infos de chaque image, le tout a été agencé dans un CSS.

```

1 <?php
2 $i = 0 ;
3 $dir = 'data/19/' //chemin du dossier
4 if ($handle = opendir($dir)) { //si l'on peut ouvrir le dossier
5     while (false !== ($file = readdir($handle))) { // tant qu'il contient des fichiers
6         if ($file != "." && $file != "..") { //si les fichiers ne sont pas des fichiers invisibles ou des dossiers
7             if (substr($file, 0, 1) == ".") { //si le premier caractère du nom de fichier est un point
8                 unlink($dir . $file); //on l'efface
9             } else { //sinon on liste les fichiers
10                $java_photos .= "tab_photos[" . $i . "]=" . $dir . $file . ";\n";
11                $i++;
12            } //if else
13        } //if
14    } //while
15    closedir($handle); //et on ferme le dossier
16 } //if
17 //affichage de la page HTML
18 ?>
19 <html>
20 <head>
21 <link rel="stylesheet" type="text/css" href="style_pop.css" >
22 <script type="text/javascript">
23     var c=0;
24     var tab_photos=new Array(); //initialisation du tableau contenant les photos
25
26     <?php echo $java_photos;?> //affichage de la liste des photos
27
28     function init() { //fonction d'initialisation de la popup à la première photo
29         document.getElementById('photo').src=tab_photos[c];
30     }
31     function precedent() { //bouton précédent
32         c--; //diminution de c
33         if (c<0) { //si c est inférieur à 0 alors c est égal à la longueur du tableau-1 (retour à la dernière photo)
34             c=tab_photos.length-1;
35         }
36         document.getElementById('photo').src=tab_photos[c]; //envoi de la photo correspondante dans le html à partir du tableau des photos
37     }
38     function suivant() { //bouton suivant
39         c++; //augmentation de c
40         if (c==tab_photos.length) { //si c est égal à la longueur du tableau alors c vaut 0 (retour à la première photo)
41             c=0;
42         }
43         document.getElementById('photo').src=tab_photos[c]; //envoi de la photo correspondante dans le html à partir du tableau des photos
44     }
45 </script>

```

Figure 11 Lecture des fichiers et passage dans le tableau JavaScript

```

33     document.getElementById('photo').src=tab_photos[c];
34 }
35 function precedent() { //bouton précédent
36     c--; //diminution de c
37     if (c<0) { //si c est inférieur à 0 alors c est égal à la longueur du tableau-1 (retour à la dernière photo)
38         c=tab_photos.length-1;
39     }
40     document.getElementById('photo').src=tab_photos[c]; //envoi de la photo correspondante dans le html à partir du tableau des photos
41 }
42 function suivant() { //bouton suivant
43     c++; //augmentation de c
44     if (c==tab_photos.length) { //si c est égal à la longueur du tableau alors c vaut 0 (retour à la première photo)
45         c=0;
46     }
47     document.getElementById('photo').src=tab_photos[c]; //envoi de la photo correspondante dans le html à partir du tableau des photos
48 }
49 </script>
50 </head>
51 <body>
52 <br />
53 <table border="0" cellpadding="0" cellspacing="0">
54 <tr>
55 <td class="photo"><img src="" id="photo" /></td>
56 </tr>
57 <tr>
58 <td class="navigation">
59 <br />
60 <a href="#" onclick="precedent();" ></a><a href="#" onclick="suivant();" ></a><br><br><br><br><br><br>
61 </td>
62 </tr>
63 </table>
64 
65 <table border="0" cellpadding="0" cellspacing="0" width="580">
66 <tr>
67 <td id="t1">test</td>
68 <td class="t2" style="width:196px;">test</td>
69 <td class="t2" style="width:79px;"><br></td>
70 <td class="t2">test</td>
71 <td class="t2" style="width:96px;">test</td>
72 </tr>
73 </table>
74 <script type="text/javascript">init();</script>
75 </body>
76 </html>

```

Figure 12 Affichage HTML

Résultat

<http://www.erilia.fr>

offline



Figure 13 La fenêtre pop-up

4.4 KM817

4.4.1 Programmation orientée objet

Au départ il s'agit d'un travail personnel : j'ai fait un formulaire de contact à insérer dans site et qui permet de récolter des données sur l'internaute, nom, prénom, email, téléphone. Puis le visiteur, choisit un titre pour son message dans une liste (ou a la possibilité de le personnaliser), ensuite il rédige son message, il peut s'abonner à une newsletter, noter le site si il le désire : à la validation le mail est envoyé. On m'a alors demandé de l'améliorer pour qu'il puisse également envoyer des pièces jointes.

La demande

Enrichir la bibliothèque de script.

Tâche à réaliser

Une classe permettant d'envoyer des mails sans limitation ni du nombre de destinataires ni du nombre de pièces jointes. L'utilisateur doit pouvoir ajouter ou enlever les destinataires et les pièces jointes à la demande.

Conception

La contrainte par rapport à ce que j'avais déjà fait était que ce « mailer » devait envoyer les emails à plusieurs destinataires, en copie et en copie cachée, et également plusieurs pièces jointes.

Mais comme tout le monde peut se tromper, il a été également souhaité de pouvoir supprimer les différents destinataires et les pièces jointes de façon dynamique.

J'ai donc géré l'affichage avec l'ajout et la suppression dynamique qui incrémentait ou décrémentait le remplissage du mail lui-même.

