2021

GERARD Gaëtan

LOUREIRO-RIAL Alice

PARCOLLET Virgil

01/01/2021

C\_easy blog – Projet

****

Table des matières

[Présentation du projet : 1](#_Toc82173641)

[Equipe : 2](#_Toc82173642)

[Analyses & modélisation : 3](#_Toc82173643)

[**Utilisation de Trello :** 3](#_Toc82173644)

[**Modèle MVC :** 3](#_Toc82173645)

[**Composition de l’architecture MVC :** 3](#_Toc82173646)

[ Modèle 3](#_Toc82173647)

[ Vue 3](#_Toc82173648)

[ Contrôleur 3](#_Toc82173649)

[Diagramme d’activité : 4](#_Toc82173650)

[Diagramme de cas d'utilisations : 5](#_Toc82173651)

[Diagramme de classe : 6](#_Toc82173652)

[Modèle conceptuel des données : 6](#_Toc82173653)

[User 7](#_Toc82173654)

[Articles 7](#_Toc82173655)

[Category 7](#_Toc82173656)

[Comments 7](#_Toc82173657)

[Article\_category 7](#_Toc82173658)

[Réalisation : 8](#_Toc82173659)

[Visionner les articles : 8](#_Toc82173660)

[Publier un article : 8](#_Toc82173661)

[Publier un article : 8](#_Toc82173662)

[Séparation du Header et du footer des page html : 8](#_Toc82173663)

[Connexion a la base de données de façon généraliser : 8](#_Toc82173664)

[Conclusion : 9](#_Toc82173665)

# Présentation du projet :

“C easy blog” est un projet de blog axé sur les jeux vidéo, il est basé sur un Model MVC utilisant les technologie HTML/CSS/JS avec le Framework Bootstrap pour le front-end et PHP pour le back-end

Un Trello a été mis en place afin de lister l’ensemble des idées et fonctionnalités composant le projet et d’avoir un suivie de leur avancement.

Un Git a aussi été mise en place afin de facilité le travail en groupe en prenant un minimum de risque de perte de données.

# Equipe :

L’équipe de dev est composée : 

D’Alice en tant que développeur Back-end qui s’occupera de créer les fonctions et classe en PHP 

De Virgil en tant que développeur front-end qui s’occupera de créer des Views en html/css/js 

Et de Gaëtan qui fera office de chef d’équipe, qui s’occupera de la gestion de l’équipe au cours du projet.

# Analyses & modélisation :

## Utilisation de Trello :

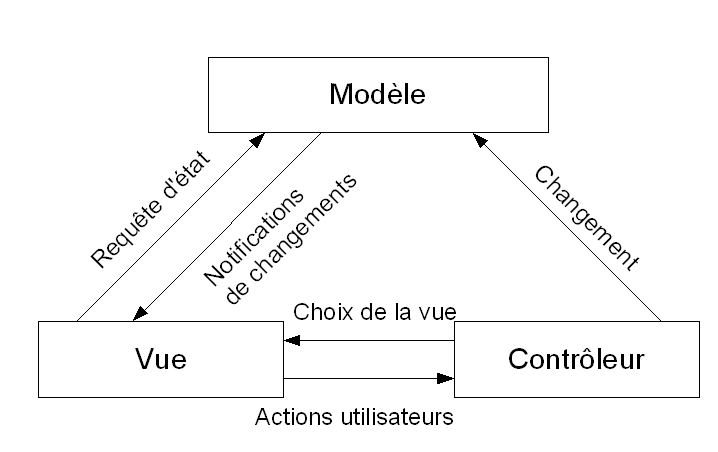
Nous avons fait le choix d’utiliser l’outil [Trello](https://trello.com/b/QO9vDFkm/c-easy-blog) afin de mieux nous organiser dans la planification des taches, d’avoir un suivie sur ce que tout le mondes fait et de son avancé, Trello nous permet aussi de garder une trace de nos idées.

## Modèle MVC :

Nous avons pris la décision d’appliquer l’architecture MVC (Modèle Vue Contrôleur) a notre projet afin d’avoir une meilleure organisation de notre code.

