



IUT de Provence, site d'Arles
Rue Raoul Follereau (Route de Crau)
13200 Arles

RAPPORT DE STAGE

Boyer Julien
Informatique 2ème année
Promotion 2004 - 2006

Stage effectué du 11 avril au 17 juin à la DSIT d'Arles



ARLES

DSIT
Place de la République
13200 Arles.

Tuteur de stage à l'IUT :

M. Romain Raffin

Tuteur de stage dans l'entreprise :

M. François Raynaud

Remerciements

Je tiens à remercier Mr François Raynaud pour m'avoir permis d'effectuer ce stage au sein de son service ainsi que pour ses conseils et le temps qu'il m'a accordé.

Ensuite, je souhaite exprimer ma gratitude à toute l'équipe de la DSIT ainsi que les quatre autres stagiaires présents lors de mon stage avec qui j'ai passé un agréable moment tout au long du stage.

Je remercie également Mr Romain Raffin mon tuteur, Mr Reboulet, responsable des stages et Youssef Chikhaoui, responsable des secondes années de DUT informatique ainsi que tous les professeurs de l'IUT d'Arles.

« Le contenu et les opinions de ce mémoire n'engagent que leur auteur et ne sauraient en aucun cas être considérés comme écrits sous une autre responsabilité. »

Introduction	5
English Introduction	6
1 Contexte	7
1.1 Présentation de l'entreprise	7
1.2 Organigramme du service	8
1.3 Contraintes de développement	8
1.4 Le stage	9
1.4.1 Le sujet	9
1.4.2 Le projet	9
2 la demande et l'existant	10
2.1 La demande	10
2.2 Description du système actuel	10
3 Description du travail réalisé.....	10
3.1 Mise en place de l'environnement de travail	10
3.2 La compréhension des classes open Mairie	11
3.3 La compréhension de l'architecture d'un site open Mairie	11
3.4 Prise en main du code existant	11
3.5 Ajout des fonctionnalités à l'application	12
4 Les classes open Mairie.....	12
4.1 La classe dbForm	13
4.2 La classe Formulaire	13
4.3 La classe tab	14
5 Présentation de l'application	14
5.1 Cas d'utilisation.....	14
5.1.1 Saisie de courrier	14
5.1.2 Modification	15
5.1.3 Suppression	16
5.1.4 Recherche	16
5.2 Architecture et explication de l'application	17
5.2.1 Affichage d'un formulaire	17
5.2.2 Affichage d'un sous formulaire	18
5.2.3 Recherche de courriers	19
6 Activités annexes.....	20
7 Problèmes rencontrés	20
8 Conclusion.....	21
Glossaire.....	22
Bibliographie.....	23
Annexes.....	24

Introduction

Ce stage de 10 semaines (du 10 avril au 16 juin) vient clôturer la formation en DUT informatique option imagerie numérique.

J'ai effectué mon stage à la mairie d'Arles et plus précisément à la DSIT qui est le service informatique.

Le but de mon stage était d'apporter de nouvelles fonctionnalités au logiciel open courrier qui sert à la gestion du courrier ; en effet une administration telle que la mairie d'Arles recevant plusieurs centaines de courriers par jour implique la nécessité d'en informatiser la gestion.

Dans un premier temps, je me suis familiarisé avec les classes open Mairie à partir desquelles open courrier a été créée.

Ensuite, j'ai dû développer un moteur de recherche permettant de rechercher parmi tous les courriers saisis selon plusieurs critères laissés au choix de l'utilisateur.

Enfin, la dernière partie consistait à gérer la diffusion d'un courrier en permettant la création de sous formulaires rattachés à un courrier.

English Introduction

My placement in the **DSIT** started on April 10th 2006 and lasted for 10 week until June 15th 2006.

Its goal was to give me a first professional experience and is an essential part of the DUT in computer science formation.

The **DSIT** is the computing department of the town of Arles. This department had developed many open source applications in the object oriented language PHP.

During this period of time, I had to implement new functionalities in the mail management software 'open courrier'.

This placement provided me a great experience within an experienced team.

1 Contexte

1.1 Présentation de l'entreprise

J'ai fait mon stage à la **D**irection des **S**ystèmes Informatiques et **T**élécommunications.

Ce service a été créé en 1997 et est composé de 5 services :

- Administration ;
- Développement ;
- Réseau et télécommunications ;
- Bureautique ;
- Administration système.

Au total, ce sont 14 employés qui sont répartis dans ces services.

J'ai pour ma part effectué mon stage au sein du service développement.
La DSIT est à l'origine des composants open Mairie.

A partir de ces composants, plusieurs applications ont été développées ou sont encore en cours de développement comme par exemple openElec qui sert à la gestion des listes électorales, openCimetière utilisé comme son nom l'indique dans la gestion des cimetières ainsi que des applications de gestion de parcs informatiques, de stocks, de demandes de logements,...