C'est encore très perfectible, mais cela fonctionne.

```

3  *class_mailto_pj: envoyer un mail avec des pièces jointes
4  *@version $0.5$
5  *@copyright pj 2007
6  */
7  global $mode_admin;
8  $mode_admin = 'off';
9  class classe_mailto_pj {
10 //attributs
11     private $arg_expt;
12     public $dest = array();
13     private $arg_dest;
14     private $destmail;
15     public $dest_copie = array();
16     private $arg_dest_copie;
17     private $destmailcc;
18     public $dest_copie_cachee = array();
19     private $arg_dest_copie_cachee;
20     private $destmailbcc;
21     private $ind;
22     private $arg_sujet;
23     private $arg_message;
24     private $arg_coord_client;
25     private $arg_note_site;
26     private $arg_news;
27     public $pje = array();
28     private $arg_attached;
29     private $arg_pj_2_encode;
30     public $pj_encodee = array();
31     private $entete;
32     private $limite_unique;
33     private $pm;
34     public $envoi;
35     public $uri;
36     public $limite;
37
38 //methodes
39     function expt($arg_expt){
42     function destinationaire($arg_dest){
45     function del_dest($ind){
48     function destinationaire copie($arg_dest_copie){
51     function del_dest copie($ind){
54     function destinationaire copie cachee($arg_dest_copie_cachee){
57     function del_dest copie cachee($ind){

```

Figure 14 La « classe_mailto_pj »

```

1  <?php
2  /**
3  *CLASS CompteurPage
4  *
5  *@version 0.1
6  *@copyright 2006
7  */
8  class CompteurPage{
9  var $Page = 1;
10 var $limit = 15;
11 var $start = 0;
12 var $tab = array();
13 var $nb = 0;
14 var $nb_page = 0;
15 var $Ends;
16
17     function CompteurPage(){
22     function NewLimit($limit){
27     function MakeTab($tab){
32     function Page($Page){
43     function Suiwant(){
52     function Precedent(){
61     function Separateur(){
69     function index(){
108     function _init(){
113     }
114     ?>
115

```

Figure 15 La classe compteur appartenant à la bibliothèque de scripts de km817

```

102      <input type="file" name="lefichier"><input type="submit" value="Ajouter au mail..."><br />
103    </form>
104    <form name="formenvoie_mail" action="index.php" method="POST" enctype="multipart/form-data">
105      <input type="hidden" name="action" value="mail"><br />
106      <b><u>\`est prêt?</b></u><b> >> </b><input type="submit" value="Envoyez!"><br />
107    </form>
108  </td>
109  </tr>
110  <td bgcolor="#CCCCCC">
111    '.$mon_mail->preview_mail().'
112  </td>
113 </tr>
114 </table>
115 </body>
116 </html>';
117
118 //aps clicsur bouton envoyer
119 if (isset($_POST['action']) && $_POST['action'] == 'mail'){
120   //verification de l'existence d'un objet
121   if (isset($_SESSION['classe_mailto_pj']) and $_SESSION['classe_mailto_pj'] != "" )
122   (
123     //envoi du mail
124     $mon_mail->envoie_mail();
125     //aps envoi mail verification du status
126     if ($mon_mail->envoie_mail() == '1'){
127       echo 'Votre Mail à bien été envoyé!';
128       //destruction objet
129       unset($_SESSION['classe_mailto_pj']);
130       $_SESSION['action']='';
131       $mon_mail->__destruct();
132     } else {
133       echo 'Mail non envoyé!';
134       echo $leform;
135       $_SESSION['classe_mailto_pj'] = serialize($mon_mail); //sérialise l'objet mon_mail
136     }
137     //si refresh de la page avec les variable du formulaire presente mais mail déjà envoyé et objet détruit
138   ) else {
139     echo $leform;
140   }
141   //si pas de formulaire envoyé ou si formulaire autre que bouton envoyer
142 } else {
143   echo $leform;
144 }

```

Figure 16 les conditions d'affichage du formulaire

```

73 $start = $_CompteurPage->start;
74 $limit = $_CompteurPage->limit;
75 //requête SQL qui vas'effectuer maintenant que l'on à créé le compteur
76 $sql = 'SELECT nom, prenom, email FROM client LIMIT ' . $start . ', ' . $limit;
77 $result = mysql_query($sql) or die ('Erreur SQL: ' . mysql_error());
78 $nb lignes = mysql_num_rows($result);
79 // si on a récupéré un résultat on l'affiche.
80 if ($nb lignes != '') { //si il y des lignes
81   // début du tableau
82   echo '<table bgcolor="#FFFFFF">'. "\n";
83   // première ligne: on affiche les titres
84   echo '<tr>
85     <td bgcolor="#999999"><b><u>Prénom</u></b></td>
86     <td bgcolor="#999999"><b><u>Nom</u></b></td>
87     <td bgcolor="#999999"><b><u>Envoyer en tant que</u></b></td>
88   </tr>'. "\n";
89   // lecture et affichage des résultats
90   $i=0;
91   // print_r($mon_mail->dest);
92   while ($rangee = mysql_fetch_array($result)) { //tant qu'il y a des résultats
93     unset($selected1, $selected2, $selected3); //désélectionne les
94     if (in_array($rangee[2], $mon_mail->dest)) { //si les destinataires sont dans le tableau des destinataires
95       $selected1 = 'selected'; //remet la sélection sur un
96     } elseif (in_array($rangee[2], $mon_mail->dest_copie)) { //si il sont dans le tableau des destinataires en copie
97       $selected2 = 'selected'; //remet la sélection sur 2
98     } elseif (in_array($rangee[2], $mon_mail->dest_copie_cachee)) { //si il sont dans le tableau de destinataire en copie cachée
99       $selected3 = 'selected'; //met la sélection sur 3
100     }
101     echo '<tr>
102       <td bgcolor="#CCCCCC">'. $rangee[1] . '</td>
103       <td bgcolor="#CCCCCC">'. $rangee[0] . '</td>
104       <td bgcolor="#CCCCCC">
105         <form name="form_dest' . $i . '" action="action_mon_destinaire.php" method="POST" enctype="multipart/form-data">
106           <input type="hidden" name="email" value="' . $rangee[2] . '">
107           <select name="type_dest" onchange="document.form_dest' . $i . '.submit();">
108             <option value="0"></option>
109             <option value="1" ' . $selected1 . '>destinataire principal</option>
110             <option value="2" ' . $selected2 . '>destinataire en copie</option>
111             <option value="3" ' . $selected3 . '>destinataire en copie cachée</option>
112           </select>
113         </form>
114       </td>
115     </tr>';

