

---

## Cahier Des Charges:

---

**EG Développement**  
Développeur PHP/Symfony



# SOMMAIRE

## I. CADRE DU PROJET

## 2. SPECIFICATIONS FONCTIONNELLES

### 2.1. Identification des acteurs et des besoins

#### 2.1.1. Les acteurs

#### 2.1.2. Userstories

### 2.2. Représentation des besoins par l'UML

#### 2.2.1. Diagramme de packages

#### 2.2.2. Diagramme de cas d'utilisation - gestion des posts

#### 2.2.3. Diagramme de cas d'utilisation - gestion des commentaires

#### 2.2.4. Diagramme de cas d'utilisation - gestion de l'authentification

### 2.3. Détails de fonctionnalités

#### 2.3.1. Lecture d'un post et ses commentaires

#### 2.3.2. Ajout d'un commentaire

#### 2.3.3. Modification d'un post

#### 2.3.4. Gestion d'un commentaire

#### 2.3.5. Authentification

#### 2.3.6. Création d'un compte utilisateur

### 2.4. Diagramme de classes

### 2.5. Arborescence

## 3. SPECIFICATIONS TECHNIQUES

### 3.1. Développement du site

### 3.2. Les besoins techniques

### 3.3. Formulaire de contact

### 3.4. Responsive

## 4. CHARTE GRAPHIQUE

### 4.1. Polices et les couleurs du blog

### 4.2. Le visuel

## I. CADRE DU PROJET

Conception d'un blog professionnel composé des pages suivantes:

- la page d'accueil
- la page listant l'ensemble des blog posts ;
- la page affichant un blog post ;
- la page permettant d'ajouter un blog post ;
- la page permettant de modifier un blog post ;
- les pages permettant de modifier/supprimer un blog post ;
- les pages de connexion/enregistrement des utilisateurs.

Certaines fonctionnalités ne devront être accessibles que par l'administrateur du site ou utilisateurs autorisés.

## 2. SPÉCIFICATIONS FONCTIONNELLES

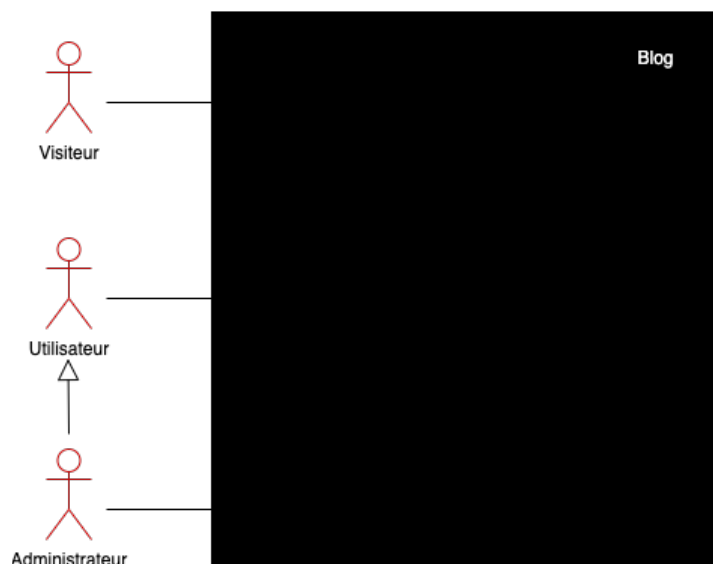
### 2.1. Identification des acteurs et des besoins

#### 2.1.1. Les acteurs

Sont représentés ici les acteurs principaux qui utilisent les fonctionnalités du blog:

- **L'utilisateur** détient un compte utilisateur. S'il a le rôle d'**administrateur** alors il accède à toutes les fonctionnalités du blog.
- **Le visiteur** n'a pas de compte utilisateur, ses actions seront donc limitées.