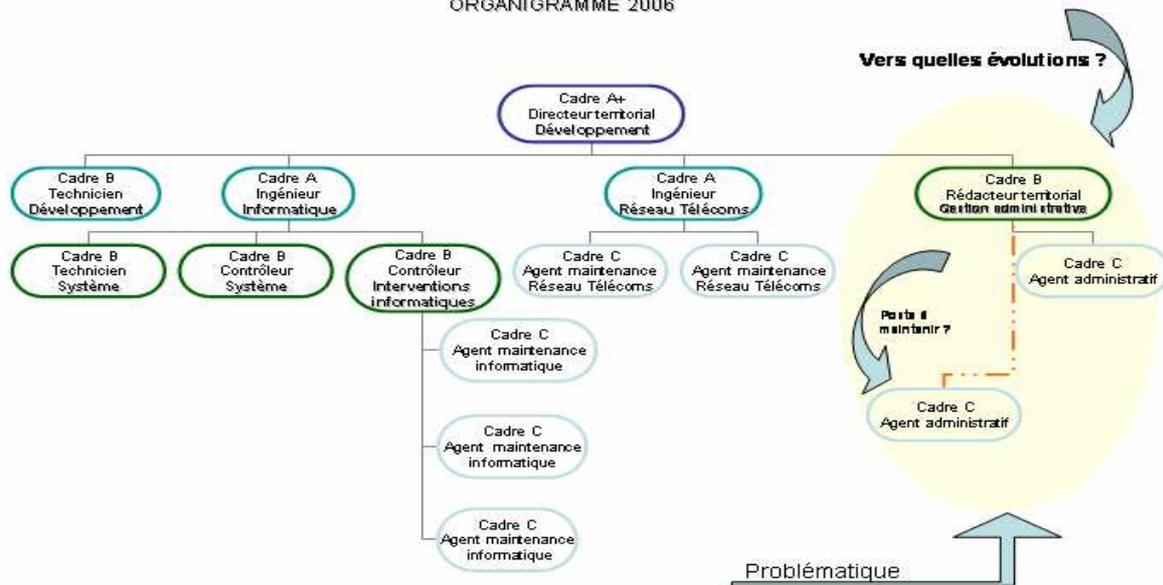
Les composants open Marie et les différents logiciels développés à partir de ceux-ci sont open source et sont téléchargeables sur le **GFORGE** de l'**ADULLACT**
L'adhésion de la DSIT à l'**ADULLACT** ainsi que sa collaboration avec des sociétés de services telles que atReal elle aussi impliquée dans le développement de logiciels open sources montrent sa volonté de s'investir dans le domaine du logiciel libre.

1.2 Organigramme du service

Annexe I

Direction des Systèmes d'Information
et de Télécommunication

ORGANIGRAMME 2006



1.3 Contraintes de développement

Les applications sont basées sur un composant développé par la DSIT baptisé open Mairie.

Open Mairie, écrit en PHP objet, permet la création d'applications basées sur des formulaires, sous formulaires, d'affichage sous forme tabulaire et l'édition au format PDF du contenu de la base de données. Afin d'adapter les possibilités du composant à une application spécifique, la surcharge de méthodes est nécessaire.

L'application que j'ai développé devait fonctionner sous la version 4 de PHP et pour ce faire j'ai dû utiliser les classes mères open Mairie. Il m'a aussi été demandé d'utiliser easyphp version 1.7.

En ce qui concerne la gestion des différents SGBD pouvant être utilisés de paire avec les applications dérivées d'open Mairie, l'utilisation du connecteur open source DBPear était imposée.

Ce connecteur est un ensemble de classes écrites en PHP permettant de faire abstraction du SGBD utilisé en proposant un ensemble de méthodes s'adaptant à n'importe quel SGBD.

La base de données est de type MySql, mais l'application doit permettre d'utiliser d'autres SGBD ; pour ce faire, il suffit de mettre au point un fichier spécifique au SGBD utilisé.

La mairie d'Arles étant adhérente à l'**ADULLACT**, et cette dernière a mis en place un système de GFORGE accessible par l'intermédiaire d'un logiciel de type **CVS** (Tortoise par exemple).

Enfin, l'application devait fonctionner sous Internet Explorer et Mozilla Firefox, ces deux navigateurs étant les plus répandus.

1.4 Le stage

Ce stage doit être effectuée dans le cadre de l'obtention du Dut informatique. D'une durée de dix semaines, il a pour but d'apporter une expérience professionnelle ayant pour but une mise en pratique des acquis au cours des deux années que comprends la formation et aussi de faciliter l'insertion dans le milieu professionnel.

1.4.1 Le sujet

Le sujet de mon stage était d'apporter de nouvelles fonctionnalités au logiciel existant open Courrier version 1.0 développé en PHP afin qu'il corresponde mieux aux attentes des utilisateurs.

1.4.2 Le projet

Le projet que j'avait à réalisé peut être découpé en deux parties :

La première consistait à développer un moteur de recherche permettant la recherche d'un ou plusieurs enregistrements dans la table courrier selon plusieurs critères.

La seconde partie portait sur la mise en place d'un sous formulaire servant à la diffusion du courrier.

2 la demande et l'existant

2.1 La demande

Le service du courrier de la mairie d'Arles utilisait auparavant le logiciel open courrier dans sa version 1.0. Cette version ne proposait pas de fonction de recherche avancée; en effet il était seulement possible de rechercher une chaîne sur l'ensemble des champs d'une table.

Actuellement, la base de données contenant plus de 40 000 enregistrement, ce type de recherche s'avérait être peu efficace.

De plus, certains courriers reçus donnent lieu à une diffusion au sein d'un service; cette diffusion à partir d'un courrier était impossible dans la version précédente.

Les modifications à apporter au logiciel étaient donc de développer un moteur de recherche multi critères au niveau de la table courrier et la création d'un sous formulaire permettant l'affectation de tâches suite à la réception d'un courrier.

2.2 Description du système actuel

Le service du courrier de la mairie d'Arles emploie trois personnes chargées de saisir les différents courriers reçus.

Elles utilisent l'application open courrier version 1.0 pour saisir les courriers reçus dans la journée.

Une fois un courrier saisi, une personne est chargée de distribuer ce courrier aux services ou personnes concernées. Cette même personne est aussi chargée de récupérer les différents courriers destinés soit à un autre service soit à l'extérieur de la mairie.

