
PROJET SYMFONY

ComicsTeams

Sommaire

Nos modèles de conception.....	3
Modèle de conception des données (MCD)	3
Diagramme de classe.....	3
Notre schéma architectural	3
Controller	3
Repository	3
Entity.....	3
DataFixtures	4
Templates	4
Form.....	4
Documentation	4
Les rôles et tâches de chacun	4
Tableau récapitulatif	4
Détail page par page	5
Nos fonctionnalités.....	6
Identification Utilisateur / Administrateur	6
Nos choix	6
Nos choix « techniques »	6
Nos choix « pratiques »	6
Notre choix graphique	7
Les difficultés rencontrées.....	7
De XAMPP à WAMP	7
L'ajout d'article	7
Manque d'expérience / Trop de dépendances	8
Manque de temps	8
Conclusion.....	8
Source	8
Annexe	9

Nos modèles de conception

La définition de la structure du projet ainsi que la modélisation orientée objet été nécessaire sur notre projet afin de mettre en évidence les points clés de notre application et faciliter le développement. Vous trouverez tous le diagramme de classe et notre modèle conceptuel de données dans le dossier « documentation » de notre projet.

Modèle de conception des données (MCD)

Après l'initialisation du projet et la mise en place de la base de données, nous avons réalisé un modèle conceptuel de données, l'élément le plus connu de MERISE et certainement le plus utile, qui permet d'exprimer graphiquement les règles de gestion qui correspondent aux contraintes d'intégrité de données, afin de spécifier de façon formelle les données qui seront utilisées, une représentation abstraite des données (sous forme d'entités et associations entre entités) de notre application. Ce schéma que nous avons réalisé a été évolutif tout au long du projet (au fur et à mesure des implémentations des fonctionnalités, nous nous sommes aperçus qu'il fallait changer certaines relations et ajouter certains champs à quelques entités).

Diagramme de classe

Nous avons réalisé un diagramme de classe, l'un des types les plus populaires en langage UML, afin de spécifier la structure interne de notre application et la vue statique du modèle, définir la modélisation des concepts du domaine et les différentes classes du système sans oublier leurs relations. Le but principal de ce dernier était de mieux comprendre l'aperçu général du schéma de notre application. Vous trouverez donc notre première version du diagramme de classe, qui correspond à ce que nous avons voulu réaliser pour ce projet (qui, du coup, n'est pas aussi complet que notre MCD).

Notre schéma architectural

Comme nous l'avons appris pendant ce cours, nous avons décidé de découper notre code dans différents fichiers. Voici les dossiers (« principaux ») que nous avons décidé d'utiliser.

Controller

Ce dossier contient tous les contrôleurs du fameux MVC proposé par le Framework. On s'occupe principalement ici de contrôler les différentes actions que nous avons mis en place, afin de permettre la redirection souhaitée.

Repository

Nous retrouvons ici tous les fichiers de repository, qui permettent d'organiser les liens avec la base de données (contient principalement les requêtes telles que les récupérations d'entités ou encore même la session d'utilisateur).

Entity

Comme son nom l'indique, contient toutes les entités présentes sur notre blog.

DataFixtures

Ce dossier contient toutes les données que nous avons voulu générer dans la base de données. Après plusieurs concertations et plusieurs changements, nous avons décidé de générer toutes les données depuis un seul fichier.

Templates

Ce dossier représente les « views » du model MVC. Nous avons souvent utilisé la génération automatique du « CRUD », vous pouvez donc remarquer que les fichiers html.twig sont rangés par dossier d'entité.

Form

Afin de mener à bien nos formulaires, ce dossier est indispensable pour pouvoir les gérer.

Documentation

Vous trouverez tous nos documents complémentaires dans ce dossier.

Les rôles et tâches de chacun

Voici brièvement les rôles et tâches de chacun (les détails se trouvent juste après).