Diagramme de contexte



### 2.1.2. Userstories

Pour récapituler les besoins spécifiés plus haut, voici la liste des actions que chaque acteur souhaite effectuer sur le blog:

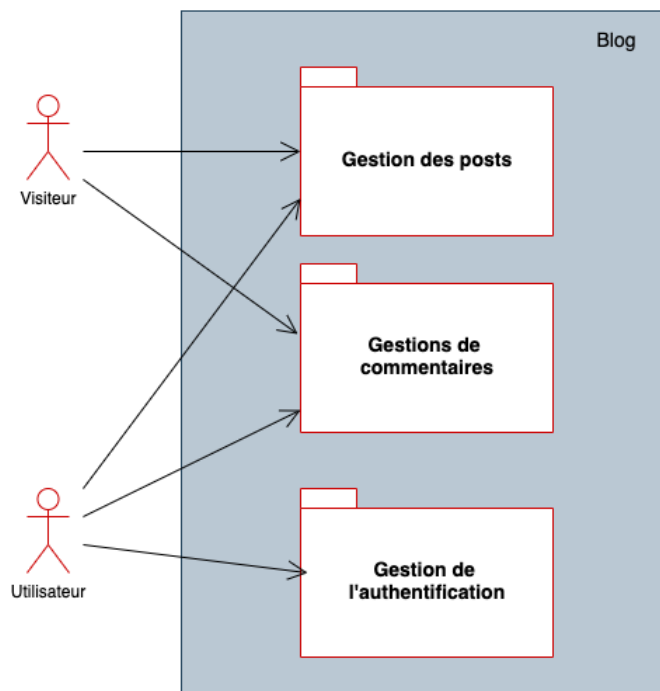
Le visiteur veut :	L'utilisateur veut:	L'administrateur veut:
consulter la liste des posts	se connecter	ajouter un post
lire un post et ses commentaires	écrire un commentaire	modifier un post
envoyer un mail de contact		supprimer un post
accéder aux réseaux sociaux		valider des commentaires

## 2.2. Représentation des besoins par l'analyse UML

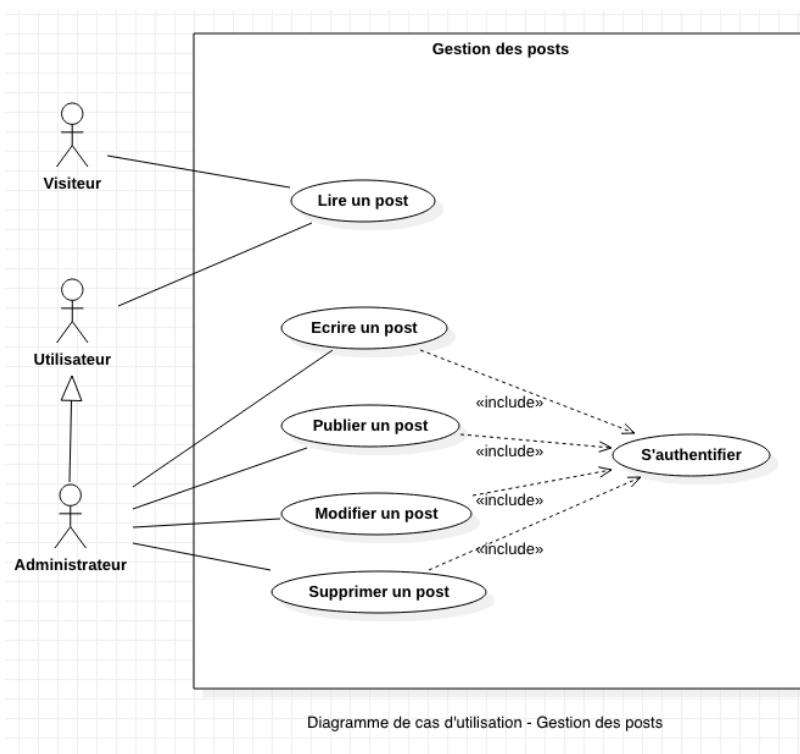
### 2.2.1. Diagramme de packages

Les fonctionnalités peuvent être regroupées en trois grandes parties (« packages »): la gestion des posts, la gestion des commentaires et la gestion de l'authentification. Elles sont illustrées dans ce diagramme (certaines fonctionnalités seront détaillées dans les prochains chapitres par les diagrammes de cas d'utilisation et de séquences). Pour plus de clarté, seul l'utilisateur dans sa signification globale est représenté.