3 Description du travail réalisé

3.1 Mise en place de l'environnement de travail

La première tâche à laquelle je me suis attelé était de mettre en place l'environnement de travail dans lequel j'allais devoir évoluer.

Dans un premier temps, j'ai du installer easy php version 1.7 et l'installer.

Une fois l'application installée, j'ai eu à la paramétrer; c'est à dire lui indiquer où se trouvaient les librairies requises; c'est pourquoi dans un deuxième temps j'ai du télécharger les librairies nécessaires au développement de l'application à savoir :

- DB Pear, le connecteur de bases de données;
- Open Mairie;

- Fpdf.

Restait ensuite à inclure leurs chemins d'accès dans le fichier php.ini se trouvant dans le répertoire apache de easyphp (un redémarrage des serveurs est nécessaire pour prendre en compte les changements).

Enfin, l'utilisation du logiciel CVS Tortoise étant de rigueur, j'ai dû le télécharger dans sa version la plus récente (v 1.8.25) et l'installer.

Une fois Tortoise installé et un compte ouvert à mon nom sur le GFORGE de l'ADULLACT par Mr Raynaud, l'administrateur des logiciels open Mairie, il m'était alors possible de mettre à jour mon travail et de le récupérer à tout moment au niveau de ce serveur.

3.2 La compréhension des classes open Mairie

Ce fût un travail assez difficile; en effet, open Mairie est composé de sept classes abstraites (la plus courte comporte 248 lignes et la plus longue en fait 961) qui sont malheureusement peu commentées. J'ai principalement utilisé trois de ces classes qui seront présentées dans le chapitre suivant.

Une documentation proposant une présentation de chaque classe et des méthodes qui y sont rattachées est consultable en ligne sur le site de open Mairie dont le lien est proposé dans la bibliographie.

De plus, un document de prise en main élaboré par un autre stagiaire était aussi disponible sur le site.

Malgré le fait que j'ai eu assez de mal à comprendre le fonctionnement des classes open Mairie, toute l'équipe chargée du développement m'a fourni une aide précieuse

3.3 La compréhension de l'architecture d'un site open Mairie.

Parallèlement à la compréhension des classes open Mairie, j'ai dû comprendre la façon dont est créé un site à l'aide des classes open Mairie. Outre la surcharge des méthodes des classes mères, une application basée sur open Mairie fait appel à au moins 3 fichiers séparés pour créer un affichage de formulaire ou sous forme de tableau.

3.4 Prise en main du code existant

Dans le but de comprendre plus rapidement le code, j'ai passé mes premiers jours à effectuer des recherches sur Internet à propos des fonctions PHP utilisées qui m'étaient inconnues.

J'ai par la suite commencé à toucher au code en améliorant sensiblement la fonction de recherche préexistante en permettant de sélectionner via un champ select (choix multiple) le champ sur lequel on souhaitait effectuer la recherche.

Bien que cette modification fût à terme remplacé par un véritable moteur de recherche, elle m'a permis de faire mes premières surcharges des fonctions de open Mairie et par la même occasion d'évoluer dans la compréhension du code.

3.5 Ajout des fonctionnalités à l'application

Une fois le code et l'architecture compris, j'ai commencé par corriger quelques problèmes que j'avais repérés au niveau de l'application existante comme implémenter une gestion des dates au niveau d'un formulaire ou permettre l'ajout de courrier à la chaîne.

Ensuite, je me suis mit à ajouter les fonctionnalités demandées c'est à dire développer un moteur de recherche et permettre la diffusion du courrier.

Le développement du moteur de recherche s'est déroule en trois parties :

- La création de l'interface où les données recherchées seront saisies;
- La surcharge de la classe tab de open Mairie ;
- L'adaptation du fichier faisant le lien entre la saisie et l'affichage des données.

Ensuite, la mise en place des sous formulaire peut être décomposée de la manière suivante :

- Création de la table 'tâche' ;
- Implémentation de l'affichage tabulaire des sous formulaires rattachés à un courrier ;
- Modification de la table 'service'.
- Création du sous formulaire.

4 Les classes open Mairie

Cette partie va nous permettre d'aborder trois des classes open Mairie que j'ai le plus utilisé afin de mener à bien le projet. La compréhension de ces classes est fondamentale dans la mise en œuvre du projet et correspond à la première tâche à laquelle je me suis attelé.

L'ensemble des classes open Mairie est composé de méthodes abstraites, il est donc souvent nécessaire de les surcharger afin de les adapter aux besoins de l'application.

Les classes et les méthodes qui les composent présentées ci-dessous sont celles qui m'ont été le plus utiles lors de la mise en œuvre du projet.

Il est à noter que je ne traiterais pas de **DB Pear** ni de **fpdf** étant donné que je l'ai implicitement utilisé; ses méthodes étant appelées au niveau des fonctions de open Mairie.

4.1 La classe *dbForm*

Cette classe sert à la création et la gestion de l'interface entre l'utilisateur et la base de données.

L'ensemble des méthodes qui la constitue est donc orienté vers la construction des données du formulaire et la manipulation des données au niveau de la base :

Les fonctions ajouter – modifier – supprimer (\$val,&\$db,\$DEBUG) :

Ces fonctions permettent de manipuler directement la base de donnée via l'utilisation de fonctions du connecteur **DBPEAR**.

Les données qui seront traitées au niveau de la base sont représentées par \$val ;
Le dernier paramètre sert à afficher des informations complémentaires utiles pour débayer l'application.

