

BarBeasts



Sommaire

Contexte	3
Objectifs	3
Fonctionnalités de l'application	3
Modèle conceptuel de données.....	4
Extrait de code commenté	5
Maquette.....	6
Technologies utilisées	7
Répartition des rôles	7
Etude ergonomique.....	8
Avantages et limites	9
Conclusion	9

Contexte

Actuellement, tous les sites concernant la musculation ne sont pas tous facile d'utilisation et certains ne permettent pas de faire participer la communauté pour mettre à jour leur site. De cela nous avons déterminé qu'il existait un besoin de créer un site sur ce thème et qui comblerai le manque des sites existants.

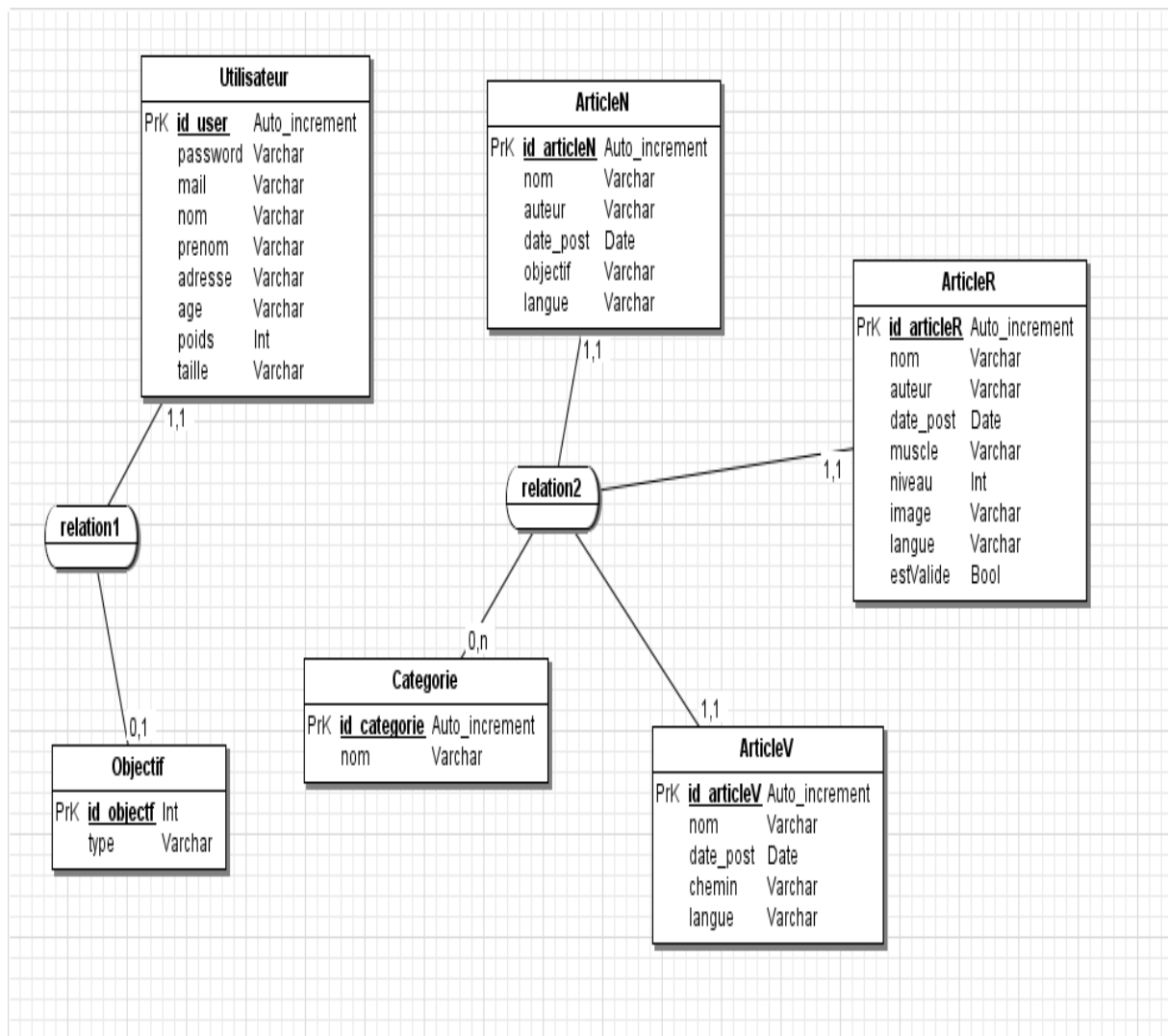
Objectifs

Nous voulons fournir une plateforme sur une passion commune qu'est la musculation « workout », pour cela nous permettons à nos utilisateurs d'être passifs ou actifs, s'ils veulent juste des informations, ils peuvent uniquement consulter le site à titre informatif alors que s'ils veulent contribuer, ils pourront poster des articles « homemade » (sous le contrôle de l'administrateur).

Fonctionnalités de l'application

Nous avons divisé l'application en plusieurs parties, une qui concernera ce qu'on appelle les routines, elles consistent en une suite d'exercice détaillées, une autre qui concernera toute la partie alimentation (calcul d'indice de masse corporelle et consultation de programmes alimentaires) et une dernière parties qui concernera les vidéos, avec possibilité de les consultés. A tout ceci s'ajoute un menu qui donne la possibilité aux membres connecté de poster une vidéo ou de créer/modifier un article et enfin des suggestions d'améliorations.

Enfin un petit menu à droite permettra de nous diriger vers les réseaux sociaux de BarBeasts afin de pouvoir partager au maximum notre contenu.