Diagramme de packages

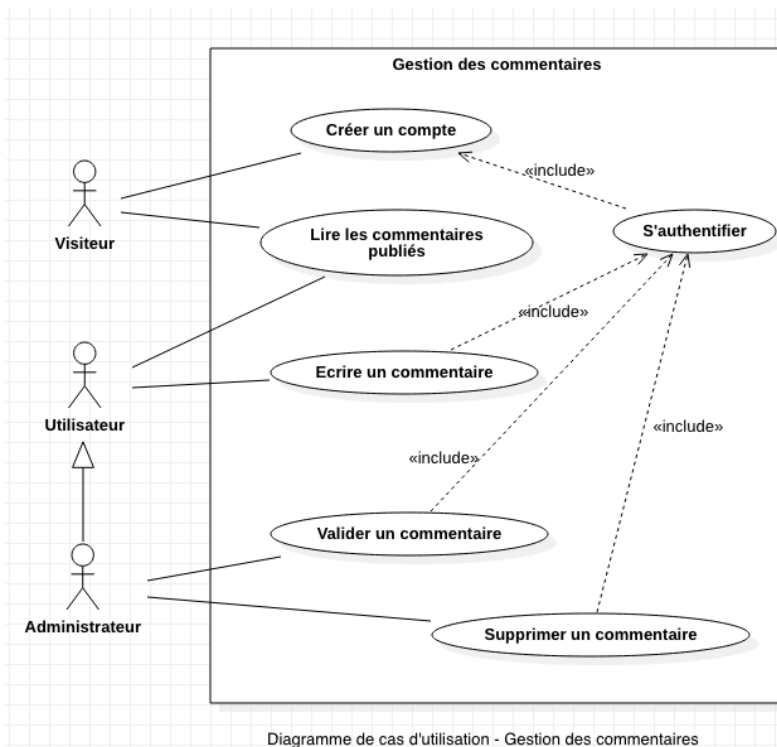


### 2.2.2. Diagramme de cas d'utilisation - gestion des posts



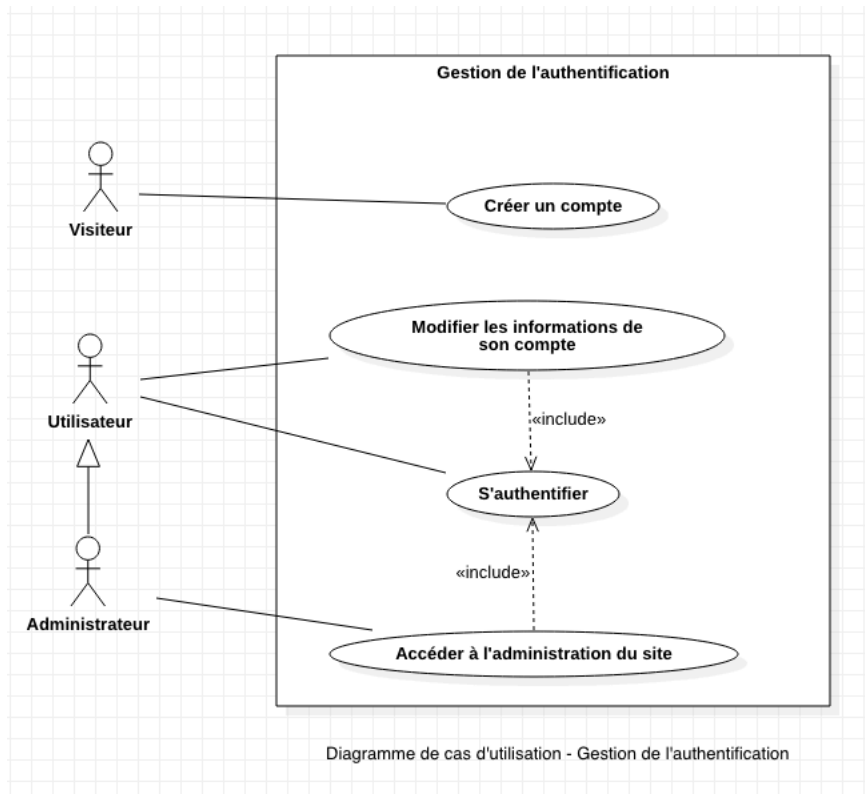
Seul l'utilisateur ayant un rôle d'administrateur peut intervenir dans la gestion des posts. Les autres utilisateurs et les visiteurs peuvent uniquement lire les posts.

