

Ali QUENAHAT

Thibault ROY

Niveau : *TC04*

Rapport LO07 :

Genutt:

*Attribution des conseillers à l’UTT*

UV : LO07

Responsable : Marc LEMERCIER

Semestre :Printemps 2014

Sommaire

Introduction3

1. Cahier des charges4
2. Conception de la base de données5
3. Schéma de la base de données6
4. Choix des tables6
5. Modèle physique des données8
6. Conception du site9
7. Design9
8. Implémentation des fonctionnalités10

Conclusion14

Bibliographie15

# 

# Introduction

L’UV LO07 nous a été autorisé durant notre semestre de printemps 2014 à l’université de technologie de Troyes en tant que TM de branche accessible aux Troncs Communs. Nous l’avons choisie afin d’avoir les bases nécessaires concernant le développement web.

Dans le cadre de cette UV, nous avons l’opportunité de concevoir et développer un outil web permettant de gérer les conseillers à l’UTT en passant de l’analyse à la conception. Le DRH, les responsables de programme et le service scolarité disposeront chacun d’un compte pour pouvoir gérer l’interface. Ce projet a pour but de nous familiariser avec le développement web et nous apprendre à travailler en équipe sur ce type de projet tout en respectant les contraintes fixées par le cahier des charges.

Dans ce rapport, nous allons exposer le cahier des charges et expliquer la création de la base de données ainsi que sa structure pour pouvoir ensuite présenter l’ensemble des étapes de la conception du site en analysant l’arborescence du site, l’organisation du code et le fonctionnement du site.

1. **Cahier des charges**

Voici la liste des fonctionnalités que nous devions développer :

* DRH
  + Vider la liste des enseignants-chercheurs
  + Ajouter un enseignant-chercheur
  + Ajouter une liste d’enseignants-chercheurs
  + Supprimer un enseignant-chercheur
  + Visualiser les enseignants-chercheurs
  + Visualiser les enseignants-chercheurs avec les étudiants conseillés
  + Visualiser les enseignants-chercheurs avec le nombre d’étudiants conseillés dans l’ordre décroissant
* Responsable de programme
  + Ajouter une habilitation
  + Supprimer une habilitation
  + Ajouter les habilitations par pôle
  + Visualisation les étudiants de son programme avec leurs conseillers
* Service scolarité
  + Vider la liste des étudiants
  + Ajouter un étudiant
  + Ajouter une liste d’étudiant
  + Supprimer un étudiant
  + Attribuer un conseiller à nouvel étudiant
  + Attribuer des conseillers à tous les nouveaux étudiants
  + Déclarer le transfert d’un étudiant du TC vers un programme
  + Visualiser les étudiants selon le programme
  + Visualiser les étudiants sans conseiller
  + Visualiser les étudiants avec leurs conseillers
  + Visualiser les conseillers selon l’ordre décroissant du nombre de conseillés

Le projet doit utiliser les éléments qui suivent :

* un serveur Apache
* des scripts PHP
* des pages HTML
* des feuilles de styles CSS
* des instructions Javascript
* une base de données relationnelle
* des variables de session

Les éléments optionnels que nous avons mis en place :

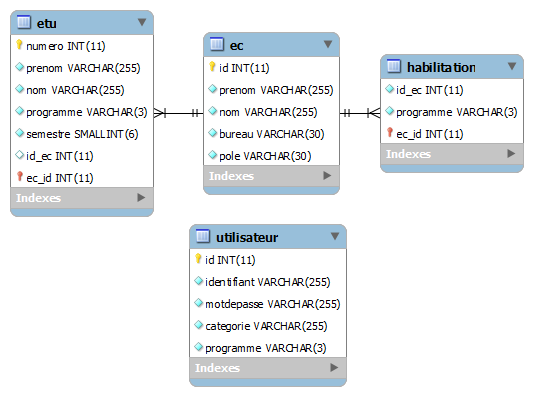
* une architecture MVC
* un framework CSS (BootStrap)

1. **Conception de la base de données**

La création de la base de données a été une étape très importante car elle stockera toutes les informations du site. Le site web va recevoir des informations en provenance de la base de données pour les afficher aux utilisateurs. Il faut que la base de données soit la plus optimisée possible afin de permettre au site d’accéder aux informations rapidement et facilement. La structure est très importante car la base de données doit prendre le moins de place possible de manière à économiser de l’espace de stockage en évitant la redondance d’informations.

Après avoir décortiqué le sujet et fait plusieurs schémas sur papier. Nous avons souhaité choisir la base de données contenant le moins de table. Pour mieux comprendre la structure de notre base de données, nous l’avons modélisé grâce au logiciel MySQL Workbench.