Modèle conceptuel de données

Extrait de code commenté

```

// Création d'un exercice
public function ajoutArticleAction(Request $request)
{
    // Création du formulaire d'ajout.
    $article = new ArticleR();
    $article->setDateArticleR(new \Datetime());
    $article->setEstValide(0);

    // Création d'un formulaire
    $form = $this->get('form.factory')->createBuilder('form',$article)
        ->add('nom','text')
        ->add('auteur','text')
        ->add('muscle','choice',array('choices' => array('Bras'=> 'Bras','Epaules'=>'Epaules', 'Dos' => 'Dos', 'Pectoraux'
        ->add('niveau','choice',array('choices' => array(0 => 'Facile',1 => 'Moyen', 2 => 'Difficile', 3 => 'Expert'))
        ->add('description','textarea')
        ->add('image','file')
        ->add('langue','choice',array('choices' => array('FR' => 'FR', 'EN' => 'EN', 'RU' => 'RU'))
        ->add('creer','submit')
        ->getForm();

    $form->handleRequest($request); // On attrape la requête POST

    // Si Le formulaire est valide

    if($form->isValid())
    {
        $article->setUrl($form['image']->getData()); // Je défini Le path de l'image pour l'upload sur le serveur
        $manager = $this->getDoctrine()->getManager(); // Je récupère Le gestionnaire d'entité
        $manager->persist($article); // Persiste Les données en bases
        $manager->flush(); // Exécution

        $request->getSession()->getFlashBag()->add('notice','Annonce bien enregistrée.');// J'enregistre une notice pour affiché

        // On redirige vers l'accueil quand l'article est crée
        return $this->redirect($this->generateUrl('bar_beasts_bar_beasts_accueil'));
    }

    // Sinon Le formulaire est pas valide donc j'affiche Le formulaire et je renvoi la vue associé à l'ajout
    return $this->render("BarBeastsBarBeastsBundle:Exos:ajouter.html.twig",array('form' => $form->createView()));
}

```

Maquette



Technologies utilisées



Symfony 2 : Nous avons choisi Symfony2 car c'est un framework MVC en php qui permet de rendre ses applications web facilement maintenable et évolutive, il assure une sécurité forte et un chargement de page très rapide en production.



Notepad++ : C'est un éditeur de texte qui facilite le développement et qu'on utilise depuis plusieurs années. C'est un classique en développement



Paletton : C'est un site web qui fournit une palette de couleur et tous les codes hexadécimaux correspondant, nous l'avons utilisé pour tous le côté design/couleur du site.

Répartition des rôles

Gerard Christophe -> Chef de projet/Développeur.

Safronov Kirill -> Web Designer.

Kielbasa Alexis -> Développeur.

Etude ergonomique

La navigation :

Nous avons pensé la navigation pour que l'utilisateur n'ait pas plus de 3 clics à faire pour qu'il accède à n'importe quelle page du site.

Nous avons aussi pensé à la façon de naviguer de façon cohérente, c'est-à-dire que le menu qui donne la possibilité d'ajouter/modifier un article n'apparaît que si l'on est identifié sur le site. Chaque partie comme « Exos » ou « Nutritions » dispose de contenu cohérent avec le type de partie. Par exemple dans la partie « Nutritions » je ne trouverais pas d'exercice.

L'interactivité :

Nous avons utilisé du JavaScript pour rendre le site interactif, quelques animations sont disponibles et font un rendu visuel agréable, nous vous montrerons cela plus en détails lors de la présentation. L'utilisateur peut cliquer sur chaque partie afin de naviguer et les animations se déclenchent la plupart du temps sur des clics/hover.

Le contenu :

Pour le contenu nous avons choisi de le générer de façon fixe au milieu de l'écran, c'est-à-dire que toute la structure autour ne change jamais seulement le contenu au milieu change (et les animations du menu).

Aspects techniques et fonctionnels :

On laisse la possibilité aux utilisateurs de créer/modifier ou poster une vidéo n'importe où ils se trouvent sur le site. Nous avons pensé aussi à rendre simple la modification ou l'ajout d'un article, pour ajouter ils ont juste à remplir un formulaire avec une image jointe et l'application met en forme toute seule, pour la modification on liste tous les articles de l'auteur qui veut modifier un article et s'il clique dessus il pourra modifier son contenu.

Avantages et limites

Avantages

Symfony2 permet de rendre l'application facilement maintenable et évolutive. Le développement est modulaire.

Le framework permet également d'optimiser les chargements de pages afin de rendre la navigation agréable.

L'application web est « Responsive design », le style s'adapte à la taille de l'écran.

L'utilisateur peut être actif pour ajouter du contenu.

Limites

Mettre en place de l'Ajex n'est pas évident avec Symfony2 car le framework n'a pas été pensé pour ça, comme je l'ai dit dans les avantages il se focalise sur l'optimisation des chargements des pages.

Le « Responsive design » est une limite dans le sens où le contenu s'adapte selon la résolution mais pas selon la taille de l'écran, ce qui peut provoquer des problèmes d'affichages sur les terminaux mobiles ou tablettes.

Conclusion

Même si le projet n'est pas encore abouti, c'est un plaisir de développer une application web qui sera certainement utile à une grande communauté. Nous avons lors de ce projet appris à mieux travailler en équipe et s'organiser. Nous envisageons par la suite d'ajouter plusieurs fonctionnalités à l'application comme la possibilité de proposer des programmes de musculation selon la taille le poids et le but de l'utilisateur.