### 2.2.3. Diagramme de cas d'utilisation - gestion des commentaires



Ici, seul l'utilisateur pourra écrire un commentaire car il faut être authentifié. Le visiteur pourra cependant les lire et devra créer un compte pour en écrire. L'administrateur sera le seul à valider et supprimer les commentaires en attente.

#### 2.2.4. Diagramme de cas d'utilisation - gestion de l'authentification



Le visiteur devra créer un compte en amont pour s'authentifier.

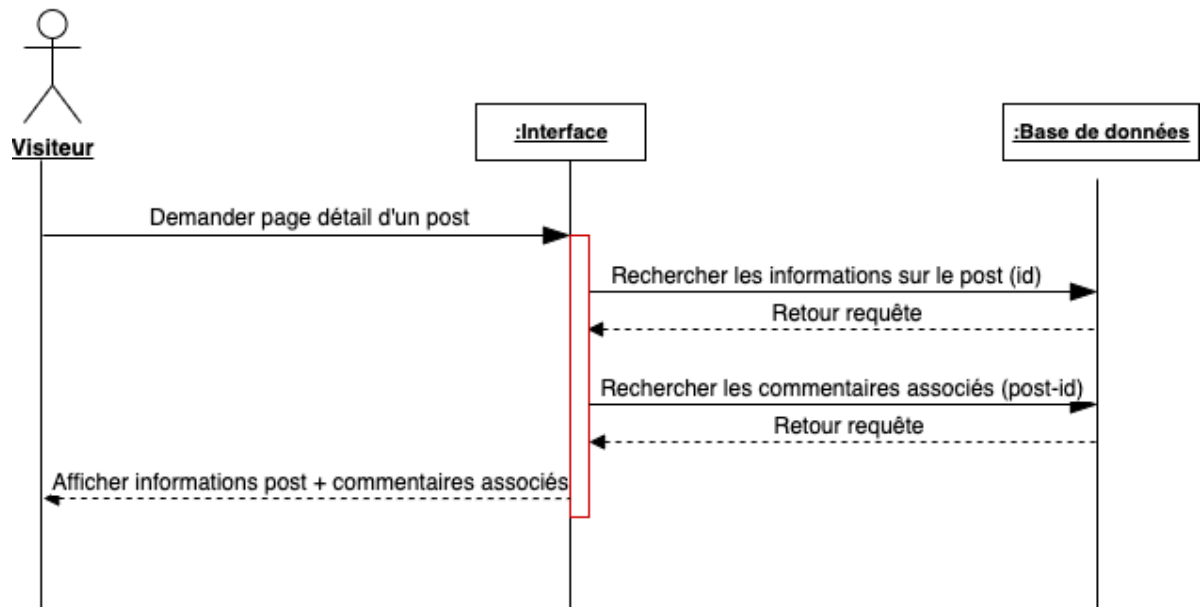
L'utilisateur pourra modifier les informations de son compte mais être authentifié au préalable.

Seul l'administrateur aura accès aux pages d'administration du blog.