```

Figure 17 Le SQL et l'affichage avec le CompteurPage

```

1 <?php
2 /*
3 Gère les action pour la construction de l'en-tête de mail (destinataires)
4 */
5 session_start();
6
7 function bugg_on($etat,$string,$color){
21 bugg_on('off', $_SESSION, 'red');
22 require("classe_mailto_pj.php");//inclusion de la classe dans ce script
23 if ($_SESSION['classe_mailto_pj'] == '' or $_GET['obj'] == 'new') //désérialise l'objet $mon_mail
24 $mon_mail = new classe_mailto_pj(); //si il n'y a pas d'objet mon_mail, créé en un nouveau
25 }else //sion désérialise celui qui existe
26 $mon_mail = unserialize($_SESSION['classe_mailto_pj']);
27 }
28 if ($_POST['email'] != '') //si il y a pas 'email' dans le $_POST
29 $type_dest = $_POST['type_dest'];
30 switch ($type_dest) //selon le type de destinataire
31 {
32 case 1 :
33 if (!in_array($_POST['email'],$mon_mail->dest)) //si il n'est pas déjà dans le tableau des destinataires de l'objet
34 {
35 $mon_mail->destinataire($_POST['email']); //le post va dans le tableau des destinataires de l'objet
36 }
37 break;
38 case 2 :
39 if (!in_array($_POST['email'],$mon_mail->dest_copie)) {
40 $mon_mail->destinataire_copie($_POST['email']);
41 }
42 break;
43 case 3 :
44 if (!in_array($_POST['email'],$mon_mail->dest_copie_cachee)) {
45 $mon_mail->destinataire_copie_cachee($_POST['email']);
46 }
47 break;
48 }
49 $_SESSION['classe_mailto_pj'] = serialize($mon_mail); //sérilize l'objet $mon_mail pour le garder dans la session
50 header("Location: mail_base.php");//va sur la page mail_base
51 >>

```

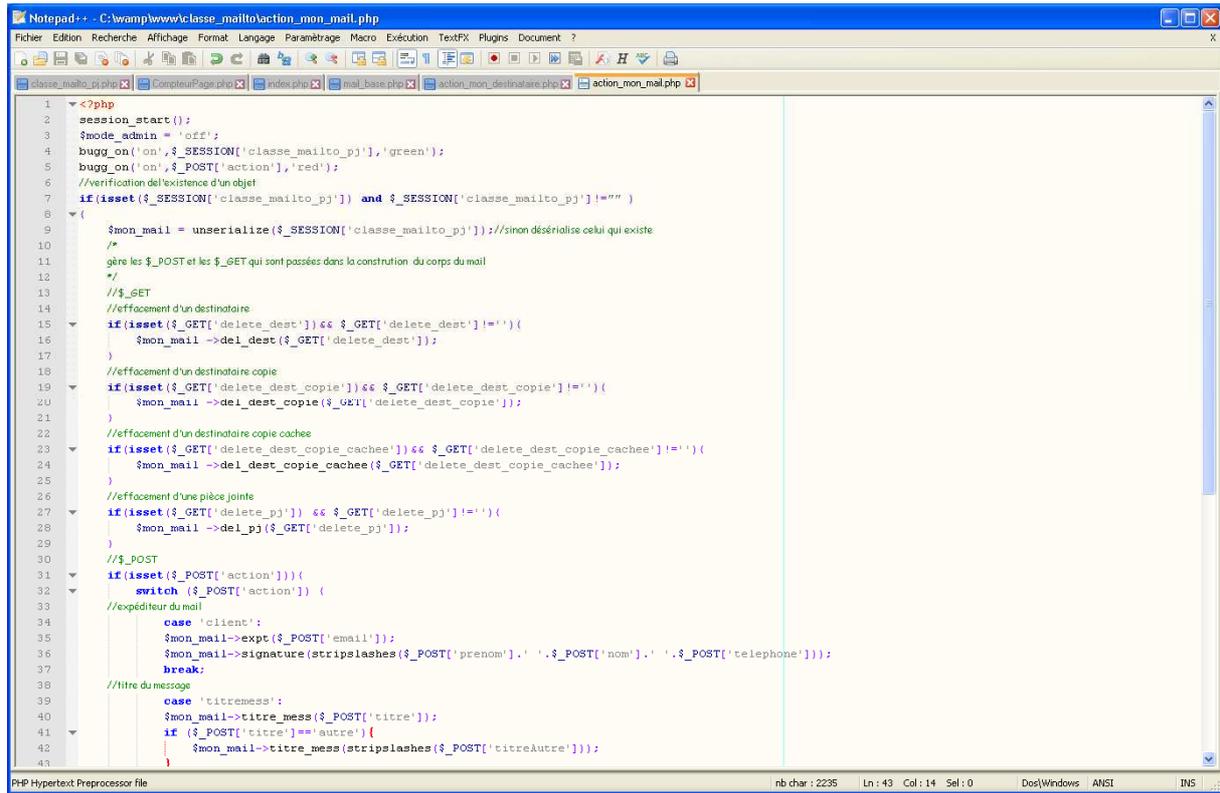
Figure 18 Gestion des destinataires du mail

```

26 //effacement d'une pièce jointe
27 if(isset($_GET['delete_pj']) && $_GET['delete_pj']!=''){
28 $mon_mail->del_pj($_GET['delete_pj']);
29 }
30 //$_POST
31 if(isset($_POST['action'])) {
32 switch ($_POST['action']) {
33 //expéditeur du mail
34 case 'client':
35 $mon_mail->expt($_POST['email']);
36 $mon_mail->signature(stripslashes($_POST['prenom']).' '.$_POST['nom'].' '.$_POST['telephone']);
37 break;
38 //titre du message
39 case 'titremess':
40 $mon_mail->titre_mess($_POST['titre']);
41 if ($_POST['titre']!='autre') {
42 $mon_mail->titre_mess(stripslashes($_POST['titreAutre']));
43 }
44 break;
45 //message
46 case 'message':
47 $mon_mail->message(stripslashes($_POST['lemessage']));
48 break;
49 //note du site
50 case 'note':
51 $mon_mail->note_site($_POST['note']);
52 break;
53 //newsletter
54 case 'news':
55 $mon_mail->newsletter($_POST['newsletter']);
56 break;
57 //pièce jointe
58 case 'pj':
59 $mon_mail->pj_2_encode($_FILES['le fichier']['tmp_name']);
60 $mon_mail->attached($_FILES['le fichier']);
61 break;
62 //quand c'est fini
63 /* case 'mail':
64 $mon_mail->envoie_mail();
65 break;*/
66 } // switch
67 //if
68 $_SESSION['classe_mailto_pj'] = serialize($mon_mail);