D'autres méthodes sont consacrées à la création du formulaire faisant office d'intermédiaire entre l'utilisateur et la base et permettent entre autre la création de champs de saisie, l'affectation d'un libellé à un champ, le groupement de plusieurs champs :

- Settype : sert à définir le type de champ;
- Setval : affecte une valeur par défaut à un objet ;
- Setlib : permet d'afficher un libellé correspondant à un champ;
- Settaille : sert à affecter une taille à un champ;
- Setmax : permet de définir le nombre de lignes d'un champ;
- Setselect : sert à affecter les différentes valeurs à un champ select;
- SetOnchange : effectue des contrôles lors des changements de valeurs;
- SetGroupe : regroupe plusieurs champs.

Toutes ces fonctions prennent en paramètre l'adresse d'un formulaire et un entier servant à indiquer quel type de traitement on souhaite réaliser (ajout, suppression, mise a jour). Les fonctions portant le même nom de la classe formulaire sont appelées grâce à l'adresse du formulaire.

Cette classe dispose également d'une fonction retour qui permet de faire le lien entre l'affichage du formulaire et l'affichage de la base.

4.2 La classe *Formulaire*

Cette classe est appelée au niveau de la fonction formulaire de dbForm.

Le formulaire est créé en fonction des valeurs passées par les différentes méthodes de dbForm.

On y retrouve les fonctions Settype, Setval, Setlib, Settaille, Setmax, Setselect, SetOnchange et Setgroupe qui sont appelées par les fonctions éponymes de dbForm. Chacune de ces fonctions prends en paramètre un champ et une valeur à affecter à ce champ.

Un autre ensemble de fonctions est disponible. Ces fonctions ont pour but la création des champs au niveau du formulaire :

- Hiddenstatic : crée un champ non modifiable ;
- date : fait apparaître un champ de saisie de date;
- select : permet de créer un champ à choix multiple;
- comboG : affiche le champ à droite au niveau du formulaire;
- comboD : affiche le champ à gauche au niveau du formulaire;
- upload : ouvre un sélecteur de fichier;
- hidden : champ non modifiable.

Toutes ces fonctions prennent en paramètres le nom du champ et créent un champ de type souhaité.

4.3 La classe tab

Cette classe permet de présenter le contenu d'une table de la base en créant un affichage sous forme de tableau.

Le tableau est composé de l'entête, du corps du tableau, d'une entête propre à chaque colonne, de boutons servant à naviguer entre les différentes pages du tableau. A ceci vient s'ajouter une fonction qui crée la requête pour sélectionner les différents éléments que l'on souhaite afficher.

L'entête du tableau est créée par la fonction entête et par défaut affiche un champ permettant une recherche sur la table concernée.

Le corps du tableau est généré par la fonction afficher dont les paramètres sont les suivants : (\$premier,\$recherche,\$href,\$db,\$DEBUG).

On a donc un affichage créé avec \$premier comme premier élément, la chaîne recherchée \$recherche passée en paramètre, le tableau de liens href correspondants aux icônes et la connexion à la base. Cette fonction appelle la méthode requête qui va sélectionner les éléments à afficher.

Enfin viennent les fonctions de pagination qui permettent la navigation entre les différentes pages du tableau.

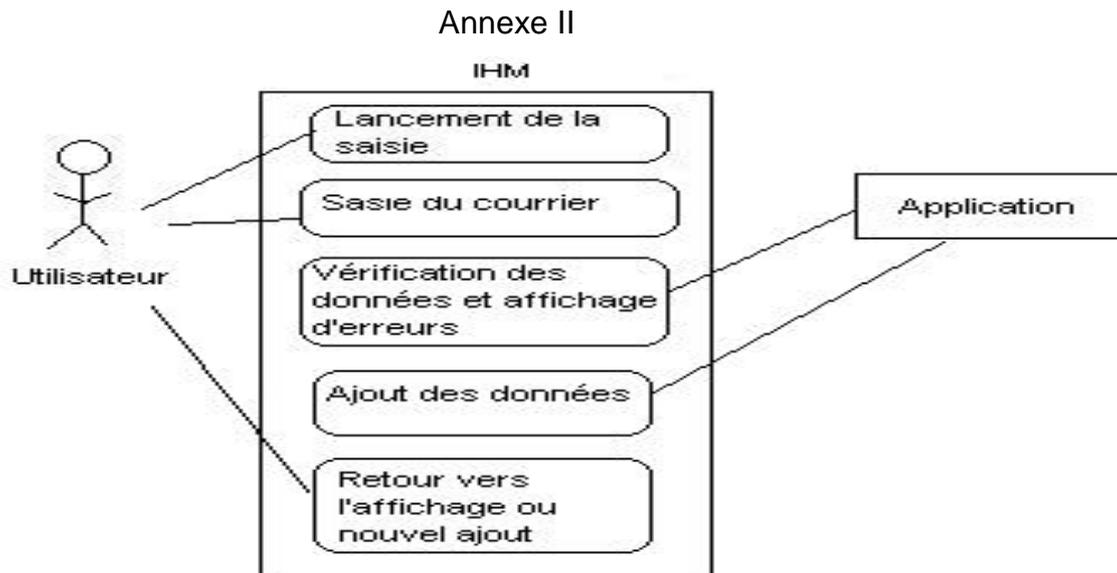
5 Présentation de l'application

5.1 Cas d'utilisation

5.1.1 Saisie de courrier

Le service reçoit un courrier;

L'utilisateur lance la saisie, la modification ou la suppression de courrier;
Le Formulaire est crée et adapté à la saisie de courrier;
L'utilisateur renseigne les champs du formulaire et valide;
En cas d'erreur, un message s'affiche;
Le courrier est ajouté dans la base;
L'utilisateur ajoute un nouveau courrier ou revient sur l'affichage de la table.