## 2.3. Détails de fonctionnalités

### 2.3.1. Lecture d'un post et ses commentaires

Diagramme de séquence - Lecture d'un post et ses commentaires

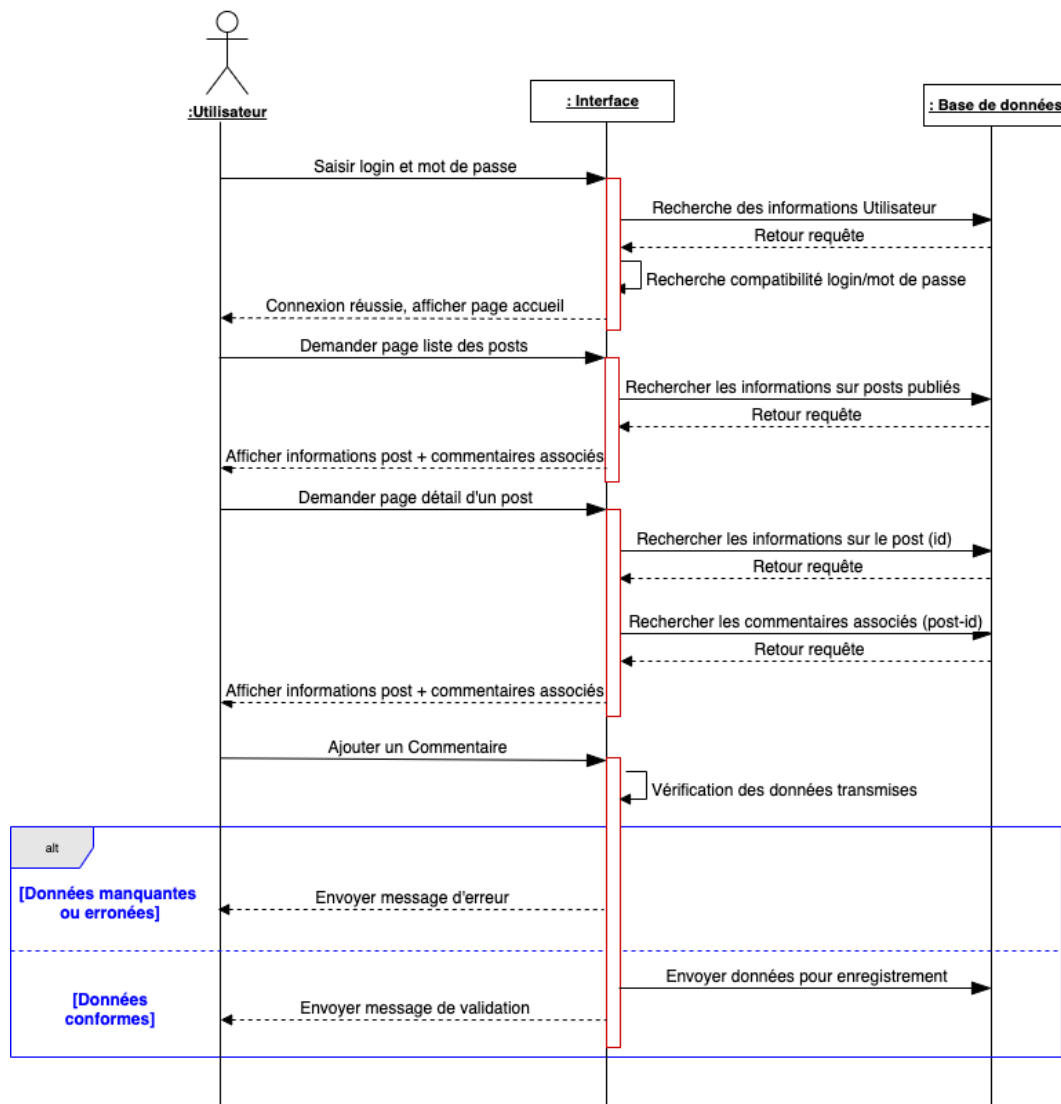


Dans le diagramme de séquence ci-dessus:

- le visiteur demande d'accéder à la page de détails d'un post: deux requêtes sont lancées en base de données pour récupérer les informations de ce post ainsi que les commentaires associés. Le système retourne la page demandée.