Tableau récapitulatif

Membre de l'équipe	Rôle	Tâches
BAILLEUL Valentin	Développement Back, également Scrum Master	<ul style="list-style-type: none"> - Réalisation MCD / Diagramme de classe - Mise en place des Fixtures de toutes les entités - CRUD Article, Commentaire - Recherche - Réalisation du compte rendu
CAILLAT François	Travaillant sur le Framework Symfony en entreprise, François a été le leader technique en développement de notre équipe	<ul style="list-style-type: none"> - Réalisation MCD / Diagramme de classe - CRUD User, Article, Commentaire - Authentification - Pagination - Aide constante sur les tâches des autres membres de l'équipe
ENAIM Safae	Développement Front principalement	<ul style="list-style-type: none"> - Réalisation MCD / Diagramme de classe - Onglet Contact - CRUD Article
NEGRO Fabio	Développement Back et Front, Designer de l'équipe	<ul style="list-style-type: none"> - Réalisation MCD / Diagramme de classe - CRUD Article, Commentaire, User - Mise en place du design du sites - Mise en place des Fixtures de toutes les entités

Détail page par page

Voici sous forme de tableau toutes les pages présentes dans le cahier des charges ainsi que les personnes qui ont contribué à la réalisation de ces dernières. Même si tous les membres de l'équipes ont travaillé en collaboration constante, vous trouverez la contribution de chacun sur les pages du site.

Afin de ne pas surcharger notre tableau inutilement, une section « Les difficulté rencontrées » est présente à la fin du document (cette dernière regroupe toutes les difficultés dans les moindres détails que nous avons pu rencontrer durant la réalisation de notre projet).

Page	Réaliser par	Quelle partie
Page Accueil	Valentin	Affichage articles, Mise en place de la recherche
	Fabio	Affichage articles, design
	François	Ajout pagination
Page Article	Valentin	Affichage article, formulaire commentaires
	Fabio	Affichage article, formulaire commentaires, design
	François	Formulaire commentaire
Page à propos	Valentin	La totalité de la page
Page Contact	Safae	La totalité de la page
Page Créer un compte	François	La totalité de la page
Page Se connecter	François	La totalité de la page
Page Compte utilisateur (avec tous les onglets)	François	La totalité de la page et des onglets
Page Compte administrateur - Accueil	Valentin	Affichage articles
	Fabio	Affichage articles
	François	Ajout des onglets pour administrateur
Page Compte administrateur - Articles	Valentin	Affichage article
	Fabio	Affichage article
	François	Fonction de tris
Page Compte administrateur - Gérer les commentaires	François	L'affichage des commentaires
	Valentin	Les redirections
	Fabio	La gestion avec la base de données
Page Ajouter un article	Valentin	Réglages routes
	Fabio	Design
	Safae	Mise en place du formulaire
	François	Finalisation
Page Modifier un article	Non réalisée	

Nos fonctionnalités

Vous allez avoir la possibilité de tester notre site, cependant nous vous proposons un bilan des fonctionnalités que vous allez pouvoir tester.

Rôle	Fonctionnalité
Utilisateur	Créer un compte / Se connecter / Se déconnecter
	Consulter / Rechercher / cliquer sur un article
	Ajouter un commentaire à un article
	Changer ses informations personnelles
	Consulter ses articles aimés
Administrateur	Mêmes fonctionnalités que l'utilisateur
	Ajouter un article
	Confirmer / Rejeter un commentaire
Commun à tous	Consulter la page À propos
	Contacteur les « gérants » du site

Identification Utilisateur / Administrateur

Afin de découvrir nos fonctionnalités par vous-même, voici les identifiants que nous vous proposons.

Type	Identifiant	Mot de passe
Administrateur	udmin0@admin.com	admin
Utilisateur	user3@user.com	user

Nos choix

Nos choix « techniques »

Choix	Utilisation
Framework Symfony (Twig, doctrine)	Framework PHP basé sur le pattern MVC.
Langage PHP v7.3	Langage de script.
Composer	Gestionnaire de dépendances.
GitHub (SourceTree)	Suivie des versions d'un code source. Pour certains membres de notre groupe, nous avons utilisé l'interface SourceTree.
XAMPP, WAMP	Environnement des serveurs locaux (Tout le monde est passé sur WAMP).

Nos choix « pratiques »

Choix	Utilisation
PHP Storm / Visual Studio Code	IDE de chacun (3 personnes sur VS, 1 sur PHP Storm).

Live Share	Cette fonctionnalité présente sur le Live Share nous a permis de coder tous ensemble sur une même version du projet.
Teams	Nous a permis de nous réunir et partager les écrans de chacun.
Méthodes agiles	Malgré la courte durée du projet, nous avons tout de même respecté le modèle agile.
Trello (ajout de tags)	Nous avons pu avoir une vue d'ensemble des tâches restantes à faire grâce à ce site. (Rouge : Pas encore faite, Orange : En cours de réalisation, Vert : Réalisée).

