

Projet 5

Blog Professionnel

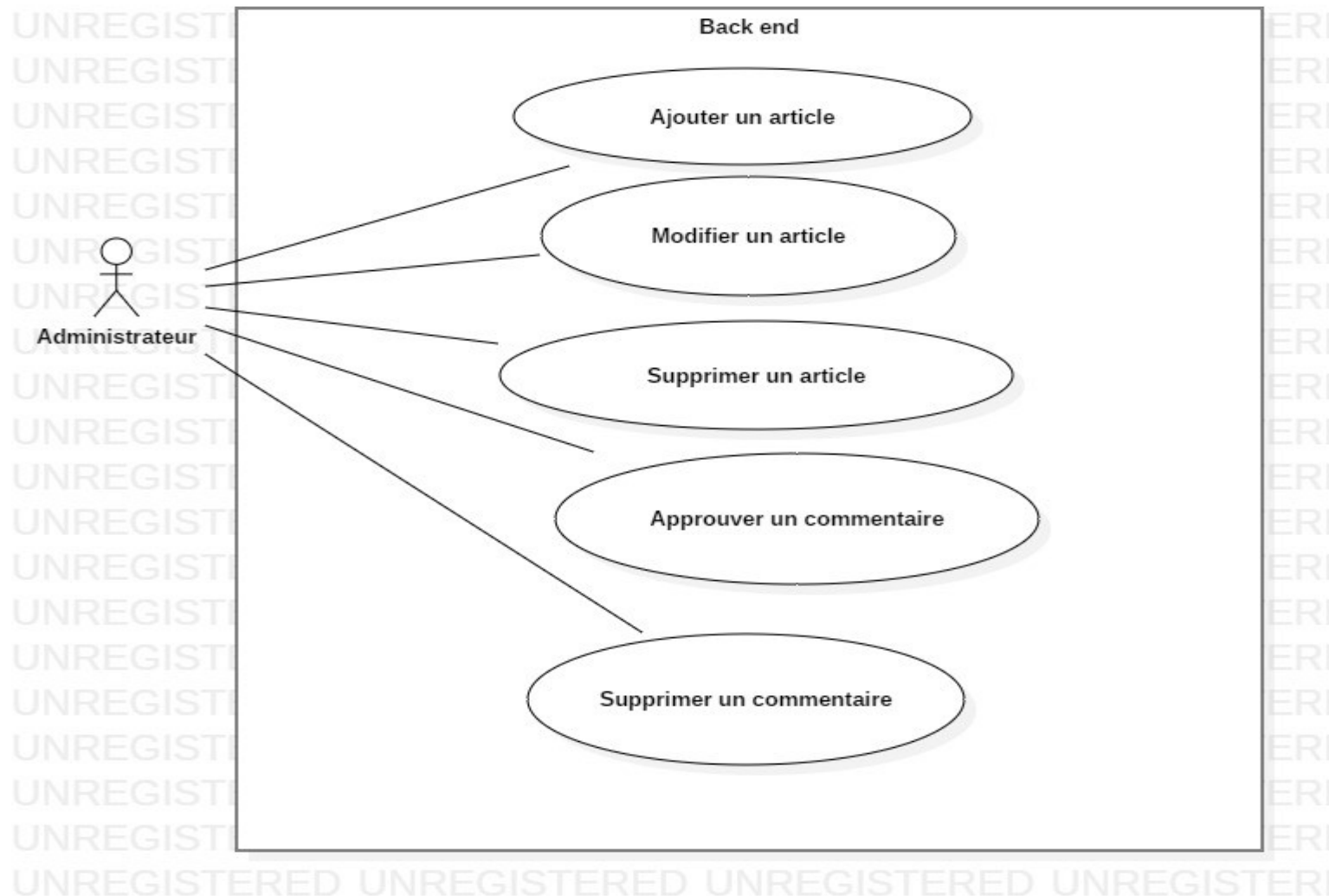
A) Contexte du projet

Création d'un blog professionnel composée de :