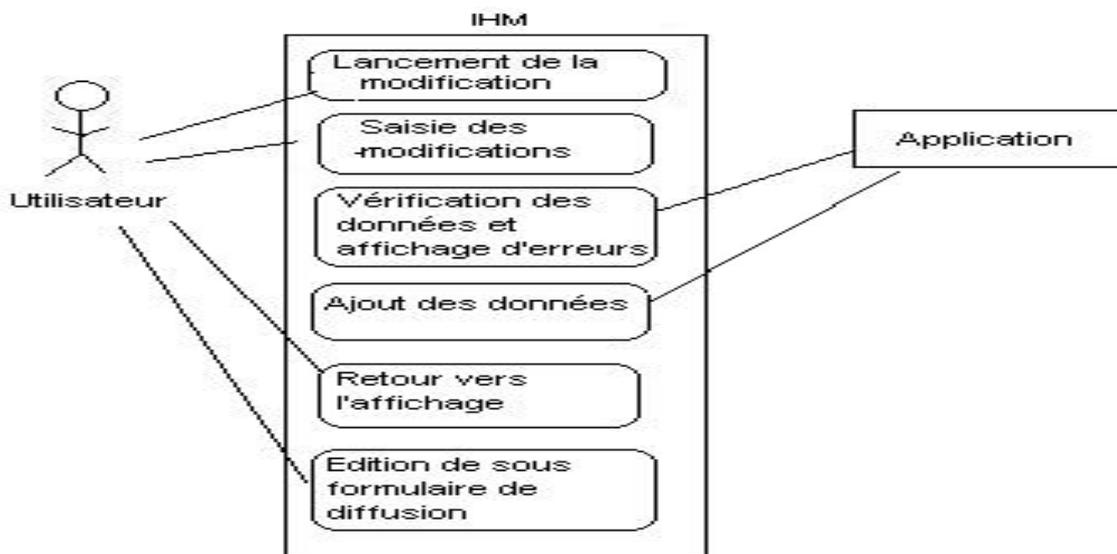


5.1.2 Modification

Ce cas est quasiment similaire à la saisie, la différence réside dans le fait que les champs du formulaire seront automatiquement remplis en fonction du courrier que l'on souhaite modifier.

De plus, lors d'une modification il est possible d'éditer des sous formulaires rattachés au courrier. Ces sous formulaires fonctionnent sur le même principe que les formulaires.

Annexe III

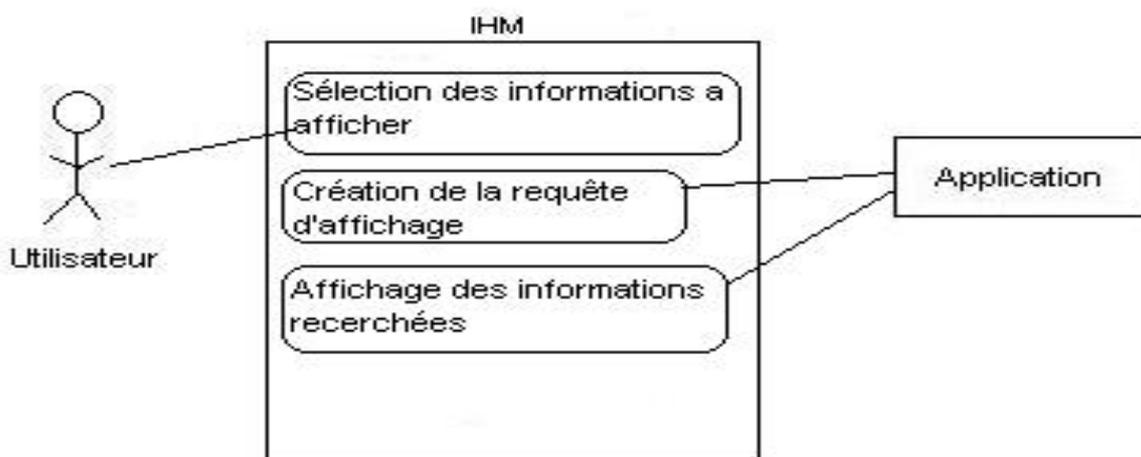


5.1.3 Suppression

De même, lors d'une suppression, les champs seront automatiquement remplis mais il ne sera pas possible de les modifier ; le seul choix laissé à l'utilisateur est la confirmation de la suppression ou le retour vers l'affichage de la table.

5.1.4 Recherche

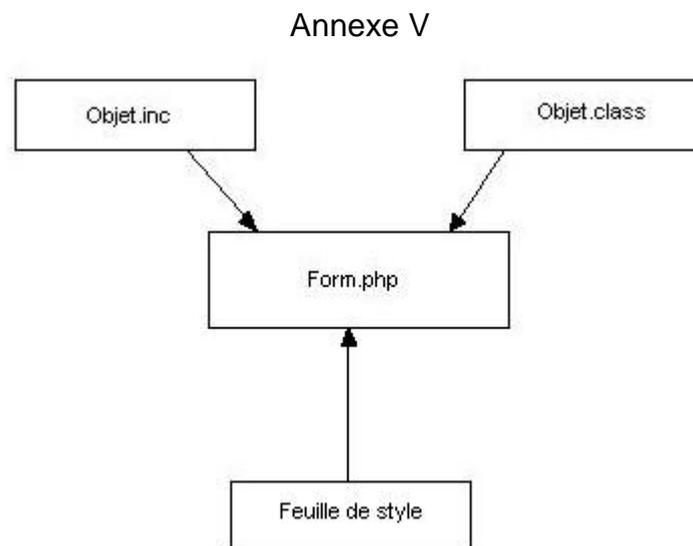
Annexe IV



5.2 Architecture et explication de l'application

5.2.1 Affichage d'un formulaire

Un formulaire est obtenu en surchargeant les méthodes de la classe dbForm



On remarque sur ce schéma que la création d'un formulaire qui a lieu au niveau de form.php nécessite l'inclusion de trois autres fichiers :