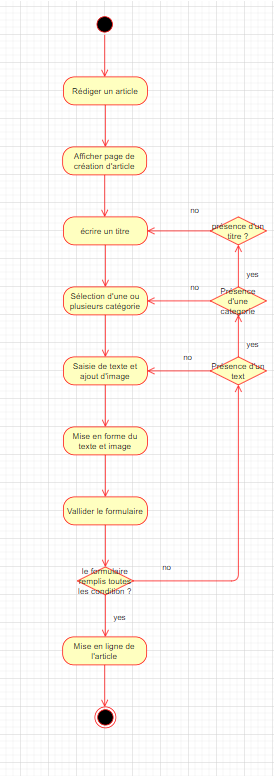
## Composition de l’architecture MVC :

* Modèle **:** un noyau de l’application qui gère les données, permet de récupérer les informations dans la base de données, de les organiser pour qu’elles puissent ensuite être traitées par le contrôleur.
* Vue **:** composant graphique de l’interface qui permet de présenter les données du modèle à l’utilisateur.
* Contrôleur **:** composant responsable des prises de décision, gère la logique du code qui prend des décisions, il est l’intermédiaire entre le modèle et la vue.



# Diagramme d’activité :

Diagramme d’activité lors de l’ajout d’un nouvel article.



L’utilisateur une fois connecter appuiera sur le bouton « crée un article » qui le renverra sur une page avec un formulaire.

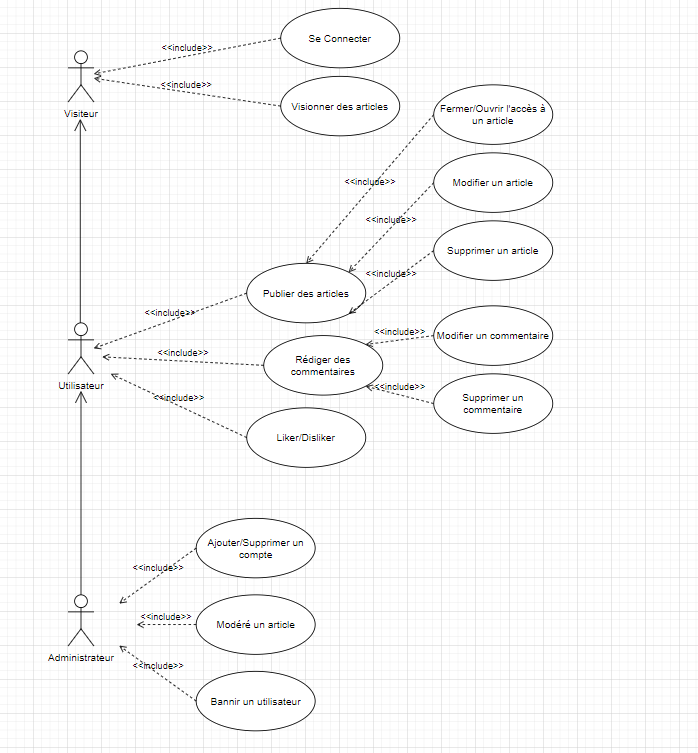
L’utilisateur devra saisir un titre, une catégorie et saisir un texte pour que le programme accepte l’envoie de l’article.

Si le formulaire ne remplie pas toutes les conditions, il affichera un message d’erreur dans les zones de saisie ne remplissant pas les bonnes conditions.