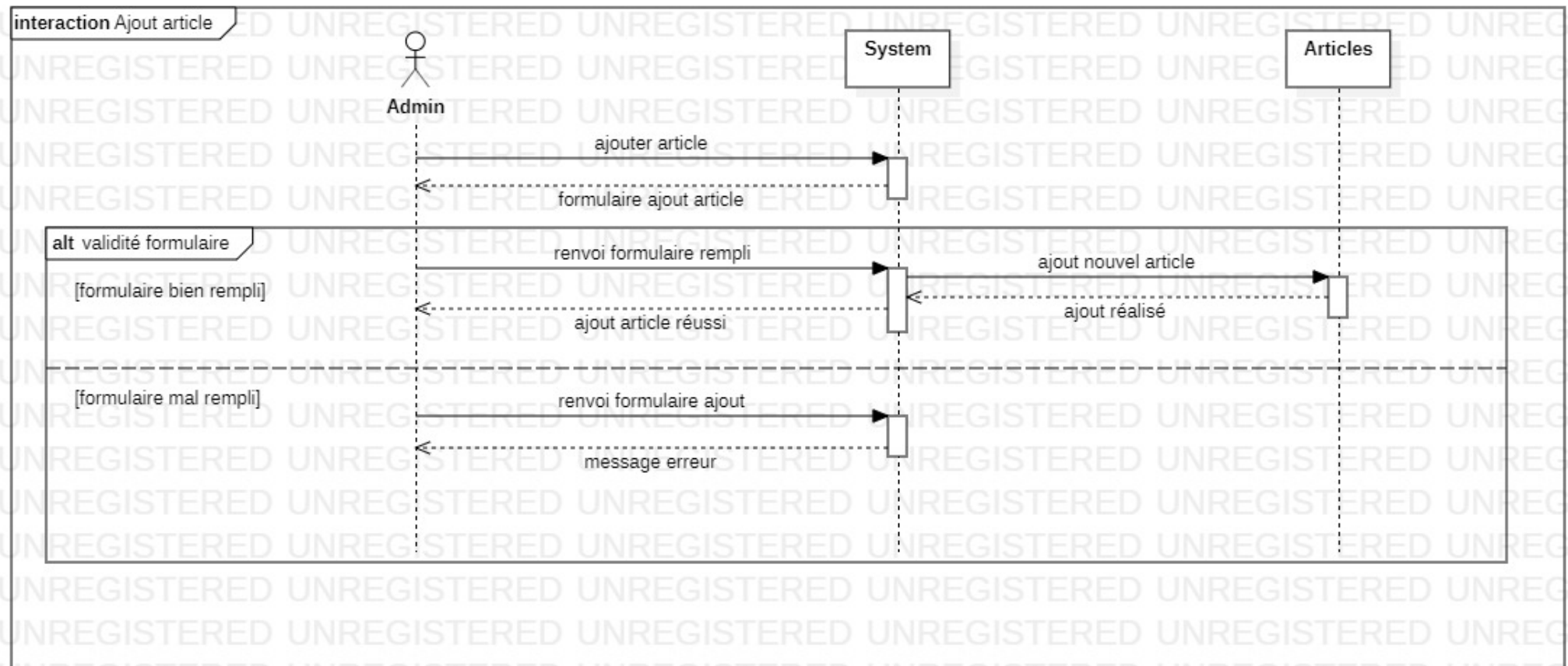
- Page d'accueil avec formulaire du contact
- Liste des articles
- Détail de chaque article avec possibilité de soumettre un commentaire soumis à approbation
- Formulaire d'inscription
- Formulaire de connexion à l'espace membre et espace admin avec système de récupération de mot de passe

