

## Projet Programmation WEB PHP-MYSQL

### 1. But du projet

Créer un site WEB (intranet) pour regrouper les informations relatives aux anciens étudiants d'une formation afin de fournir une source d'informations aux étudiants actuels et permettre une communication entre anciens et nouveaux.

### 2. Orientations souhaitées

a) Il y a trois types d'utilisateur du site :

1. l'administrateur (un enseignant) qui outre le fait d'avoir fait la page d'accueil du site, valide les données saisies par un ancien étudiant avant de publier sa page (page créée ou page modifiée) , on suppose que l'ancien est prévenu par téléphone si sa page n'est pas acceptée;
2. un ancien étudiant :
  - a. il propose des informations le concernant selon différents thèmes :
    - informations personnelles,
    - s'il travaille : descriptif de l'organisme où il travaille et thème sur lequel il travaille,
    - conseil aux étudiants actuels ;
  - b. il peut apporter des modifications;
  - c. il peut ajouter 1 ou plusieurs photos ou en enlever ;
  - d. il peut poster des commentaires sur la formation, accessibles depuis la page d'accueil ;
3. un visiteur (un enseignant, un étudiant ancien ou actuel) peut consulter ce qui est publié et éventuellement mettre des commentaires.

Les types 1 et 2 possèdent un compte (déjà créé pour l'administrateur, créé par l'ancien étudiant quand il crée sa page) et un mot de passe.

b) Sur la page d'accueil du site, se trouvent :

1. la présentation du site ;
2. l'accès à la liste des anciens (par promotion ou par nom), à partir de laquelle il est possible d'accéder à la page de l'un d'eux ;
3. le lien permettant de voir les commentaires selon deux modes de tri : par auteur (nom) ou par date (pour chaque commentaire on doit voir par qui il a été écrit, et quand il a été posté)
4. le lien permettant de saisir un nouveau commentaire et de le poster (il est laissé toute liberté aux auteurs, aucune vérification n'est faite par l'administrateur)

c) la page d'un étudiant doit avoir la même structure quel que soit l'étudiant, on peut y retrouver les 3 thèmes positionnés toujours au même endroit de la fenêtre. Certains peuvent ne pas être remplis. Un 4<sup>ème</sup> emplacement est réservé pour mettre des photos.

Une page consultée doit entrer dans un écran, mais il doit être possible de grossir un emplacement.

- d) Si un étudiant souhaite apporter une modification à un texte existant il faut le lui fournir préalablement.
- e) Pour la saisie d'un commentaire ou d'un conseil prévoir la possibilité de changer la couleur, de grossir la fonte, de mettre en gras ou de souligner le texte.
- f) Une fois créée, la page d'un ancien ne peut être supprimée que par l'administrateur. Tout ce qui le concernait est alors supprimé de la base. Ceci ne peut se faire que si la demande est faite par l'ancien.

## 2) Gestion d'une base de données :

Table **tancien** (idAnc, nom, prénom, numTel,  
idPromo, ←..... ex : GI2010, SI2008  
idOrgEmployeur, idThemeTravail, ←..... remplis s'il travaille, vide sinon  
idConseil, idPhoto (au plus 3)) ←..... peuvent ne pas être remplis

Table **ttheme** (idTheme, libelleTheme)