1. **Schéma de la base de données :**



1. **Choix des tables**

**ec**(id; prenom; nom; bureau; pole) **:**

Cette table contient les informations des enseignants-chercheurs qui sont les prénoms, noms, bureaux et pôles. Maintenant nous allons détailler le champ « id ». Ce champ s’auto-incrémente, c’est-à-dire qu’à chaque insertion un id incrémenté lui est donné en valeur. Une clé primaire est associée à ce champ. Une clé primaire permet d'identifier d’une façon unique un enregistrement dans une table. Une clé unique est associée au champ bureau car il n’y a qu’un bureau par enseignant-chercheur, c’est-à-dire que deux enseignant-chercheurs ne peuvent pas avoir le même bureau. Cette table est liée à la table « etu » et « habilitation ».

**etu**(numéro;prénom;nom;programme;semestre;id\_ec):

Cette table contient les informations des étudiants. Chaque étudiant a un numéro unique, un couple de prénom et de nom unique, un programme et un semestre.Une clé primaire est associée au champ « id » de la table. Le champ « id\_ec » est une clé étrangère qui se réfère à l’id de la table « ec » et vaut NULL par défaut, c'est-à-dire lorsque l’étudiant n’a pas de conseiller. La valeur de ce champ prend l’id du conseiller seulement au moment de l’attribution. Cette table est donc liée à la table « ec ».

**habilitation** (id\_ec; programme) :

Cette table va permettre de contenir les habilitations. Chaque enseignant-chercheur va être associé à un programme s’il est autorisé à conseiller les étudiants de ce même programme. Le champ « id\_ec » est une clé étrangère se référant au champ « id » de la table « ec ». La table « habilitation » est donc liée à la table « ec ».

**utilisateur** (id; identifiant; motdepasse; categorie; programme) :

Cette table contient les informations des utilisateurs de l’outil web et donc n’est pas liée aux autres tables. Cette table n’est pas demandée explicitement par le cahier des charges mais elle permet d’identifier les droits de chaque utilisateur. Grâce aux champs « identifiant » et « motdepasse », nous vérifions si l’utilisateur a les droits nécessaires pour effectuer la gestion. Il y a plusieurs catégories d’utilisateurs qui sont les responsables de programme, le service scolarité et la direction des ressources humaines. Une clé primaire est associée au champ « id ». Le champ « motdepasse » contient les mots des passes cryptés en sha1 pour empêcher la lecture des mots de passes. La valeur du champ « programme » est à NULL par défaut, si c’est un responsable alors la valeur doit être modifiée avec le nom du programme.

Donc au final, nous avons vraiment tâché de faire le moins de table possible tout en évitant la redondance d’informations. Nous avons créé la table de façon à pouvoir faire des jointures SQL qui permettront donc de simplifier les requêtes et alléger la base de données.