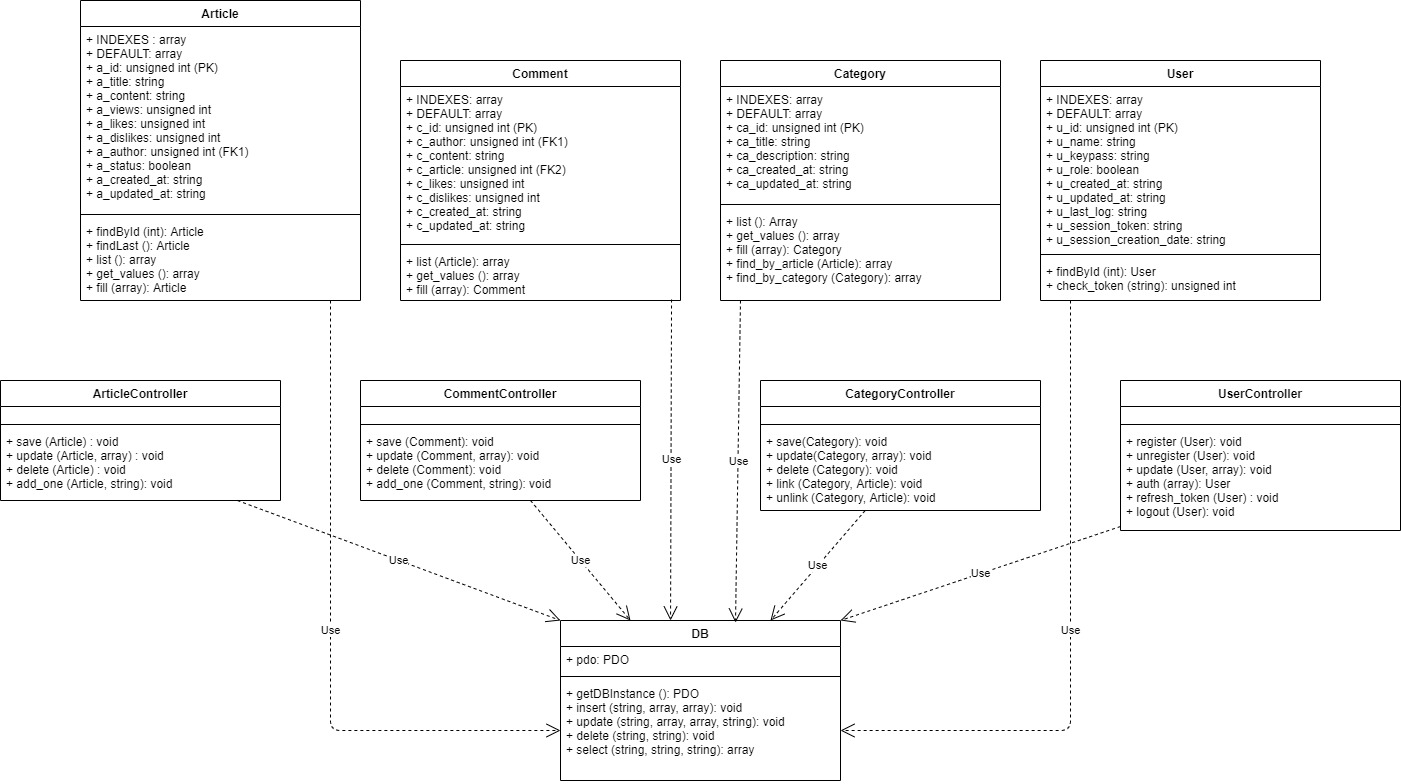
# Diagramme de cas d'utilisations :

Ce diagramme représente l’ensemble des fonctionnalités du produit final en fonction de chaque utilisateur.

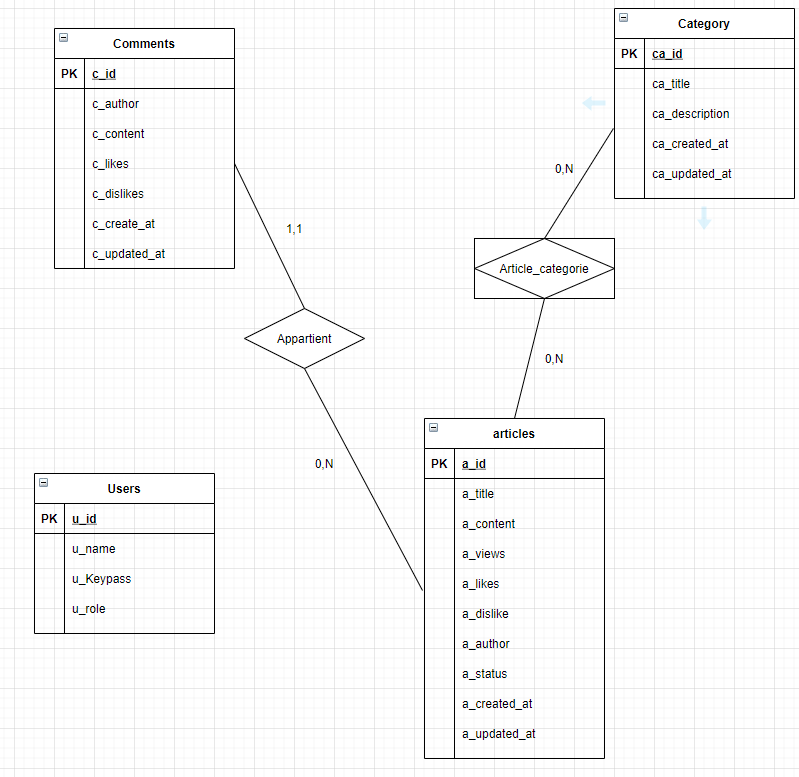
* Les visiteur, utilisateur et administrateur on la possibilité de se connecter et de visionner les articles
* Les utilisateur et administrateurs on la possibilité de publier des articles, écrire des commentaires et de réagir avec des likes au commentaire et articles.
* Les administrateurs on le doit à une gestion de compte utilisateur en pouvant supprimer / crée un compte ou le bannir, ils ont le droit à une gestion d’un article (suppression, modification)