- La feuille de style ou **CSS**, elle contient les informations sur l'apparence de la page concernée;
- Objet.class, ce fichier contient les surcharges des fonctions de la classe open Mairie dbForm; c'est lui qui définit les types de champs qui vont apparaître, la possibilité ou non qu'aura l'utilisateur d'y entrer ou non des informations, il fait office de lien entre l'utilisateur et la base de données.
- Le fichier de paramétrage Objet.inc contient toutes les informations relatives à l'interaction avec la base de données telles que le paramétrage des requêtes, les liens correspondants aux icônes, la présence ou non de sous formulaires. Il est important de noter que chaque table de la base a un fichier.inc correspondant et qu'ils sont propres au **SGBD** utilisé.

Un exemple de formulaire est disponible en annexe VIII (8).

5.2.2 Affichage d'un sous formulaire

Les sous formulaires sont accessibles lors de la modification d'un courrier et sont présentés dans un tableau.

Les fichiers appelés lors de la création d'un sous formulaire sont les même que précédemment plus deux autres : `tab.class` et `soustab.class`.

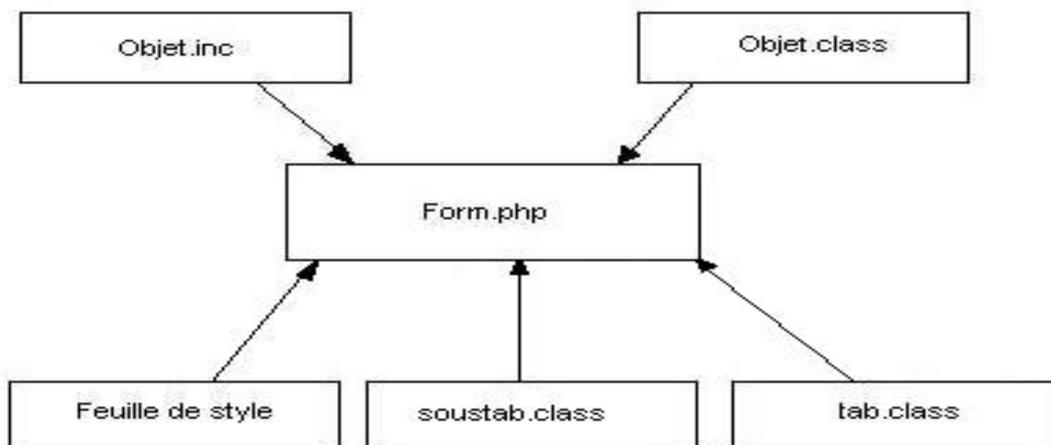
Dans le cas de la gestion du courrier, les sous formulaires représentent les tâches reliées à un courrier c'est pourquoi la table 'tâche' comprend un champ identifiant de courrier.

Un courrier est donc rattaché à plusieurs sous formulaires qui seront affichés par l'intermédiaire de `soustab` et la classe `tab` de `open Mairie`.

De plus, ces sous formulaires devant permettre de gérer la diffusion d'un courrier au sein d'un service, j'ai donc du modifier la table `service` en rajoutant un champ 'servicerattachement' ; ce champ permet de définir quels services seront sélectionnables lors de l'édition du sous formulaire en fonction du service auquel est destiné le courrier.

Les structures des tables courriers et service sont disponibles en annexe XII

Annexe VI



- `tab.class` est une des classes `open Mairie` décrite précédemment.
- `Soustab` permet l'affichage du tableau présentant les sous formulaires rattachés au courrier en cours d'édition.

Un exemple de sous tableau et de sous formulaire sont respectivement disponible en annexes IX (9) et X (10)

5.2.3 Recherche de courriers

La recherche s'effectue à partir d'une page développée sans utiliser les classes open Mairie; en effet, le moteur de recherche étant spécifique au courrier, il m'a été demandé de créer l'interface sans utiliser les classes open Mairie.

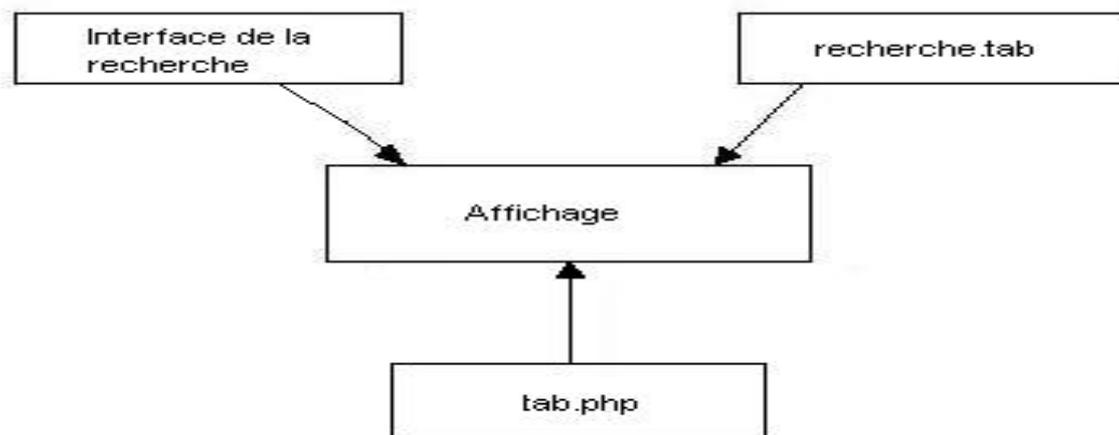
Ce moteur de recherche comprend quatorze champs différents qui permettent d'effectuer des recherches sur l'intégralité de la table courrier.