1. **Modèle physique des données**

--

-- Structure de la table `ec`

--

**CREATETABLEIFNOTEXISTS** `ec` **(**

`id` **int(**11**)NOTNULL** AUTO\_INCREMENT**,**

`prenom` **varchar(**255**)NOTNULL,**

`nom` **varchar(**255**)NOTNULL,**

`bureau` **varchar(**30**)NOTNULL,**

`pole` **varchar(**30**)NOTNULL,**

**PRIMARYKEY(**`id`**),**

**UNIQUEKEY** `bureau` **(**`bureau`**)**

**)** ENGINE**=**InnoDB**DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**1**;**

-- --------------------------------------------------------

--

-- Structure de la table `etu`

--

**CREATETABLEIFNOTEXISTS** `etu` **(**

`numero` **int(**11**)NOTNULL** AUTO\_INCREMENT**,**

`prenom` **varchar(**255**)NOTNULL,**

`nom` **varchar(**255**)NOTNULL,**

`programme` **varchar(**3**)NOTNULL,**

`semestre` **smallint(**6**)NOTNULL,**

`id\_ec` **int(**11**)DEFAULTNULL,**

**PRIMARYKEY(**`numero`**)**

**)** ENGINE**=**InnoDB**DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**1**;**

-- --------------------------------------------------------

--

-- Structure de la table `habilitation`

--

**CREATETABLEIFNOTEXISTS** `habilitation` **(**

`id\_ec` **int(**11**)NOTNULL,**

`programme` **varchar(**3**)NOTNULL**

**)** ENGINE**=**InnoDB**DEFAULT** CHARSET**=**latin1**;**

-- --------------------------------------------------------

--

-- Structure de la table `utilisateur`

--

**CREATETABLEIFNOTEXISTS** `utilisateur` **(**

`id` **int(**11**)NOTNULL** AUTO\_INCREMENT**,**

`identifiant` **varchar(**255**)NOTNULL,**

`motdepasse` **varchar(**255**)NOTNULL,**

`categorie` **varchar(**255**)NOTNULL,**

`programme` **varchar(**3**)NOTNULL,**

**PRIMARYKEY(**`id`**)**

**)** ENGINE**=**InnoDB**DEFAULT** CHARSET**=**latin1 AUTO\_INCREMENT**=**4**;**

1. **Conception du site**

Nous ne jugeons pas utile de mettre les images de chaque partie du site pour décrire les fonctions, car nous ne pensons pas que c’est le but du rapport. Nous préférons expliquer le déroulement de la réalisation du site et ensuite le fonctionnement.

1. **Design du site**

Tout d’abord, nous avons décidé de nous occuper du design du site web avant de commencer à coder les fonctionnalités. Après plusieurs recherches, nous avons trouvé plusieurs frameworks CSS. Mais c’est BootStrap qui nous a convaincu, nous avons donc à l’aide de BootStrap et d’un tutoriel disponible sur internet conçu le design du site web. Notre ligne de conduite étant la professionnalité et la simplicité, ce framework nous a ainsi fait gagner énormément de temps pour un résultat dont nous sommes pleinement satisfaits. Nous avons fait plusieurs essais pour arriver à ce résultat concluant, nous avons notamment ajouté un menuayant un effet particulier avec un plugin JavaScript.

Nous avons utilisé les langages SQL, HTML, CSS et JavaScript. Nous avons aussi utilisé la bibliothèque jQuery puisque Bootstrap fournit plusieurs composants sous forme de plugins utilisant cette bibliothèque. Nous avons fait attention à faire un site extensible qui permet donc l’utilisation sur tablette, smartphone et ordinateur.

Grâce à ceci, nous avons pu concevoir le gabarit du site web tout en respectant les normes HTML5. Il ne nous restait donc à coder les fonctionnalités du site web.

1. **Implémentation des fonctionnalités**

**Outils utilisés :**

Nous avons utilisé le logiciel wamp pour pouvoir tester notre site en local et utilisé Github pour faciliter le travail d’équipe. Le logiciel NetBeans nous a été utile tout le long pour ses fonctionnalités propre à elle-même (formatage du code etc…).

**Organisation du code :**

Pour cette partie, il a d’abord fallu penser à l’organisation du code. Nous avons pensé au départ à utiliser le framework PHP Symfony 2 mais avons vite compris que ce n’était pas la peine de l’utiliser car les profits que nous pouvions en tirés étaient faibles.

Le projet étant par équipe, il a fallu trouver une organisation adéquate pour simplifier le travail à plusieurs. Il nous a donc paru évident d’organiser le code selon l’architecture MVC orienté objet. L’architecture MVC (Modèle – Vue – Contrôleur) permet de bien organiser son code source et de connaître facilement le rôle de chaque fichier. L’objectif étant de séparer le code en trois parties. Ainsi nous avons pu travailler en même temps sur des fonctionnalités différentes du site.

* Modèle : Il doit aller chercher les informations dans la base de données et les organiser pour ensuite les envoyer au contrôleur. C’est là qu’on a utilisé le langage SQL.
* Vue : Il gère la partie affichage du site. Il contient essentiellement du code HTML. La vue quant à elle permet d’afficher les informations envoyées par le contrôleur.
* Contrôleur : Il fait le relais entre le modèle et la vue. Il demande des données au modèle, faire des traitements dessus et puis il renvoie ce qu’il faut afficher à la vue.

Grâce à un tutoriel disponible sur internet, nous avons pu comprendre le fonctionnement de ce principe et le mettre en place.

**Fonctionnement général :**

Le premier fichier à être appelé lors de la consultation du site est « index.php » qui se trouve donc à la racine du site. Il instancie le routeur puis lui fait router la requête. Voici le code pour mieux comprendre :

**require**'Controleur/Routeur.php';

$routeur=**new**Routeur();

$routeur->routerRequete();

Ensuite le contrôleur frontal qui est Routeur.php analyse la requête entrante pour déterminer l'action à entreprendre. Et selon l’action il fera appel à la méthode du contrôleur correspondant. Voici un petit morceau de code comme exemple :

**if**($\_GET['action']=='formchoixprogramme')

{

$this->ctrlEtudiant->choixProgramme();

}

Dans cet exemple, si le service scolarité souhaite changer le programme d’un étudiant, alors on appelle la méthode associée au contrôleur. Ce dernier, fera donc intervenir le modèle et la vue correspondant.