# Diagramme de classe :



# Modèle conceptuel des données :



User**:**

La table user s’occupe de stocker le nom, le mot de passe et le rôle de chaque personne qui s’inscrit sur le site et les identifie par un numéro.

Articles**:**

La table articles s’occupe de stocker le titre, la description, le nombre de vue, de like et dislike, le nom de l’auteur, le statut (ouvert ou clos), la date de création et de dernière modification d’un article en les identifiant par un numéro.

Category**:**

La table category stock le libellé, la description, la date de création et de dernière modification d’une catégorie en les identifiant par un numéro.

Comments**:**

La table Comments stock l’auteur, le message, le nombre de like et dislike, la date de création et de dernière modification d’un commentaire est les identifie par un numéro. Il récupère est stock aussi l’article sur lequel le commentaire est posté.

Article\_category**:**

Il s’agit d’une table qui associe un ou plusieurs articles a une ou plusieurs catégories.

# Réalisation :

## Visionner les articles :

Tout Personne qu’elle soit connectée ou non, pourra avoir accès à la liste des articles et aura la possibilité de les consulter il ne pourra cependant pas écrire de commentaire ou mettre un like sur l’article.

## Publier un article :

Lorsqu’un utilisateur voudra écrire un nouvel article, il devra remplir un formulaire qui s’il n’est pas complété correctement ne sera pas envoyé.

Il aura la possibilité de modifier ou supprimer son article après publication, il pourra aussi le clore ce qui aura pour effet d’en bloquer les commentaires et d’archiver l’article.

## Publier un article :

Un utilisateur connecté a la possibilité de publier un commentaire en dessous d’un article afin d’échanger sont ressentie. Ensuite, s’il le désire il a également la possibilité de le modifier ou le supprimer.

## Séparation du Header et du footer des page html :

Nous avons fait le choix de séparer les header et footer de nos page web dans des fichiers séparés qui seront appelés par nos pages web, afin d’accélérer le développement des prochaines pages web et de grandement faciliter la maintenance/modification de ces parties sans avoir à les changer sur toutes les pages web.

## Connexion a la base de données de façon généraliser :

Alice a créé une classe DB contenant un singleton qui va créer une instance PDO qui sera appelée par chaque fonction CRUD afin de permettre un précieux gain de temps en développement.

# Conclusion :

## Bilan général :

**Bien que le projet ne fût pas terminé dans son entièreté suite à un projet peut-être trop ambitieux pour le temps imparti et quelques imprévus, le travail de groupe était organisé et efficace, chaque personne avait son rôle et la communication dans l’équipe était présente, ce qui a pu minimiser certain imprévu**

## Bilan personnel :

### Alice : Rien de particulier à dire, la création d’une classe générique de gestion de base de donnée était une partie intéressante. Le reste était assez simple du fait du peu de temps que nous avions. J’ai pu apprendre ce qu’était un Singleton.

### Virgil : Rien de particulier, le projet s’est très bien passé, l’équipe était bien coordonnées. Il me reste toutefois de nombreux progrès à faire mais sur le code mais Alice et Gaëtan m’ont bien aidé lorsque je bloquais. Ce projet, bien que court, m’a permis d’acquérir quelques connaissances en php notamment.

### 