```

Figure 19 Remplissage du mail dans l'objet avec les \$_POST



```
1 <?php
2 session_start();
3 $mode_admin = 'off';
4 bugg_on('on',$_SESSION['classe_mailto_pj'],'green');
5 bugg_on('on',$_POST['action'],'red');
6 //verification de l'existence d'un objet
7 if(isset($_SESSION['classe_mailto_pj']) and $_SESSION['classe_mailto_pj']!="")
8 {
9     $mon_mail = unserialize($_SESSION['classe_mailto_pj']); //sinon déserialise celui qui existe
10    /*
11     gère les $_POST et les $_GET qui sont passés dans la construction du corps du mail
12     */
13     //$GET
14     //effacement d'un destinataire
15     if(isset($_GET['delete_dest']) && $_GET['delete_dest']!=''){
16         $mon_mail->del_dest($_GET['delete_dest']);
17     }
18     //effacement d'un destinataire copie
19     if(isset($_GET['delete_dest_copie']) && $_GET['delete_dest_copie']!='){
20         $mon_mail->del_dest_copie($_GET['delete_dest_copie']);
21     }
22     //effacement d'un destinataire copie cachee
23     if(isset($_GET['delete_dest_copie_cachee']) && $_GET['delete_dest_copie_cachee']!='){
24         $mon_mail->del_dest_copie_cachee($_GET['delete_dest_copie_cachee']);
25     }
26     //effacement d'une pièce jointe
27     if(isset($_GET['delete_pj']) && $_GET['delete_pj']!='){
28         $mon_mail->del_pj($_GET['delete_pj']);
29     }
30     //$POST
31     if(isset($_POST['action'])){
32         switch($_POST['action']) {
33             //expéditeur du mail
34             case 'client':
35                 $mon_mail->exp($_POST['email']);
36                 $mon_mail->signature(stripslashes($_POST['prenom'].' '.$_POST['nom'].' '.$_POST['telephone']));
37                 break;
38             //titre du message
39             case 'titremess':
40                 $mon_mail->titre_mess($_POST['titre']);
41                 if($_POST['titre']!="autre"){
42                     $mon_mail->titre_mess(stripslashes($_POST['titreautre']));
43                 }
44         }
45     }
46 }
```

Figure 20 Effaçage des destinataire ou des pièces jointes par le \$_GET

Résultat

La classe fonctionne et envoie des mails.