Quatre de ces champs sont remplis dynamiquement en fonction du contenu de la base, les autres sont de simples champs de saisie.

Une fois que l'utilisateur a validé ses choix, un fichier intermédiaire à la saisie des données et à l'affichage va préparer l'affichage.

Enfin, la classe d'affichage spécifique à la recherche affiche le contenu de la base compte tenu des valeurs entrées dans le moteur de recherche.

Annexe VII



On constate que pour effectuer une recherche, trois fichiers sont appelés

- L'interface de la recherche dans laquelle l'utilisateur va saisir les informations qu'il souhaite afficher ;
- Recherche.tab qui est une surcharge de la classe tab de open Mairie ; c'est à partir de ce fichier que sera créé l'affichage ;
- Tab.php qui sert d'intermédiaire entre les deux fichiers précédents ; il récupère les informations en provenance de l'interface de la recherche et les transmet à recherche.tab pour lui indiquer quels éléments sont à afficher.

Un exemple d'affichage est disponible en annexe XI (11).

6 Activités annexes

Au long du stage, en plus de l'activité de développement à laquelle j'ai été assigné, j'ai eu le privilège de participer à quelques activités supplémentaires.

Tout d'abord, j'ai accompagné Mr Nicolas Issart, chef du service administration système au centre technique municipal où j'ai pu assister à une négociation téléphonique entre le directeur du dit service et le directeur d'une société.

Ensuite, la période de mon stage correspondait à une période de changement du coeur de réseau faisant suite à un appel d'offres. A cette occasion, j'ai eu le plaisir d'accompagner au cours d'une après midi Mr Abdelouahed Djamel, Agent d'entretien au service téléphonie et réseau qui était alors chargé de faire migrer les câbles constituant la 'dorsale' du réseau de la ville de 100Mbits/s vers du Gigabits/s en fibre optique. J'ai ainsi pu mieux appréhender l'étendue du réseau de la Ville.

De plus, Mr Raynaud, mon tuteur m'a invité à assister à une de ses nombreuses présentations des logiciels open Mairie auprès d'une collectivité qui était intéressée notamment par la gestion des élections, des cimetières et des stocks.

Enfin, dans les derniers jours du stage j'ai entamé la création d'un générateur automatique de fichiers **CSS** à partir d'une image avec Mr Jean-Louis Bastide, technicien chef affecté au service développement. Pour ce faire, nous avons utilisé la librairie de traitement d'images intégrée à PHP baptisée GD qui contient les méthodes de base nécessaires à la manipulation d'images en 2D. Ce projet n'est actuellement pas abouti.

7 Problèmes rencontrés

Au cours du développement de l'application, j'ai rencontré plusieurs problèmes.

Pour commencer, comme dit précédemment, mon premier problème fût comprendre le code existant ainsi que l'architecture de l'application.

Ensuite, d'autres problèmes se sont posés a cours du développement tel que l'option 'magic_quote' de PHP ; cette option est à activer ou non au niveau du fichier PHP.ini et a pour but de protéger automatiquement les simples et doubles côtes dans un lien. Ayant cette option désactivée par défaut, je n'ai pas pensé à gérer le cas ou elle l'était ce qui a eu pour conséquence un plantage de l'application lorsque Mr Raynaud l'a testée sur sa machine.

Troisièmement, l'application devait pouvoir tourner à la fois sous Mozilla Firefox et Internet Explorer qui sont les deux navigateurs les plus utilisés. Ayant développé exclusivement sous Firefox, j'ai eu quelques soucis lors de l'adaptation à Internet Explorer.

Ensuite, lors du développement du moteur de recherche je devais pouvoir conserver les paramètres représentant les choix de recherche de l'utilisateur lors de la navigation entre les pages ; pour ce faire j'ai dû transformer le tableau contenant les paramètres en utilisant trois fonctions différentes : une sérialisation, un encodage au format url et une protection des quotes.

De plus, il était impossible de faire des requêtes imbriquées sous mysql, j'ai donc dû passer par des jointures pour pallier à ce problème.

Concernant le générateur de fichiers CSS, PHP n'étant pas adapté à la manipulation d'images nous avons donc rencontré des problèmes pour adapter les méthodes de la librairie GD au type de traitements que l'on souhaitait réaliser.

Enfin, un problème moindre fut le log d'erreur de PHP qui est peu explicite, ce qui est problématique au début.

8 Conclusion

Les fonctionnalités demandées ont été intégrées à l'application.

La réalisation de ce projet m'a permis de travailler par moi-même et m'a surtout appris à me conformer aux souhaits des utilisateurs ainsi qu'à la façon de travailler en entreprise.

De plus, j'ai pu participer au développement de logiciels libres, ce que je considère comme étant un privilège.

Au final, j'ai vécu une très bonne expérience à l'occasion stage aussi bien dans la façon de travailler que dans l'ambiance agréable qui régnait.