### 2.3.2. Ajout d'un commentaire

Diagramme de séquence - Ajout d'un commentaire



Dans cette séquence:

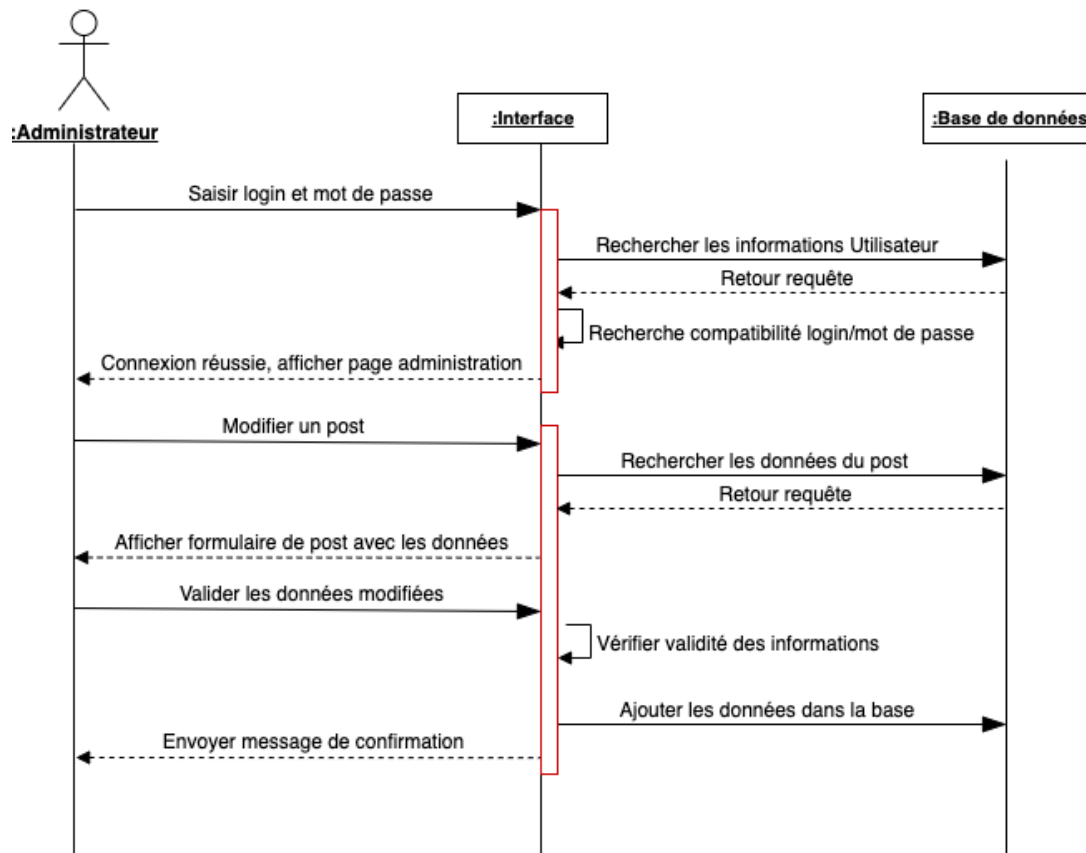
- **l'utilisateur s'authentifie**: Si le couple login/mot de passe est correct, la session s'ouvre et le système affiche la page d'accueil.
- **l'utilisateur souhaite ensuite consulter la liste des posts puis va en sélectionner un**: le système demande à la base de données de lui fournir les détails du post demandé et la liste des commentaires associés et validés.
- **l'utilisateur ajoute son commentaire**: le système affiche un message d'erreur en cas de données manquantes/erronées ou un message de validation si tout est correctement renseigné. Dans ce cas, il enverra les données pour enregistrement en base de données.

Le commentaire ne sera pas affiché sur la page avant sa validation par l'administrateur (voir le point 2.3.4. *Diagramme de séquence -validation d'un commentaire*).