Notre choix graphique

Concernant le côté graphique de l'application, nous avons décidé d'utiliser principalement le Template « SuperHero » proposé par Bootstrap afin de donner un bon visuel à l'application.

Comme vous pouvez l'apercevoir en annexe, le thème principal de notre site contient deux couleurs majeures (bleu et orange).

Au niveau des différents noms donnés dans la barre de navigation, nous avons entièrement respecté les onglets demandés dans le cahier des charges afin de n'oublier aucune fonctionnalité.

Les difficultés rencontrées

Nous avons rencontré certaines difficultés durant notre projet, cependant elles ont tout de même été limitantes de notre avancée.

De XAMPP à WAMP

L'installation du projet fut réalisée facilement, cependant l'installation des serveurs locaux n'a pas été simple. En effet, nous avons décidé de réaliser le projet à l'aide des serveurs WAMP. La migration de XAMPP à WAMP a été plus longue et difficile que prévu. Nous avons donc choisi une méthode de contournement face à cette difficulté qui devenait de plus en plus importante : l'utilisation de Live Share. Le fait d'avoir pu partager le code pour que tout le monde puisse coder a permis une véritable avancée dans le projet. Une fois la journée terminée, nous nous sommes posés quelques minutes afin de pouvoir installer correctement XAMPP dans le projet des personnes de notre groupe concernées.

L'ajout d'article

Lors de l'installation du fameux « CRUD » de l'entité article, une personne de notre groupe a décidé d'implémenter la fonctionnalité d'ajout d'article pour un administrateur. Dans un même temps, une autre personne a également travaillé sur cette même entité (pour l'affichage). Une fois le travail effectué par les développeurs réalisé, la mise en commun a dû se faire via le logiciel GitHub. C'est ainsi qu'est arrivé le plus gros problème durant notre temps

de développement durant la durée du projet : le formulaire de création d'article ne fonctionnait plus. Après plusieurs heures de recherches, nous nous sommes aperçus que certaines routes été « faussées » lors d'un merge effectué par un membre de l'équipe. Ce problème était vraiment majeur, cependant pendant que certains cherchaient la solution, d'autres ont pu continuer à faire avancer le projet.

Manque d'expérience / Trop de dépendances

Avec le manque d'expérience sur Symfony de la part de notre équipe, nous étions assez dépendants des disponibilités de notre leader technique. Malgré les différentes recherches effectuées sur le web, certaines fonctionnalités de notre site ont nécessité l'aide constante de François (par exemple l'ajout de commentaire, qui aurait pu se faire assez rapidement si nous avions eu encore plus de connaissances concernant les formulaires).

Manque de temps

Malgré tous nos efforts depuis jeudi dernier, nous n'avons pas réussi à fournir la totalité des fonctionnalités demandées dans le cahier des charges. Hormis les fonctionnalités facultatives demandées, la « responsivité » de notre site n'est présente qu'à moitié.

Conclusion

Malgré les différentes difficultés rencontrées, ce projet de groupe nous a permis d'acquérir de nouvelles connaissances concernant le Framework Symfony. Nous avons pu appliquer par nous même les principes vus en cours. Le fait d'avoir reçu énormément de travail à réaliser en si peu de temps nous a également appris à travailler sous une certaine pression, ce qui nous a permis de repousser nos limites afin de livrer un produit qui, de notre point de vue, contient de la quantité mais également de la qualité.