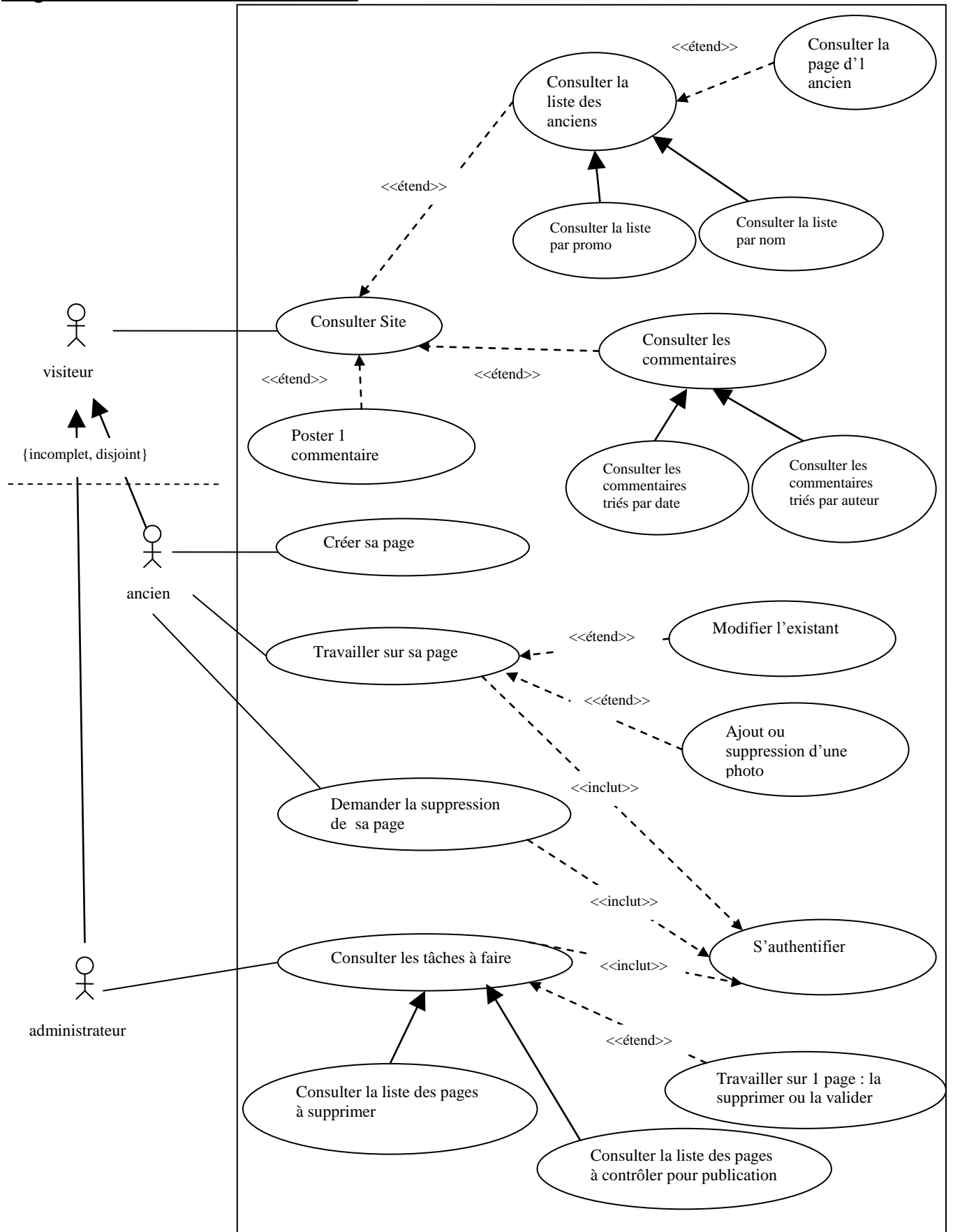
Table **tphoto** (idPhoto, cheminAcces, titre)  
Chaque photo est placée dans le répertoire photo.

Table **torganisme** (idOrg, libelleOrganisme, codePostal, ville)

Table **tcom** (idCom, libelleCom, idAuteur, dateCom)  
un conseil peut être vu comme un commentaire

Table <b>tcompte</b> (login, mdp)		
login	= admin	pour l'administrateur,
mdp	mot de passe qui doit contenir au moins une * et un chiffre et commencer par une lettre (à contrôler avec une expression régulière)	
	= a*d8min	pour l'administrateur)

Diagramme de cas d'utilisation du site :



### 3) Eléments de réalisation

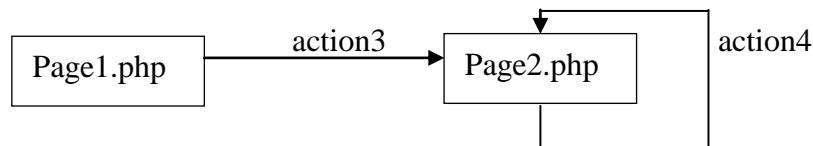
- a) Vous avez toute liberté de présentation. Utilisez le CSS.  
Définir une charte graphique pour chaque page écran que vous respecterez : partie entête, affichage, pied de page, ...
- b) Vous devez utiliser un fichier « css » pour la mise en forme du site et donc éviter de mettre du CSS dans une page HTML ou PHP.
- c) Vous devez avoir un fichier de création de la base nommé « creation.sql » dans lequel il y aura les commandes de création des tables ainsi que des commandes d'initialisation des tables.
- d) Prévoir un fichier texte **Lire avant utilisation** pour guider l'utilisateur (comment lancer l'application, quelles tâches sont à faire impérativement, ...). Vous devez vous souvenir que votre projet peut être utilisé sur différents serveurs.
- e) Pour lancer votre application vous devez avoir un fichier « index.php ».
- f) Vous devez faire en sorte de regrouper le plus possible les instructions dans un même fichier PHP afin de ne pas multiplier les pages HTML ou PHP.

Ex d'organisation d'un fichier PHP:

<u>Contrôles :</u> Existence de données, forme de données, tests de contraintes d'intégrité, autorisations, ....
<u>Traitements</u> Utilisation d'alternatives de traitements, ...

- g) Faites en sorte de ne pas répéter un même traitement, utiliser la fonction include ().
- h) Constituer un diagramme d'enchaînement des pages (DOCPAGES)

Ex :



- i) Avoir un répertoire spécifique pour contenir les photographies: PHOTOS
- j) L'ensemble des fichiers « .sql », « .css », « .html », « .php », « .doc » doit être mis dans un répertoire nommé « Projet\_nombinôme1\_nombinôme2 ».
- k) Les contrôles d'accès aux pages doivent être faits par utilisation de sessions.