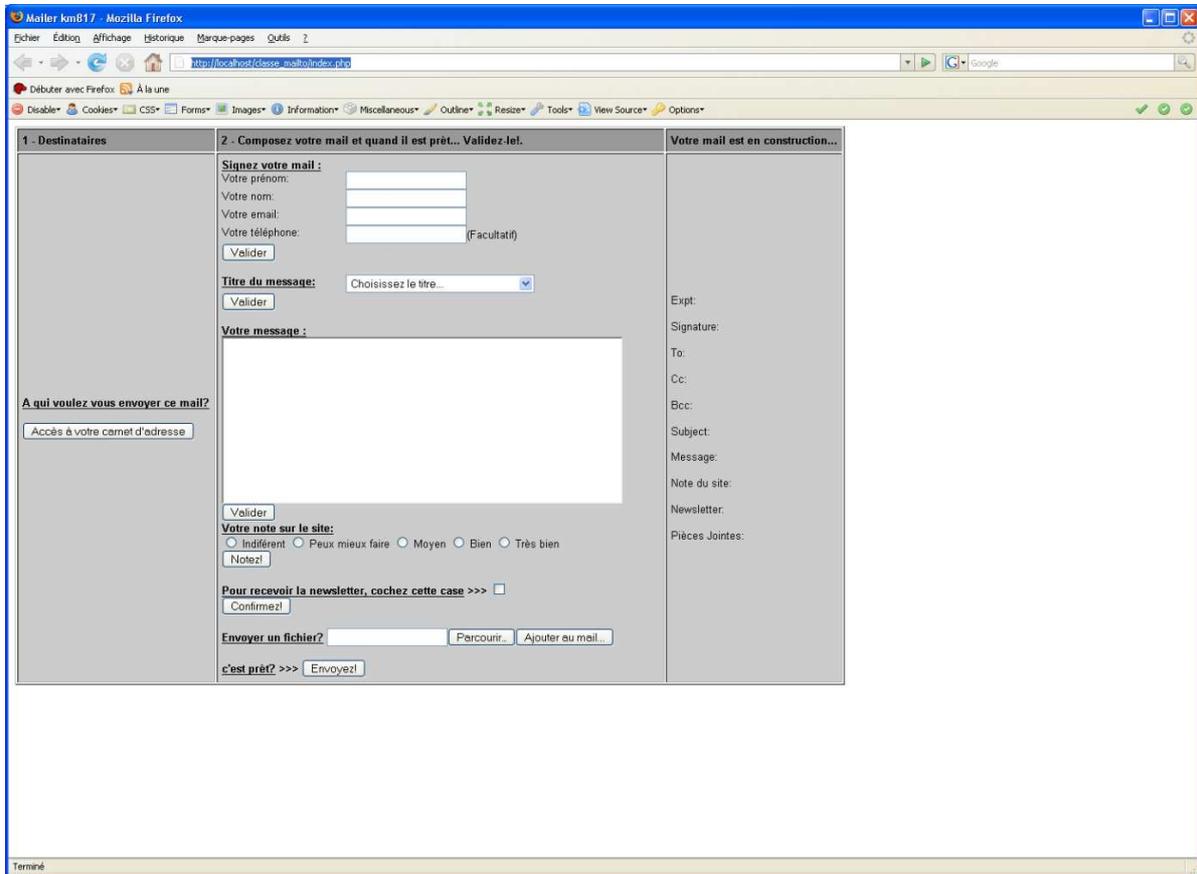


Figure 214 Le mail vide

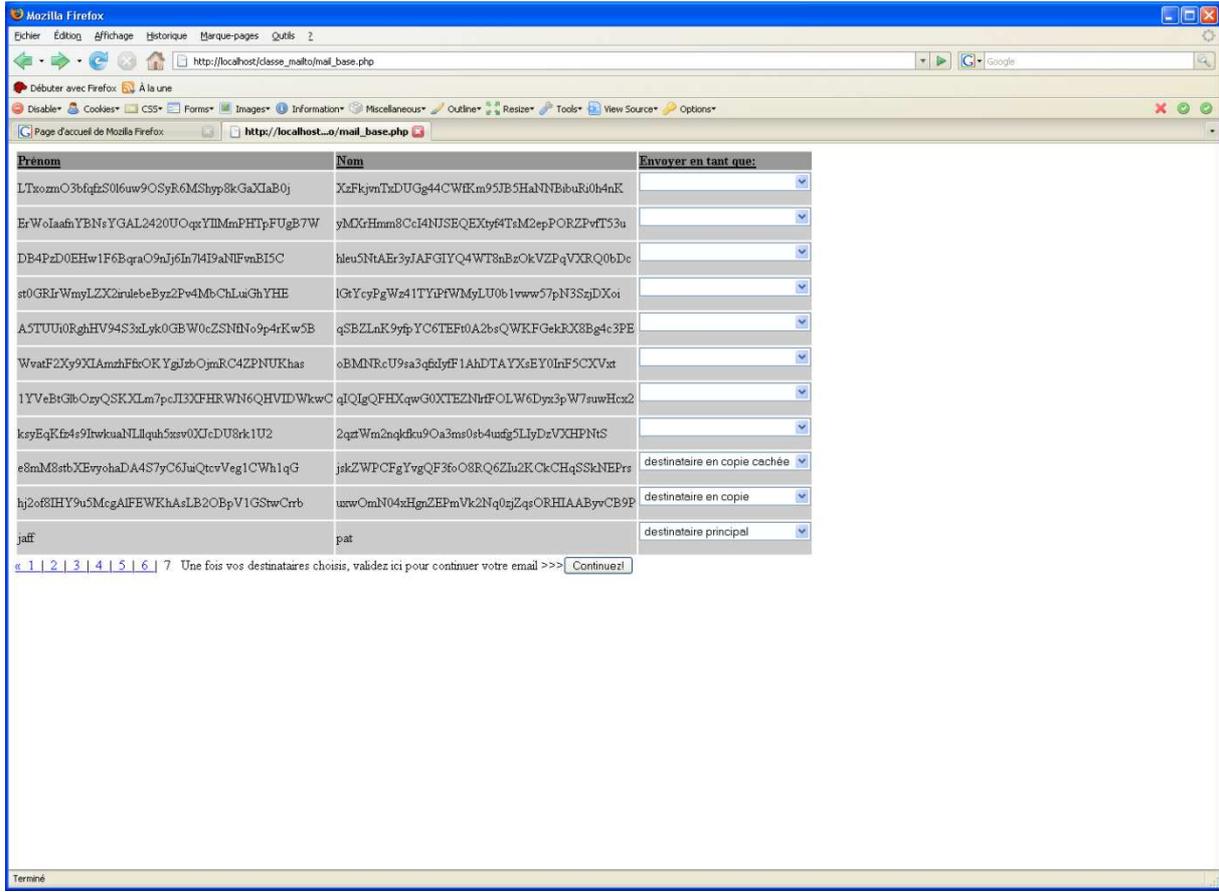


Figure 225 Le choix du ou des destinataires par l'utilisateur

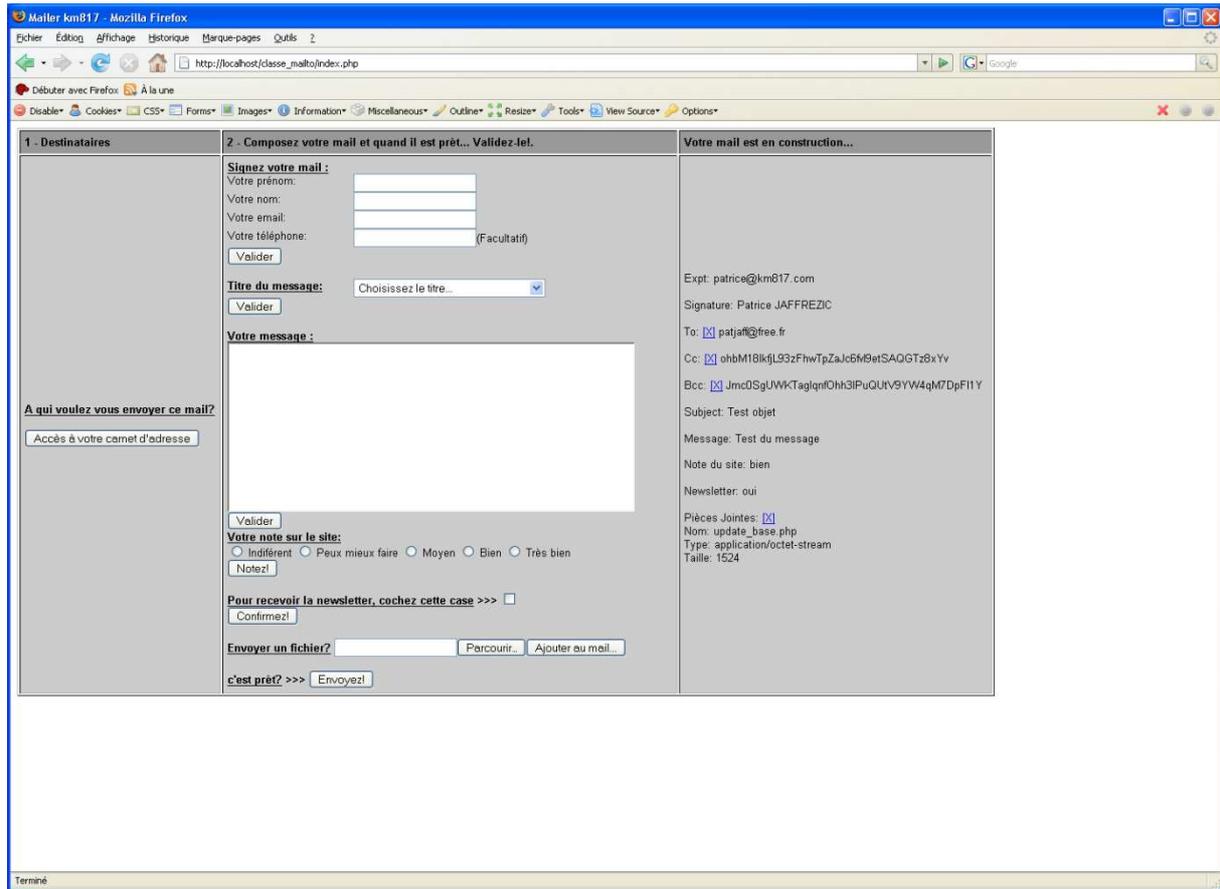


Figure 23 Le mail rempli prêt à être envoyé

4.4.2 Analyse

Quand tout à fonctionné correctement il m'a été demandé de concevoir une base de données qui puisse accompagner cette classe.

Sur un dictionnaire de données qui m'a été fourni j'ai conçu le MCD, après la validation, j'en ai déduit le MLD et j'ai codé le SQL.

Cette base n'est interfacée que partiellement avec le « mailer » et un formulaire permettant de la remplir de type carnet d'adresse doit être ajouté.

Les relations « citoyen », « exercer », « travailler », sont volontairement non contraignantes pour ne pas gêner l'utilisateur futur du carnet d'adresses.

La relation « inclus » se transforme en table pour avoir plus rapidement accès à l'information 'quel mail avec quelle pièce jointe' et inversement, car la table « pièces_jointes » contient le fichier encodé de la pièce jointe et cette table s'alourdira assez vite avec le temps.

