

Mini-projet Web côté Serveur - Application VisageLivre

Préambule

Ce mini-projet a pour but de vous faire développer une application de réseau social, de type *Facebook*, qui permettra la construction d'un réseau social. Les fonctionnalités demandées dans ce sujet sont volontairement limitées et constituent un socle de base pour l'application que nous vous demandons de développer. Votre développement se basera sur l'utilisation de CodeIgniter et une base de données et des fichiers de ressources que nous vous fournissons.

Nous rappelons sur la figure 1 le diagramme de classes pour cette application. Vous pourrez l'amender, si nécessaire, afin de réaliser les fonctionnalités demandées.

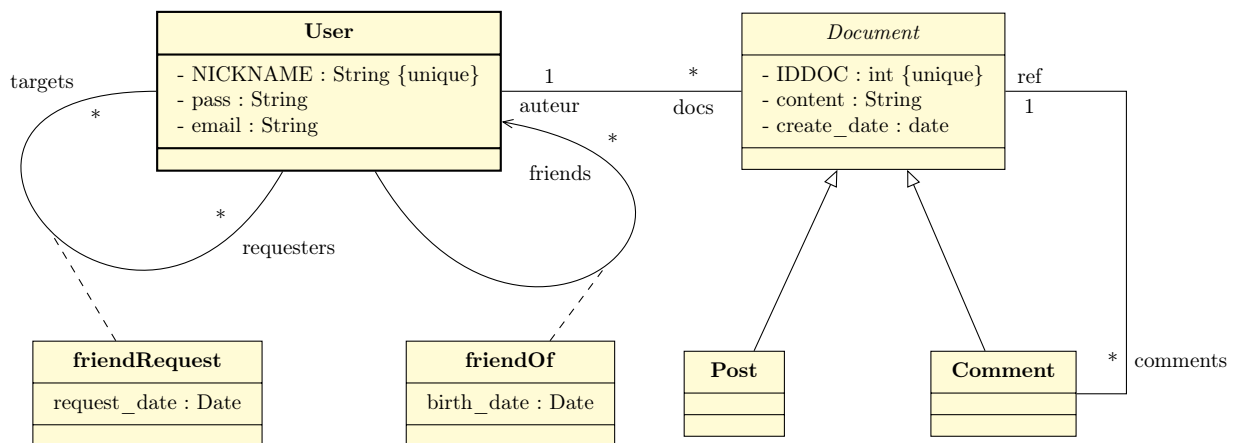


FIGURE 1 – Diagramme de Classes - Application VisageLivre

Définitions

- **L'utilisateur** : Un usager de l'application est une personne pouvant utiliser l'application. Il peut s'agir d'un *visiteur* ou d'un *utilisateur* (user).
- **Le visiteur** : Un visiteur est un usager anonyme de l'application. Un visiteur peut devenir utilisateur (voir le cas d'utilisation 1).
- **Un utilisateur** : Un utilisateur est un usager authentifié sur l'application. Il peut voir les informations de son **profil** : identifiant(*nickname*) et adresse électronique. Il peut modifier son adresse mail. Un utilisateur peut tisser des liens avec d'autres utilisateurs en se créant des amis (voir cas d'utilisation 3).
- **Les billets** : Un billet (*post*) est un message qui peut être créé par un utilisateur. C'est un des deux types de *documents* avec le *commentaire*. Ce billet est partagé (visible) par l'ensemble de ses amis sur l'application. Un billet est caractérisé par un identifiant, un contenu et une date de création.
- **Les commentaires** : Un commentaire est un message qui est rattaché à un *document* (soit un billet, soit un autre commentaire). Il n'y a pas de limite au nombre de commentaires par document, non-plus au nombre de niveaux de commentaires pour un document.

Cas d'utilisation de l'application VisageLivre

1. **Devenir utilisateur** : un visiteur peut devenir utilisateur. Il lui suffit pour cela de s'inscrire dans l'application en définissant obligatoirement un identifiant, un mot de passe et une adresse électronique. La

vérification de la validité de l'adresse électronique n'est pas demandée. Une fois inscrit un utilisateur peut voir les informations de son profil.