### 2.3.3. Modification d'un post

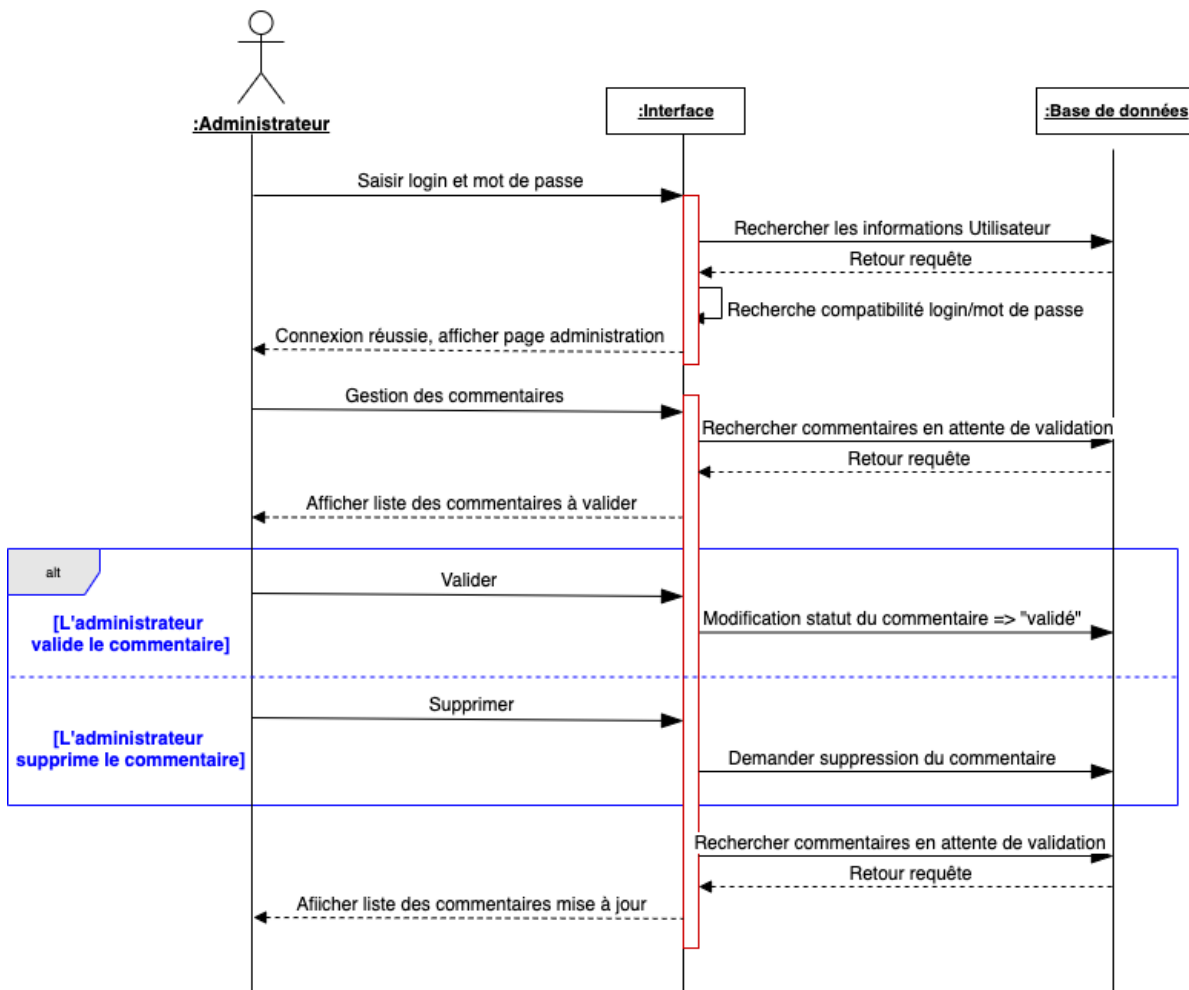
Diagramme de séquence - Modification d'un post



- **l'administrateur s'authentifie**: Si le couple login et mot de passe est correct, la session s'ouvre et le système affiche la page d'administration.
- **l'administrateur demande la page de modification de post**: le système effectue une requête puis affiche les informations du post sélectionné.
- **l'administrateur valide les informations modifiées**: le système envoie les données dans la base, après avoir vérifié leur conformité, et affiche un message de confirmation à l'administrateur.

