Mon chère Stevens, je te propose que l’on reprenne depuis le début. Notre application a pour but la gestion des clefs MSDNAA. En ce moment on travail sur la partie backoffice, ce qui représente les points suivant :

* Enregistrement de clefs
* Consultation des clefs
* Distribution aux élèves

On conçoit ce backoffice afin que les professeurs puissent s’occuper : d’enregistrer de nouvelles clefs sur l’application ; de consulter les clefs disponibles ainsi que de réagir vis-à-vis des élèves (limite de clefs atteinte, liste des clefs reçu, etc.). Bref tu l’auras bien compris, cette application représente un modèle MVC (Modèle-Vue-Contrôleur) plutôt compliquée.

Je vais donc aborder les trois points listé ci-dessous pour que tu comprennes bien tout ce que l’on fait, puis, on abordera cela d’une manière plus « technique ».

Premièrement, l’enregistrement des clefs, comme son nom l’indique signifie qu’une partie des utilisateurs, les professeurs, vont entrer des clefs dans l’application. Je sais que la question te démange alors que je vais t’y répondre : les professeurs, lorsqu’ils font-une requête sur MSDNAA pour avoir un lot de clefs, reçoivent un fichier sous le format XML. Ce fichier est en quelque sorte l’homologue de la base de données. Je m’explique, une base de données stock les informations dans des tables, bien, jusqu’à là je ne t’apprends rien (et tant mieux si dans la suite non plus !). Le XML quant à lui, permet la création de ses propres balises. Exemple :

*<Salut>*

*<Ca>*

*<Va>*

*</Va>*

*</Ca*

*</Salut>*

Je suis sûr que tu as remarqué que je n’ai pas mis le : ‘ ?’, c’est une question rhétorique… :P Je ne sais pas s’il est possible surtout de faire la balise *< ?> </ ?>* à voir !

Ces balises permettent de contenir l’information, il serait alors possible de faire comme une base de données à la différence qu’il n’existerait pas de lien entre les tables. *(Cf Clef étrange)*

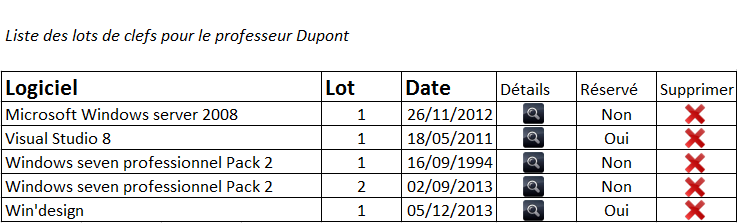
Bref, le XML c’est pour certains « le futur », car il a évolué, il y a en quelque sorte une Version 2.0 (genre le C a donné le C++/C croisillon quoi). Rien de plus la dessus … pour le côté technique, tu peux aller voir la succession de lien depuis l’application pour la première approche (même si ce n’est pas finis), puis débuter dans le conteneur à la ligne 83 : on reçoit UN fichier en ‘point’ on ne sait pas encore quoi, l’enquête s’ouvre, les contrôles s’intensifient, le traitement arrive. Sera tu relever le défi ?

Plus sérieusement, je pense que ce serait une bonne idée que tu observes tous les traitements qui se font et que tu les comprennes. Si tu veux le code est dé-commenté & il y a seulement les ‘//’ en pièce jointe, tu pourras vérifier sur la version que tu as !

Deuxièmement, c’est le point où l’on va le plus insister puisque c’est ta partie, la consultation des clefs. Quèsaco ? Et bien c’est le résultat d’une recherche par critères intuitifs, en effet si par exemple tu demandes les clefs enregistrées par l’enseignant X, il est alors logique que tu t’attendes à recevoir la liste des clefs enregistrées par l’enseignant, mais pas seulement ! Si tu fais cette recherche, c’est que tu as besoin d’une information relative à la donnée entrée. Par exemple si tu rentres un enseignant c’est que tu as besoin de savoir qu’elles clefs ont étés enregistrées par cette enseignant, quand, en reste-t-il de disponible, ont-elles encore une existence dans l’infrastructure ? Tu dois absolument te demander, Pourquoi ce critère, sur quoi il débouche, qu’est-ce que l’on recherche en le saisissant. De là tu ressors une liste de critères de recherche :

* Le logiciel
* Le professeur

Les réponses aux questions que tu t’es posé représente ce que tu vas afficher dans le tableau de la réponse. Ainsi si l’utilisateur demande seulement, en critère, un professeur il va s’attendre à avoir comme résultat :



Bien sûr comme tu peux l’imaginer lorsque tu clics sur Détails tu obtiens la liste des clefs en référence à ce lot.