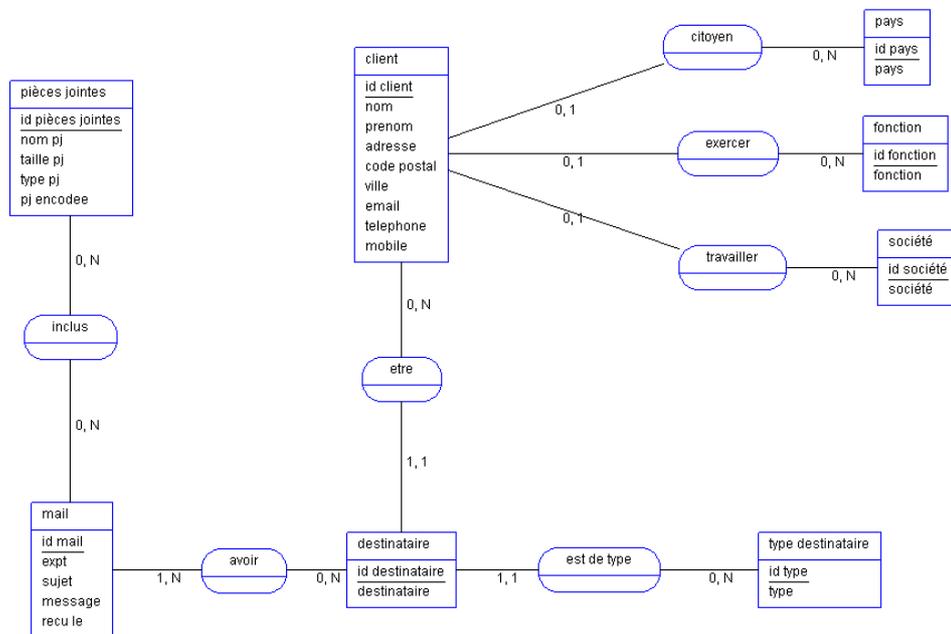


Figure 24 MCD base_mailto_pj

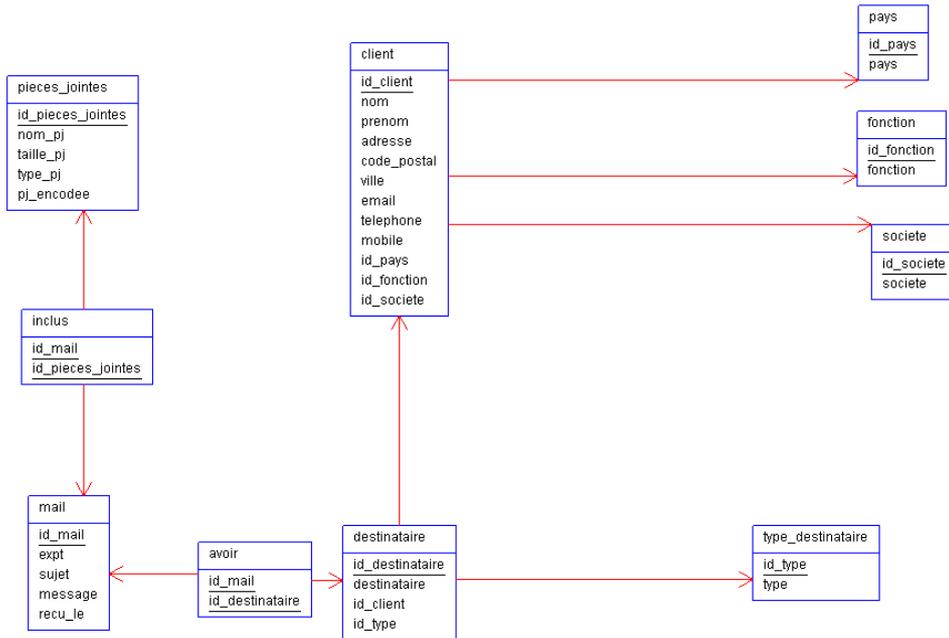


Figure 25 MLD base_mailto_pj

```

1 CREATE TABLE client (id_client int NOT NULL, nom varchar(40), prenom varchar(40), adresse varchar(100), code_postal varchar(5), ville
2 CREATE TABLE mail (id_mail int NOT NULL, expt varchar(40), sujet varchar(40), message varchar(1000), recu_le datetime);
3 CREATE TABLE pays (id_pays int NOT NULL, pays varchar(40));
4 CREATE TABLE fonction (id_fonction int NOT NULL, fonction varchar(40));
5 CREATE TABLE pieces_jointes (id_pieces_jointes int NOT NULL, nom_pj varchar(40), taille_pj float, type_pj varchar(40), pj_encodee binary(16));
6 CREATE TABLE societe (id_societe int NOT NULL, societe varchar(40));
7 CREATE TABLE destinataire (id_destinataire int NOT NULL, destinataire varchar(40), id_client int, id_type tinyint);
8 CREATE TABLE type_destinataire (id_type tinyint NOT NULL, type varchar(15));
9 CREATE TABLE inclus (id_mail int NOT NULL, id_pieces_jointes int NOT NULL);
10 CREATE TABLE avoir (id_mail int NOT NULL, id_destinataire int NOT NULL);
11 ALTER TABLE client ADD CONSTRAINT PK_client PRIMARY KEY (id_client);
12 ALTER TABLE mail ADD CONSTRAINT PK_mail PRIMARY KEY (id_mail);
13 ALTER TABLE pays ADD CONSTRAINT PK_pays PRIMARY KEY (id_pays);
14 ALTER TABLE fonction ADD CONSTRAINT PK_fonction PRIMARY KEY (id_fonction);
15 ALTER TABLE pieces_jointes ADD CONSTRAINT PK_pieces_jointes PRIMARY KEY (id_pieces_jointes);
16 ALTER TABLE societe ADD CONSTRAINT PK_societe PRIMARY KEY (id_societe);
17 ALTER TABLE destinataire ADD CONSTRAINT PK_destinataire PRIMARY KEY (id_destinataire);
18 ALTER TABLE type_destinataire ADD CONSTRAINT PK_type_destinataire PRIMARY KEY (id_type);
19 ALTER TABLE inclus ADD CONSTRAINT PK_inclus PRIMARY KEY (id_mail, id_pieces_jointes);
20 ALTER TABLE avoir ADD CONSTRAINT PK_avoir PRIMARY KEY (id_mail, id_destinataire);
21 ALTER TABLE client ADD CONSTRAINT FK_client_id_pays FOREIGN KEY (id_pays) REFERENCES pays (id_pays);
22 ALTER TABLE client ADD CONSTRAINT FK_client_id_fonction FOREIGN KEY (id_fonction) REFERENCES fonction (id_fonction);
23 ALTER TABLE client ADD CONSTRAINT FK_client_id_societe FOREIGN KEY (id_societe) REFERENCES societe (id_societe);
24 ALTER TABLE destinataire ADD CONSTRAINT FK_destinataire_id_client FOREIGN KEY (id_client) REFERENCES client (id_client);
25 ALTER TABLE destinataire ADD CONSTRAINT FK_destinataire_id_type FOREIGN KEY (id_type) REFERENCES type_destinataire (id_type);
26 ALTER TABLE inclus ADD CONSTRAINT FK_inclus_id_mail FOREIGN KEY (id_mail) REFERENCES mail (id_mail);
27 ALTER TABLE inclus ADD CONSTRAINT FK_inclus_id_pieces_jointes FOREIGN KEY (id_pieces_jointes) REFERENCES pieces_jointes (id_pieces_joi);
28 ALTER TABLE avoir ADD CONSTRAINT FK_avoir_id_mail FOREIGN KEY (id_mail) REFERENCES mail (id_mail);
29 ALTER TABLE avoir ADD CONSTRAINT FK_avoir_id_destinataire FOREIGN KEY (id_destinataire) REFERENCES destinataire (id_destinataire);
30
31