Source

<https://symfony.com/doc/current/index.html>

<https://stackoverflow.com/>

<https://fr.wikipedia.org/>

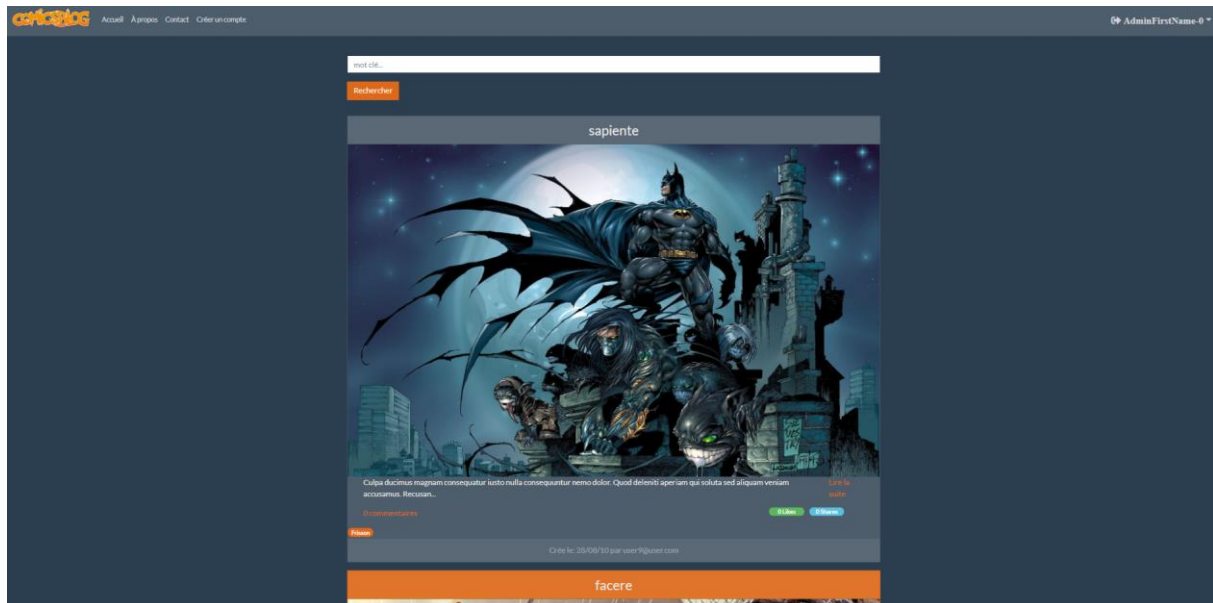
<https://bootswatch.com/>

<https://openclassrooms.com/fr/>

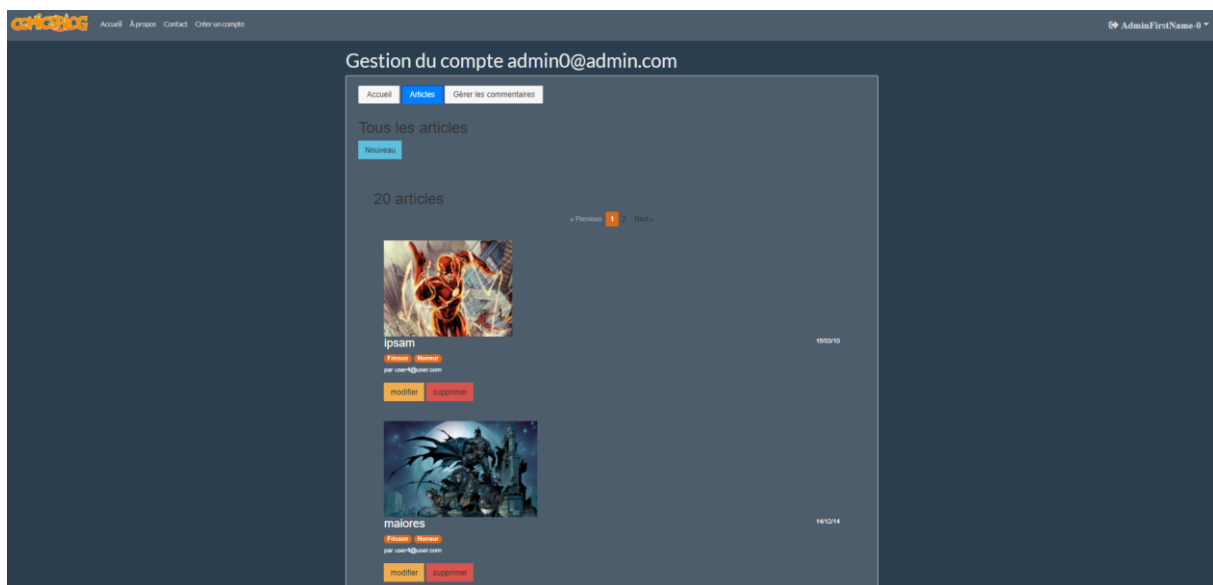
<https://github.com/>

Annexe