2. **Afficher les utilisateurs et les amis** : un utilisateur peut afficher la liste des utilisateurs. Chaque item de la liste est composé d'un *nickname*. Pour chaque item de la liste des utilisateurs, il peut demander à établir une relation d'amitié, s'il n'est pas ami avec cet utilisateur (cas d'utilisation 3), soit mettre fin à la relation d'amitié (cas d'utilisation 5). Il peut également afficher la liste de ses amis. L'affichage de la liste d'amis est du même type que celles des utilisateurs, mais on dispose uniquement de la possibilité de mettre fin à la relation d'amitié.
3. **Etablir une relation d'amitié** : cela nécessite le consentement mutuel des deux parties. Un utilisateur sollicite un autre utilisateur afin d'établir une relation d'amitié. Cette relation prend effet lorsque qu'un utilisateur sollicité accepte l'établissement de la relation. La demande d'amitié n'a alors plus lieu d'être et est supprimée.
4. **Lister les demande d'amitié** : une nouvelle page permet d'afficher la liste des demandes d'amitiés. Chaque item de la liste est suivie de liens ou de boutons permettant d'accepter ou de refuser une demande d'amitié. Pour simplifier le traitement, une demande d'amitié refusée est supprimée.
5. **Supprimer une relation d'amitié** : la suppression est prise à l'initiative de l'une des deux parties.
6. **Créer un billet** : tout utilisateur peut créer un billet.
7. **Voir ses billets** : dès sa création, un billet est visible par son créateur et tous ses amis. Un utilisateur pourra visualiser une liste de ses billets classés par ordre chronologique (du plus récent au plus ancien en descendant dans la liste). Chaque contenu de billet dans la liste des billets est résumé (Les 30 premiers caractères sont affichés suivis de "...").
8. **Voir les billets des amis** : dès sa création, un billet est visible par son créateur et tous ses amis. Un utilisateur pourra visualiser une liste des billets de ses amis sur une nouvelle page. Ces billets sont également classés par ordre chronologique (du plus récent au plus ancien en descendant dans la liste), mais peuvent être classés par amis (ordre alphabétique du *nickname*), puis par ordre chronologique. Chaque contenu de billet dans la liste des billets est résumé de la même façon que dans le cas d'usage 7. En plus, le résumé de chaque billet est précédé du *nickname* de son créateur.
9. **Voir le détail d'un billet** : Le contenu résumé dans la liste des billets est un lien vers une page affichant le billet complet ainsi que le fil des commentaires qui suit. L'affichage des commentaires est hiérarchisé comme dans la structure initiale dans la base de données (similaire à ce qu'on peut observer dans *Facebook*).
10. **Créer un commentaire** : tout utilisateur peut créer un commentaire pour un document (billet ou commentaire). Il suffit qu'il soit créateur du billet d'origine ou que l'un de ses amis soit créateur de ce billet. Un lien ou un bouton à cliquer permettant de créer un commentaire est disponible à côté d'un document sur la page affichant le détail d'un billet.
11. **Supprimer un billet** : le créateur d'un billet peut le supprimer à tout moment. Tous les commentaires liés à ce billet sont également supprimés.
12. **Supprimer un commentaire** : le créateur d'un commentaire peut le supprimer à tout moment. Tous les commentaires liés à ce commentaire sont également supprimés.
13. **Supprimer un utilisateur** : un utilisateur peut supprimer son compte à tout moment. Le bouton de suppression est disponible dans la page décrivant les informations de son profil. La suppression du compte utilisateur entraîne sa déconnexion automatique et la suppression de tous les documents liés à ce compte, avec les conséquences prévues pour la suppression des documents.