Il te reste ensuite deux cas à traiter : si l’on souhaite renseigne un logiciel et, il ne faut pas l’oublier, si l’on choisit un logiciel pour un prof.

L’affichage c’est bien beau, mais il faudrait maintenant savoir ce qu’il faut pour faire tout ça.

Ah ouais après relecture je me rends compte que je boycotte la partie « élèves » mais bon, on verra ça plus tard.

Commençons cette partie technique, je vais lister les interactions que l’application fait pour pouvoir afficher ce que l’on souhaite :

* Tout part du contrôleur, nous somme d’accords. Mais qu’est-ce qu’il fait exactement ?! Et bien il fait simplement une insertion, d’un résultat d’une méthode à travers la propriété **$monControleur** dans une variable globale puis fait appels à la vue correspondante.
  + Cela sous-entend que nous nous dirigeons dans GestionConteneur.php. A quoi il sert celui-là, et beh c’est que récemment que je me suis rendu compte que le contrôleur permettait de faire le lien entre le traitement et la vue. Et que le gestion main permet de faire le lien entre le traitement simplifier du contrôleur et le conteneur où le traitement doit être exécuter
* Du coup tu as du bien le comprendre que dans GestionConteneur on retrouve la fonction que nous avons appelé dans le contrôleur qui fais un **return** de la fonction définis dans le conteneur Lot Clef via la propriété **$tousLesLotsClefs**.
  + Aller direction ConteneurLotClef.php, *« C’est qui ce tar-ba ?! »* Simplement celui qui contient l’ensemble des Lots de clefs, très bonne déduction Stevens, c’est une liste d’objets ! Et oui tout à fait on exécute tous les traitements ici (enfin, ceux qui sont liés aux lot clefs hein ;) ).

***HERE WE ARE***

Ok le traitement, traitement, hummm traitement, alors tu te rappels qu’on a des cas d’utilisation : 1 ou 2 ou 1 et 2. Et bien c’est ici que tu vas écrire c’est cas mais il faut savoir dans quel ordre, sinon tu risques de mal imbriqué tes « IF » et ainsi ne pas avoir les données qu’il te faut. Je te laisse réfléchir le temps d’une page

……………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………..…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

Bien joué ! Ou pas :P Moi je pense (et j’en suis sûr d’ailleurs) qu’il faut choisir la combinaison :

**Si 1 ET 2**

Affichage des lots de clefs pour le logiciel Y et le professeur X

**SINON**

**Si 1**

Affichage des lots de clef pour le logiciel Y

**SINON**

**SI 2**

Affichage des lots de clefs pour le professeur X

**SINON (rien n’a été choisi)**

$\_SESSION['erreur']="Vous devez selectionner au moins un critère de recherche";

$\_SESSION['redirection']="index.php?vue=consultationClef&action=Recherche";

include ('vues/erreur.php');

return $LeResultatQuiEstLeTableauQueJaiEssayeDeModeliserPointDExclamationAhAhPointVirguleParenthèseQuiSeFerme

Mais là je te troll un petit peu, car tu sais si rien n’a été choisie, et bien ce test-là tu peux le faire depuis le contrôleur et ainsi exécuter moins de traitement en faisant : si aucun critère de recherche erreur sinon traitement. Du coup tu peux **supprimer** le dernier **Sinon Si 2** puisque ce sera forcément le deux si ce n’est pas le un.