B) Analyse du besoin pour l'administration des articles

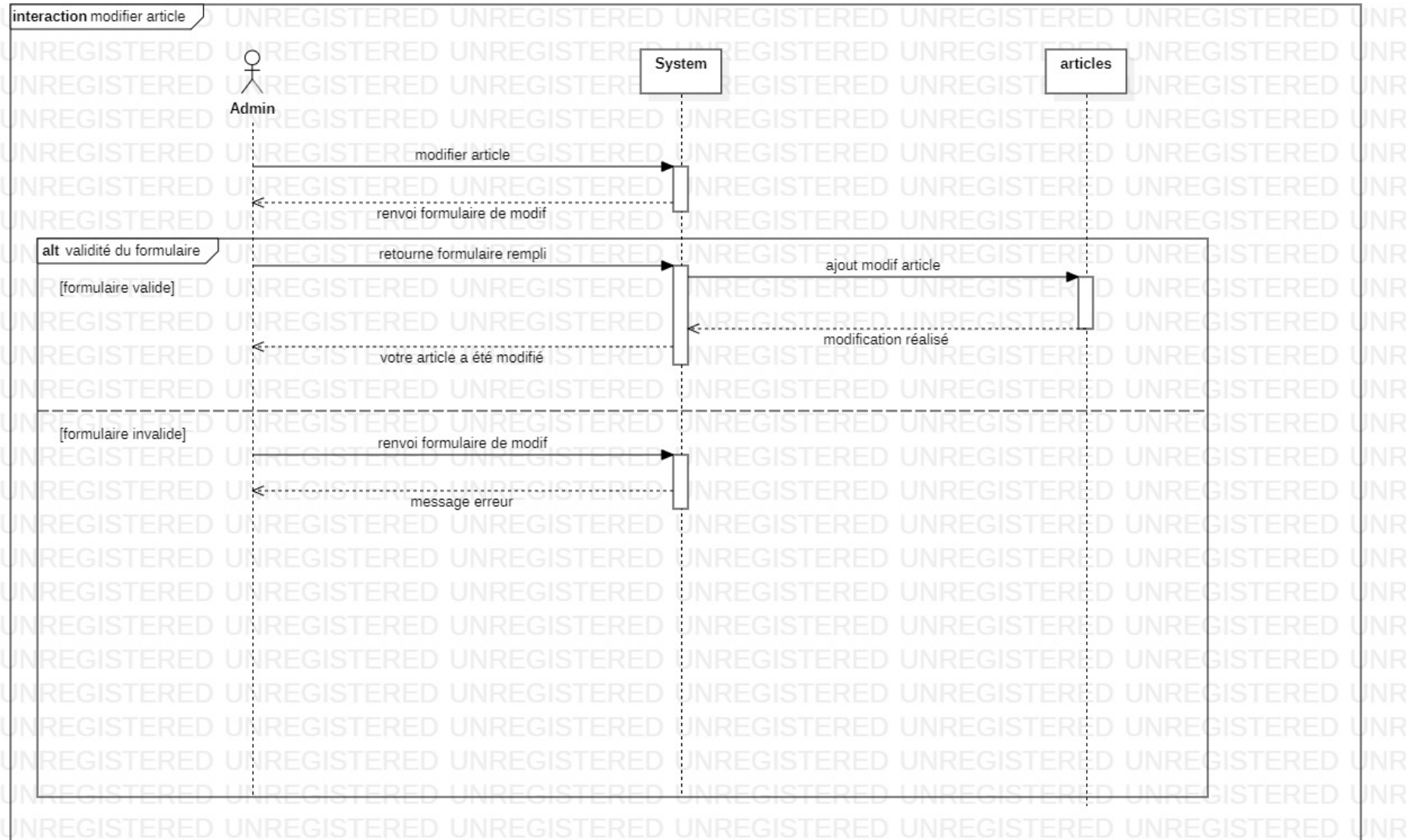
- Diagramme de cas d'utilisation



- Diagramme de séquence, ajout article.



- Diagramme de séquence, modification article.



- Diagramme de séquence, suppression article.