```

Figure 26 SQL base_mailto_pj

4.4.3 Fichiers

Ensuite j'ai écrit une petite routine PHP qui génère un fichier .CSV et qui permet de remplir la base de données aléatoires de façon à pouvoir en tester le bon fonctionnement.

```

1 <<?php
2 $mode_admin='on';
3 $fichEcriture = "test_base_mailto.csv";
4 $tab = array();
5
6 function bugg_on($etat,$string,$color){
7
8 }
9
10 function random_string($long) //génère une chaîne aléatoire de la longueur désirée
11 {
12     $char = 'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890abcefghiijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ1234567890';
13     $string = substr(str_shuffle($char), 0, $long);
14     return $string;
15 }
16
17 //Génération des enregistrements
18 $nb_enr = 100; // choix du nb d'enregistrements à générer
19
20 for ($i=0; $i<$nb_enr; $i++){
21     bugg_on (on, $tab, blue);
22     $tabs = implode(" ", $tab); //mise du tableau en string
23     file_put_contents($fichEcriture, $tabs); //Ecriture dans le fichier
24     $tab = " "; //réinitialisation du tableau
25 }
26
27 >>

```

Figure 27 Fonction random string qui génère une chaîne aléatoire de la longueur désirée

```

26 //Génération des enregistrements
27 $nb_enr = 100; // choix du nb d'enregistrements à générer
28
29 for ($i=0; $i<$nb_enr; $i++){ //laisser vide pour l'auto-incrémentation dans le table client (1er champ)
30     $stringid_client = " "; //laisser vide pour l'auto-incrémentation dans le table client (1er champ)
31     array_push($tab,$stringid_client);
32     $stringnom = random_string(40);
33     $stringnom = " ".$stringnom." ";
34     array_push($tab,$stringnom);
35     $stringprenom = random_string(40);
36     $stringprenom = " ".$stringprenom." ";
37     array_push($tab,$stringprenom);
38     $stringadresse = random_string(100);
39     $stringadresse = " ".$stringadresse." ";
40     array_push($tab,$stringadresse);
41     $stringcp = random_string(5);
42     $stringcp = " ".$stringcp." ";
43     array_push($tab,$stringcp);
44     $stringville = random_string(40);
45     $stringville = " ".$stringville." ";
46     array_push($tab,$stringville);
47     $stringemail = random_string(40);
48     $stringemail = " ".$stringemail." ";
49     array_push($tab,$stringemail);
50     $stringtel = random_string(15);
51     $stringtel = " ".$stringtel." ";
52     array_push($tab,$stringtel);
53     $stringamob = random_string(15);
54     $stringamob = " ".$stringamob." ";
55     array_push($tab,$stringamob);
56     $stringid_pays = rand(1, 4);
57     $stringid_pays = " ".$stringid_pays." ";
58     array_push($tab,$stringid_pays);
59     $stringid_fonction = rand(1, 4);
60     $stringid_fonction = " ".$stringid_fonction." ";
61     array_push($tab,$stringid_fonction);
62     $stringid_societe = rand(1, 2);
63     $stringid_societe = " ".$stringid_societe." :zet' ";
64     array_push($tab,$stringid_societe);
65 }
66
67 bugg_on (on, $tab, blue);
68 $tabs = implode(" ", $tab); //mise du tableau en string
69 file_put_contents($fichEcriture, $tabs); //Ecriture dans le fichier
70 $tab = " "; //réinitialisation du tableau
71
72 >>

```

Figure 28 Génération de différents champs de la table 'client' (on choisi le nb d'enregistrements à insérer dans la table)



Figure 29 fichier .csv importé dans la base par PHPMYAdmin

5 Conclusion

Incontestablement ce stage est une première expérience professionnelle réussie qui a permis l'utilisation de toutes mes compétences accumulées pendant la formation proprement dite, y compris l'anglais.

Cette mise en situation m'a également permis d'acquérir des compétences non acquises durant la formation (Merise), en effet, rien de mieux que d'être confronté à un cas réel pour mieux comprendre le pourquoi et le comment des choses.

J'ai également obtenu de nouvelles connaissances puisque j'ai eu l'occasion d'apprendre un nouveau langage, PHP, et d'approfondir celles que j'avais déjà en JavaScript, HTML.

Surtout j'ai appris une méthode de travail : apprendre à chercher des informations efficacement : « Google est ton ami ! » (Merci Éric !), bien analyser un problème avant de se lancer « tête la première » dans le code, et enfin de toujours tester son code au fur et à mesure de façon à ne pas avoir à rechercher le « bug » dans les « 300 lignes » de code précédentes, ce qui au final nous paraît une perte de temps en réalité nous en fait gagner beaucoup ! (merci Liz, je n'oublierai pas tes conseils).

J'ai trouvé dans cette formation et ce métier que je m'appête à exercer ce que j'étais venu y chercher : « Un nouveau défi chaque jour ! ».