Maintenant que nous avons toutes l’architecture et le chemin que nous allons faire, il nous manque plus que le traitement final : comment trouver les lots de clefs du professeur Dupont (Et oui désolé tu vas quand même faire deux cas tout seul comme un grand !). Rien de plus simple : Tu es dans le conteneurLotClef, tu as une liste d’objet LotClef. Tu vas donc parcourir tous les Lots de clefs et insérer dans un tableau les résultats correspondant au critère de recherche. Ça devrait donner un truc du style :

J’utilise la variable instanciée au début de ma fonction pour créer le tableau qui correspond au résultat escompté. Je parcours tous les Lots de clefs. Si le lot courant a été insérer par le professeur passer en paramètre je l’ajoute dans ma variable contenant le tableau. Sinon je prends le lot suivant. Une fois que je suis sortie de ma cascade de If je pense à refermer mon tableau……………

Bien sûr tu te rends compte du problème la non ? Réfléchie qu’est ce qui peut t’arriver ? Aller dis-moi ça devrait te sauter à la tête !! Même pas j’ai de l’espoir …………. Aller un petit indice quand même tu dois créer une variable qui reçoit False, et qui à partir du moment où tu rentres dans un des Ifs de dernier niveau (‘Si le lot courant a été insérer par le professeur […]’) il faut passer cette variable en True. Alors ?

Là j’ai de l’espoir ! Au cas où : si la recherche ne donne rien, genre Mister Dupont il n’a jamais rentré de clefs, et bien il faut que tu ajoute un contrôle sur le booleen cité ci-dessus tel que : s’il y a un résultat alors on ferme le tableau sinon on renvoie simplement un message adressé à l’utilisateur du style : « Sorry Bro but we didn’t find anything about your criterion. ».

Pour résumer : cette fonction reçoit deux paramètres (dont l’un des deux peux être égale à « »). On concatène les résultats grâce au « .= » sans oublier qu’on est en train de faire un tableau qu’il faut préalablement dessiner pour chaque cas. Et on finit par tester la présence de résultat(s) !

Je pense que tu devrais avoir tout compris maintenant, il ne te reste plus qu’à coder tout ça en 5/10 minutes ;)

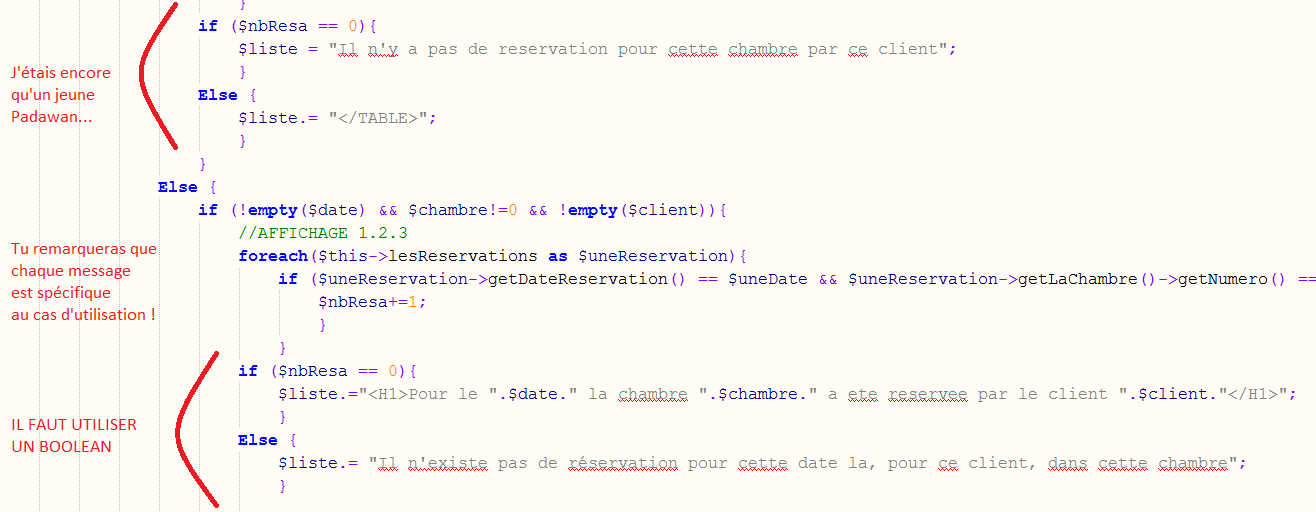
Bon courage, bonne vacances :\*

₷imon Ɲussbaum

Ps : Vu que je suis un mec gentil je te donne un exemple :



[…………..]



Return $liste ;)