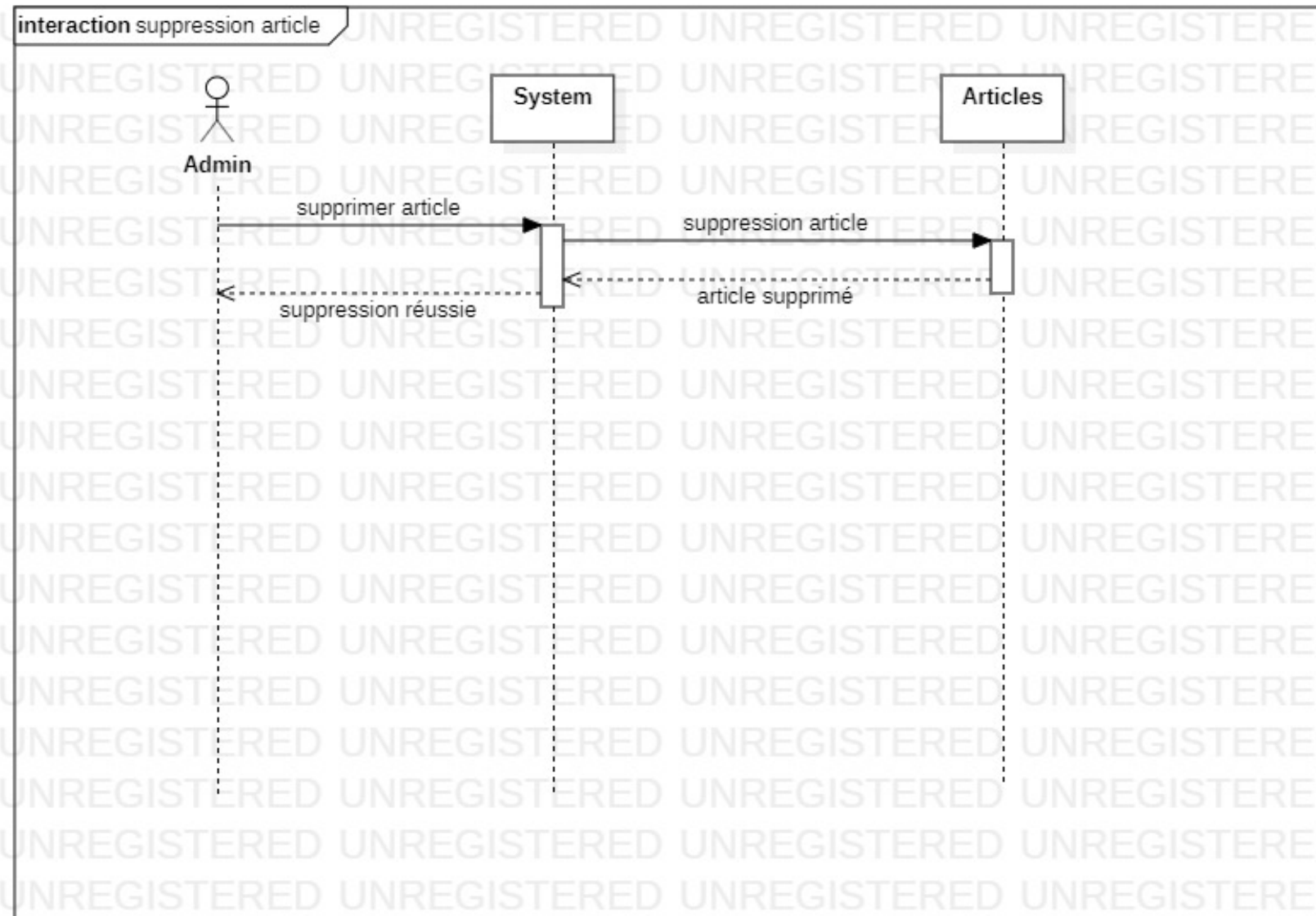