Glossaire

DSIT – Direction des Services Informatiques et Télécommunications. C'est le service informatique de la ville d'Arles.

Open Mairie – Ensemble de classes développées en PHP ayant pour but de permettre de développer rapidement des applications web.

SGBD – Signifie Système de Gestion de Base de Données ; c'est un ensemble de programmes destinés à assurer la gestion et l'accès à une base de données.

DB PEAR – PEAR signifie "PHP Extension and Application Repository", c'est un ensemble de bibliothèques organisées sous forme de packages, le package DB est une couche d'abstraction de base de données.

CVS - Concurrent Version System. Les logiciels de type CVS servent à gérer les différentes versions d'un programme. <http://fr.wikipedia.org/wiki/CVS>

GFORGE - Gforge est une application libre d'aide au travail collaboratif, qui permet d'organiser et de faciliter le travail d'équipes de développement sur des projets logiciels. C'est un outil idéal qui permet le travail distribué entre plusieurs équipes réparties à plusieurs endroits, tout en respectant les contraintes de flexibilité des développeurs. Gforge intègre en un point uniques trois composants essentiels des outils de travail collaboratif :

- Outils de communication : forums web, listes de diffusion, système de publication de nouvelles ;
- Outils de suivi : suivi de bogues et de correctifs, suivi de tickets de support, gestion de tâches et de projets ;
- Outils de partage d'informations : gestionnaire de versions, bibliothèque d'éléments partagés, gestionnaire de documentation, publication de fichiers.

Tous ces outils sont gérés par une interface web, ce qui permet une utilisation simple sans nécessiter d'homogénéité des postes de travail. Ils sont également activables et désactivables à volonté, ce qui permet d'adapter le mode d'utilisation de Gforge au mode de travail existant.

ADULLACT – Association des Développeurs et des Utilisateurs de Logiciel Libre pour l'Administration et les Collectivités Territoriales

CSS – **Cascade Style Sheets**, c'est un fichier qui contient les informations sur les couleurs à utiliser, les polices de caractères, l'apparence des boutons, les types de bordures. Cette séparation de la présentation du style et le contenu des documents ont pour but de simplifier l'édition pour le web et la maintenance d'un site.

FPDF – signifie **Free PDF**, il s'agit d'une classe PHP permettant de générer des PDF à partir d'un code PHP.

Bibliographie

Portail de open Mairie.

<http://www.openmairie.org/>

Librairie Pear, le connecteur de base de données utilisé dans le cadre du projet.

<http://pear.php.net/>

GFORGE de l'ADULLACT.

<http://adullact.net/>

Codes hexadécimaux de quelques couleurs, pour les feuilles de style.

<http://perso.orange.fr/jmpetit/dossiers/webmaster/couleurshexa.htm>

Aide sur les différentes balises HTML.

http://www.laurent-bernat.com/html-balises.php3?quoi=recap_balises

Aide sur le SQL

<http://sql.developpez.com/>

Documentations sur les différentes fonctions PHP

<http://fr3.php.net/>

Site d'aide en français sur PHP

<http://www.phpfrance.com/>

Forums d'aide sur PHP

<http://php.developpez.com/>

<http://www.developpez.net/forums/forumdisplay.php?f=25>

Annexes

Annexe VIII – Exemple de formulaire

The image shows a web form titled "Courrier tous services". The form is organized into several sections:

- Registre:** Includes a "Nature" dropdown menu with the selected value "Votre choix".
- Dates:** Contains two date input fields: "Arrivée le" and "Courrier du".
- Emetteur:** A section for sender information with fields for "Code", "Nom", "Adresse", "CP", and "Ville". Each field has a green arrow icon to its right.
- Objet Courrier:** A large text input field for the subject.
- Type Courrier:** A dropdown menu with "Votre choix" selected.
- Traitement:** A dropdown menu with "Choix d'un service" selected.
- Diffusion:** A section with a dropdown menu "Choix succès / ce service" and a large text input field.
- Pièce Jointe:** A text input field for attachments.

At the bottom of the form, there is a green button labeled "Ajouter Courrier" with a right-pointing arrow icon.

Annexe IX – Exemple de sous tableau

Type Courrier **Votre Choix** ▼

Traitement **33500 DIRECTION INFORMATIQUE** ▼

[Diffusion]

Choix successif de services ▼

33500 DIRECTION INFORMATIQUE
33000 DIRECTION ADMINISTRATION

Pièce Jointe

Modifier Courrier ↩

tache

1 - 1 enregistrement(s) sur 1

Création	Tache	Courrier	Serviceexecution	Datebutoir	Dateexecution	Solde	Chronoreponse
Supp X	1	37771	69874	0000-00-00	0000-00-00	non	789

tache

Annexe X – Exemple de sous formulaire

[ID]

Courrier No. 37771

Service de traitement **DIRECTION INFORMATIQUE**

[Dates]

Date butoir  Date d'execution 

Solde **non** ▼

Service d'execution **Choisissez un service** ▼

Observation

Scan Reponse  

Chrono reponse

Ajouter Tache ↩

Annexe XI – Exemple d’affichage tabulaire

Courrier tous services				
1 - 2 enregistrement(s) sur 2			Courriers de type :	
Création	Courrier	Emetteur	Objetcourrier	Service
Supp X	Mise à Jour	37771 PREFECTURE des bouches du Rhone Route de ganges 13000 Marseille	reponse subvention informatique mediatheque	33500 DIRECTION INFORMATIQUE
Supp X	Mise à Jour	37772 SOUS PREFECTURE arles rue des arenes 13200 arles	Adhesion à l adullact	33500 DIRECTION INFORMATIQUE