Dans le contrôleur frontal, on vérifie à chaque fois si l’utilisateur a le droit d’effectuer l’action afin d’éviter que tout le monde puisse y accéder. On gère donc les droits d’accès.

La page gabarit.php qui se trouve dans la vue contient le gabarit du site, c’est donc cette page qui va afficher la majeure partie du code html. De cette façon il est facile de changer le design du site lorsqu’on le souhaite puisqu’il n’y aura principalement que cette page à modifier.

**Gestion des erreurs :**

Nous avons pensé à la gestion des erreurs, et pour cela nous avons créé une méthode qui va s’occuper de ça. Voici comment on déclenche une erreur :

**thrownew**Exception("Cet étudiant a déjà été ajouté.");

On envoi en paramètre le message d’erreur et ensuite grâce à l’instruction catch, on transmets les erreurs à la méthode « erreur ».

Grâce à cette méthode qui se trouve dans le contrôleur frontal, nous générons une vue qui va afficher l’erreur avec le message d’erreur approprié :

**privatefunction**erreur($msgErreur){

$vue=**new**Vue("Erreur");

$vue->generer(**array**('msgErreur'=>$msgErreur));

}

**Arborescence du répertoire du site :**

* Contenu/
  + css/
  + img/
  + js/
* Modele/
* Controleur/
* Vue/

Dans le dossier Contenu se trouve les fichiers css, javacript et images liés au design.

**Utilisation du site :**

Lorsque l’utilisateur arrive sur la page d’accueil, nous lui proposons de se connecter pour pouvoir disposer de ses droits. Chaque service a un compte pour se connecter. Pour un DRH, il dispose du compte drh avec comme mot de passe drh. Le service scolarité dispose du compte « scolarite » avec comme mot de passe drh et enfin le responsable du programme ISI peut s’identifier grâce à son nom identifiant et qui est en l’occurrence Lemercier et son mot de passe drh. Ce sont juste des comptes servant d’exemple pour l’utilisation du site. Une fois identifié, l’utilisateur peut jouir de ses droits et peut procéder à des actions grâce à la barre de menu situé sur le haut.

**La réalisation :**

Tout au long de la réalisation du site, nous avons demandé l’avis de nos amis concernant la fluidité du site, nous aussi fait beaucoup de test pour voir les bugs qui subsistaient. Nous nous sommes aperçus que cette méthode a été très efficace pour trouver rapidement les bugs. Nous avons aussi testé nos requêtes directement sur le PHPMyAdmin de wamp avant d’implémenter la méthode dans le modèle. Cette dernière astuce nous a permis un gain de temps non négligeable. Grâce à la communauté anglaise et française présente sur internet, nous avons surmonté toutes les difficultés rencontrées lors de la réalisation.

**Quelques images du site :**

Voici une image du site sous un compte du service scolarité lorsqu’on pointe sur l’action « Ajouter » :

# site.png

# Voici un exemple d’affichage d’une liste d’étudiant sans conseillers :site2.png

Il est à préciser qu’au passage de la souris sur une ligne du tableau, la couleur de fond change. Ce sont certes des petits plus, mais cela contribue à l’ergonomie et au final rend le site agréable à l’utilisation.

**Conclusion**

Ce projet nous a permis d’avoir lune première expérience en développement d’outil web et de nous rendre compte de la manœuvre à suivre liée à la réalisation de ce type de projet. Il nous a permis d’apprendre à nous documenter sur internet pour pouvoir trouver les solutions à nos problèmes. Notre expérience concernant le travail en équipe dans un projet informatique nous aidera sûrement dans le futur.

Nous avons respecté le cahier des charges en pensant toujours à avoir un code indenté et propre. Nous avons utilisés l’architecture MVC (orienté objet) pour avoir un site maintenable et évolutif.

Pour finir, nos objectifs ont étés atteint, le projet est achevé et les bases en développement web sont acquises. Dans un souci de bien faire, le projet nous a pris du temps mais nous avons apprécié la réalisation de ce site internet. Conformément au cahier des charges, seules les fonctionnalités ont été développées mais nous aurions pu mettre en place un système permettant de se souvenir de l’utilisateur (cookies) qui est très simple à implémenter.

**Bibliographie**

**Lemercier, Marc**. Cours LO07. 2014.

Concevez votre site web avec PHP et MySQL [en ligne]. Disponible sur : http://fr.openclassrooms.com/informatique/cours/concevez-votre-site-web-avec-php-et-mysql