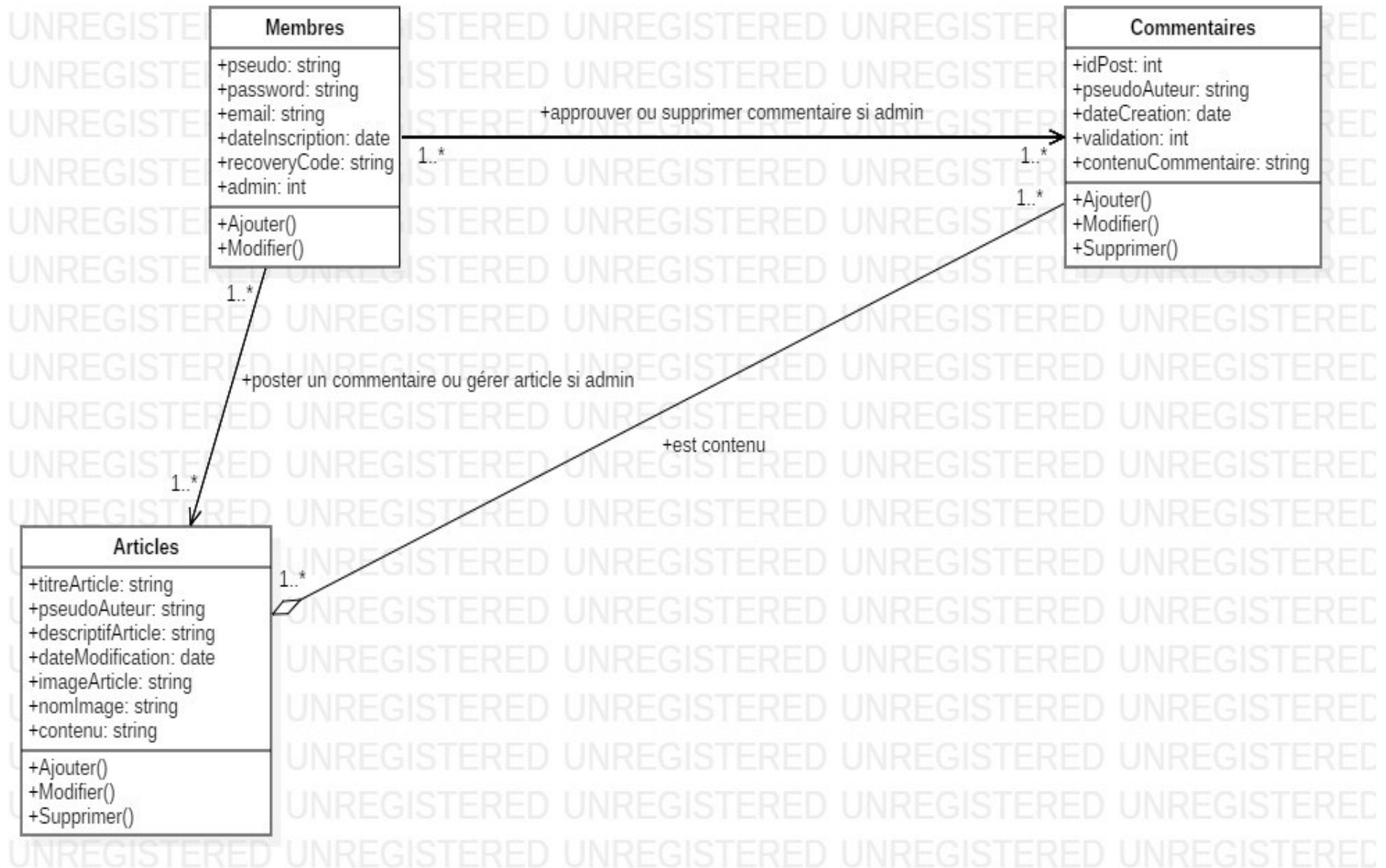