Environnement technique et Consignes

Les outils dont vous disposez

- Un compte Windows/Linux qui porte le nom de votre équipe (exemple : d21 pour une équipe du groupe D2). Ce compte spécifique vous a été attribué après la constitution des équipes. Vous devez activer ce compte en vous connectant une première fois sous Windows et en changeant le mot de passe initial qui vous a été donné par l'enseignant.
- Un dépôt Web qui porte le nom de votre équipe (exemple : d21). Ce dépôt est accessible sous Linux dans le *homedir* du compte `<nom_equipe>`, dans le sous-répertoire `~/WEB`. Le résultat de vos développements est visible à l'URL de base suivante :
`http://webetuinfo.iutlan.univ-rennes1.fr/<nom_equipe>`
Mais ce dépôt ne vous sera utile qu'à la fin du projet au moment "d'installer" votre application web.
- Une base de données commune à l'équipe sur le serveur PostgreSQL habituel : `servbdd.iutlan.etu.univ-rennes1.fr`. Cette base est nommée `pg_<nom_equipe>`. Elle est accessible à partir du rôle `<nom_equipe>`.
- Un dépôt Gitlab qui porte aussi le nom de votre équipe. Il est accessible à l'adresse suivante depuis l'intérieur et l'extérieur de l'IUT (ne pas vous fier à l'adresse donnée par l'interface graphique) :
`https://gitlab.iutlan.univ-rennes1.fr/web-m3104/<nom_equipe>`
Vous y accédez avec votre login et votre mot de passe habituels (Linux, Windows et PostgreSQL). Ainsi, vos contributions respectives seront enregistrées à vos noms.

Manipulations à effectuer impérativement

En début de projet, vous devrez :

- Activer votre compte commun Windows/Linux qui porte le nom de votre équipe
- Vérifier les accès à votre base de données, ainsi qu'à votre dépôt web commun (en y plaçant un fichier test par exemple) et, enfin, à votre dépôt Gitlab (visible dans l'interface graphique).
- **PAR L'UN DES MEMBRES DE L'EQUIPE SEULEMENT** : Télécharger le script intitulé `create_repo_prjweb.bash` sur l'ENT et l'exécuter depuis un répertoire de votre choix. Ce script s'appuie sur le client git sous Linux. Il a pour but de créer un répertoire **visagelivre** contenant les fichiers du framework CodeIgniter et de l'envoyer sur votre dépôt git.

```
> chmod u+x create_repo_prjweb.bash
> ./create_repo_prjweb.bash
....
Entrez votre identifiant d'equipe (en minuscules) : d21
...
```

Vous aurez probablement votre mot de passe d'utilisateur à saisir (celui de votre compte personnel).
- **Pour chacun des membres de l'équipe**, faire un git clone de votre dépôt git dans votre répertoire :

```
> cd ~/WEB # le votre pour le moment...
> git clone https://gitlab.iutlan.univ-rennes1.fr/web-m3104/d21.git
```

A ce stade, vous devriez observer un répertoire `d21/visagelivre` à la racine de votre dépôt Web.

Consignes

Vous respecterez à la lettre les consignes données dans ce document.

- La réalisation reposera sur un contrôleur qui sera chargé de gérer les événements. Le Modèle fera usage d'une base de données *PostgreSQL* dans laquelle seront enregistrées les informations concernant utilisateurs et la gestion des relations d'amitiés et des documents. Cette base de données repose sur le diagramme de classes qui est donné en figure 1. La conception de la partie Vue ne sera pas une question d'esthétique mais surtout d'ergonomie.
- Vous fournirez à chaque fin de séance (au minimum) une version de vos scripts PHP sur votre dépôt Gitlab **commun**. Faites attention aux conflits car vous travaillez désormais à plusieurs sur le même dépôt.

Le principe est de toujours commencer une séance (ou de précéder un `git push`) par un `git pull` pour obtenir les dernières versions en local. **Tous les membres de l'équipe doivent contribuer au dépôt de fichier sur Gitlab. La contribution de chacun entrera en ligne de compte dans la notation.**

- La base de données vous est fournie sous forme d'un script SQL `cr_base_visagelivre.sql` (fourni sur l'ENT) qui crée un schéma `visagelivre` dans votre base et les tables nécessaires à votre projet. Vous pourrez amender cette base de données ou compléter les jeux de données d'essai. **Votre application s'appuiera obligatoirement sur le schema *visagelivre* dans la base de données commune portant de le nom de votre équipe.**
- Votre travail final se trouvera sur le dépôt Web commun et **sera impérativement accessible à l'adresse :**
`http://webetuinfo.iutlan.univ-rennes1.fr/<nom_equipe>/visagelivre`