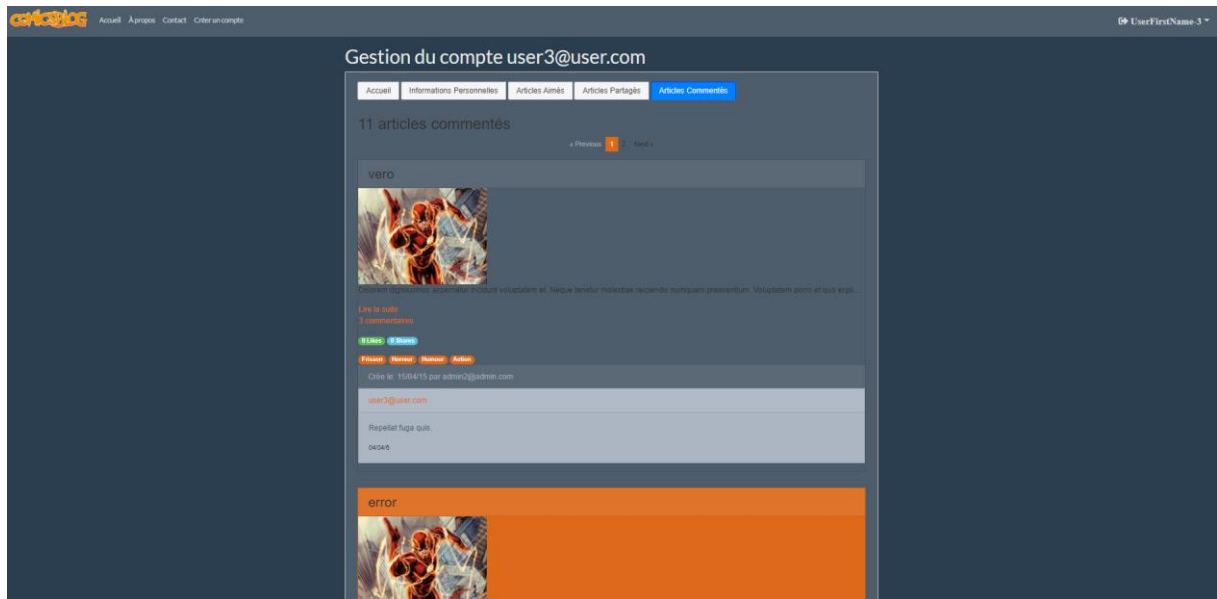
Voici également quelques screens de notre site internet



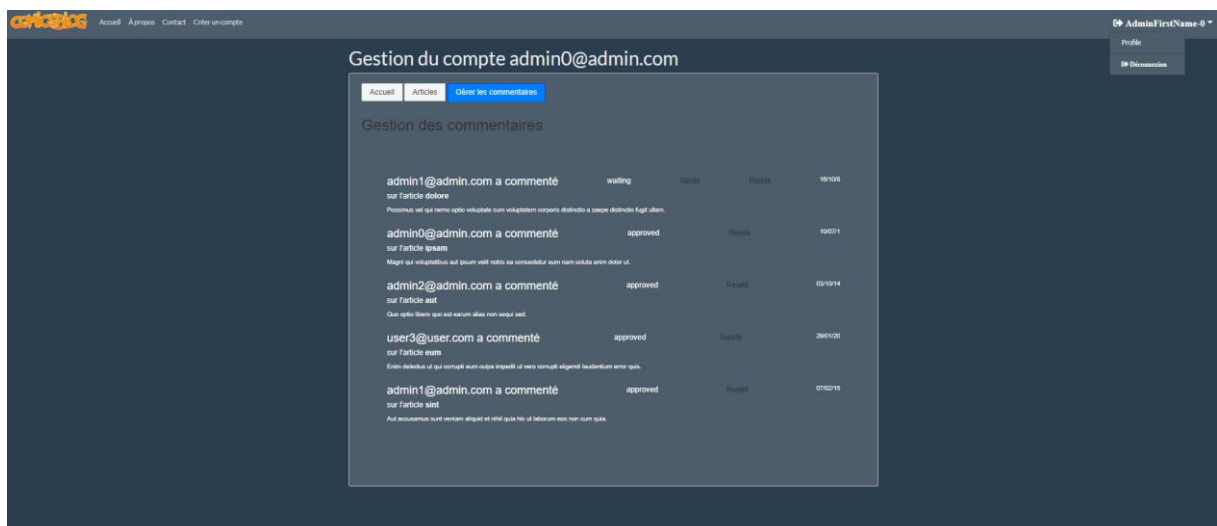
Page d'accueil



Page de gestion d'article (Administrateur)



Page des articles commentés (User)



Page Gestion des commentaires (Administrateur)