Diagramme de classe.



C) Organisation du projet

- Choix du template
- Création des issues
- Création d'une branche par tâche
- Réalisation de la tâche
- Transfert de la branche sur le dépôt distant
- Validation du code par le mentor
- Clôture des issues
- Analyse du code sur Codacy
- Apport de correctifs au code
- Réalisation des diagrammes et du fichier d'installation
- Soumission du projet

D) Versionning du projet.

All branches						
master	Updated yesterday by ckanzafluks		Default			Change default branch
code_improvement	Updated 6 days ago by Steve237	✓	2 0	#29	Merged	
secure	Updated 15 days ago by Steve237		16 0		New pull request	
site_redesigning	Updated 24 days ago by Steve237		23 0	#20	Merged	
admin_space	Updated 29 days ago by Steve237	✓	31 0	#19	Merged	
admin_space_connection	Updated 2 months ago by Steve237		45 0	#15	Merged	
myblogdatabase	Updated 2 months ago by Steve237		89 0	#14	Merged	
recovery_password	Updated 2 months ago by Steve237		50 0	#12	Merged	
member_space	Updated 3 months ago by Steve237		56 0	#11	Merged	
blogposts	Updated 3 months ago by Steve237		58 0	#10	Merged	

E) Architecture technique du projet.

- Utilisation du pattern MVC (Modèle vue contrôleur) :

-Modèle : cette partie a pour fonction d'aller récupérer les informations dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. Elle regroupe donc entre autres les requêtes SQL.

-Vue : cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple une liste de messages.

-Contrôleur : cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès) .

- Utilisation des mediaqueries et de Bootstrap pour adapter le site en responsive.
- Langages utilisés : Html5, CSS3, PHP en programmation orienté objet, Javascript, JQuery

F) Mesures mis en place pour le respect des pratiques en vigueur

- Les mesures que nous avons utilisés pour le respect des normes de codages sont :
 - L'indentation du code.
 - L'écriture du nom des variables et fonctions en Camelcase (PSR-1).
 - L'écriture du nom des classes en StudlyCaps (PSR-1).
 - Utilisation de Autoload pour le chargement des classes (PSR-1).
 - Écriture des structures de contrôle (else, elseif, etc...) conformément au PSR-2
 - Sécurisation du site contre les failles de sécurité (failles XSS, CRLF, injection sql, la CSRF, session hijacking, hashage des mots de passes en base de données) .
 - Redirection vers page 404 en cas d'erreur de requête.
 - Commentaires des fonctions des modèles.
 - Utilisation du pattern MVC.
 - Adaptation du site pour tous les types d'écran (smartphone, tablette, pc) .
 - Utilisation d'outil de versionning.
 - Analyse du code via Codacy et apport de correctifs au code

G) Exemple de correctifs apportés au code suite analyse Codacy

Ci dessous une analyse de Codacy

Avoid variables with short names like \$db. Configured minimum length is 3.

```
81 $db = $this->dbconnect();
```

model/UsersManager.php

Avoid variables with short names like \$q. Configured minimum length is 3.

```
14 $q = $db->prepare('INSERT INTO membres(pseudo, password, email, dateInscription) VALUES(:pseudo, :password, :email, NOW())');
```

Ci dessous la correction apportée suite à cette analyse

<?php

```
class CommentManager extends Manager
{
```

```
/**
 * permet d'afficher la liste des commentaires
 */
```

```
public function getListComment()
{
```

```
    $dtb = $this->dbconnect();
    $query = $dtb->query('SELECT idCommentaire, idPost, pseudoAuteur, contenuCommentaire, DATE_FORMAT(dateCreation, "%d/%m/%Y %Hh%imin%ss") AS
    dateCreation, validation FROM commentaires');
    $query->execute();
```

```
    $comments = $query->fetchAll(PDO::FETCH_ASSOC);
```

```
    $listComments = array();
```

```
    foreach ($comments as $donnees)
```

```
    {
        // on instancie notre objet
        $comments = new Comments();
        // on hydrate notre objet avec les valeurs récupérées en bdd
        $comments->hydrate($donnees);
        // puis on le met dans notre tableau
        $listComments[] = $comments;
    }
```

```
    return $listComments;
```

```
}
```

H) Présentation de la réalisation de la tâche espace administrateur

La tâche création du back-office a été réalisée en plusieurs étapes :

- Création du formulaire de connexion à l'espace administrateur
- Affichage de la liste des articles à gérer
- Création des boutons pour ajouter, modifier, supprimer un article.
- Création des formulaires d'ajout, et de modification des articles.
- Écriture des fonctions dans le modèle et le contrôleur pour le fonctionnement de l'ajout, modification, ou suppression des articles.
- Création d'un lien pour accéder à l'espace de gestion des commentaires.
- Affichage de la liste des commentaires à gérer dans l'espace gestion des commentaires.
- Création des boutons pour approuver ou supprimer des commentaires.
- Écriture des fonctions dans le modèle et le contrôleur pour le fonctionnement de l'approbation ou la suppression des commentaires.
- Embellissement de l'affichage dans l'espace administrateur en Css3