## 2.3.4. Gestion d'un commentaire

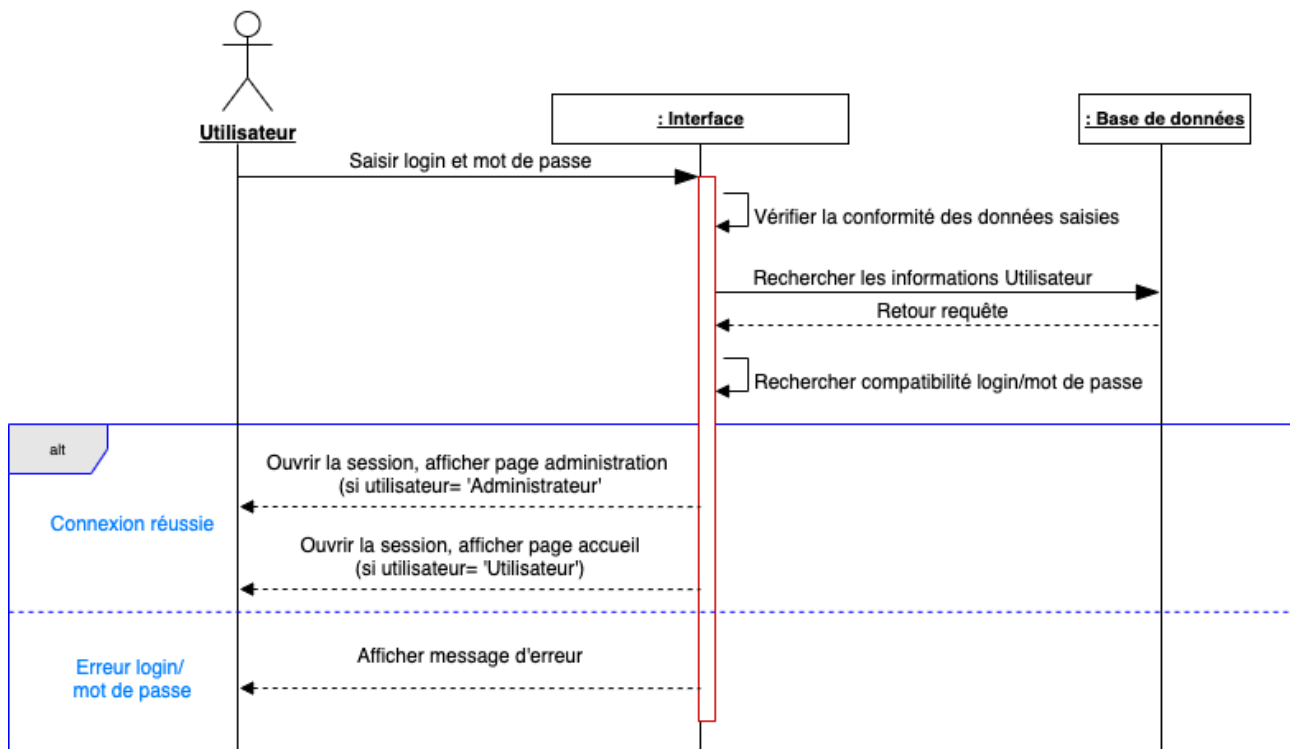
Diagramme de séquence - Gestion d'un commentaire



- **l'administrateur s'authentifie**: Si le couple login et mot de passe est correct, la session s'ouvre et le système affiche la page d'administration.
  - **l'administrateur demande la liste des commentaires à valider**: le système effectue une requête en base de données et affiche la liste des commentaires ayant pour statut « non validé ».
  - **l'administrateur choisit de valider le commentaire**: le statut du commentaire est modifié en base de données : « validé ».
  - OU **l'administrateur choisit de supprimer le commentaire**: le système demande la suppression du commentaire en base de données.
- Le système affiche la liste des commentaires à valider sans le commentaire précédemment traité.

### 2.3.5. Authentification

Diagramme de séquence - Authentification

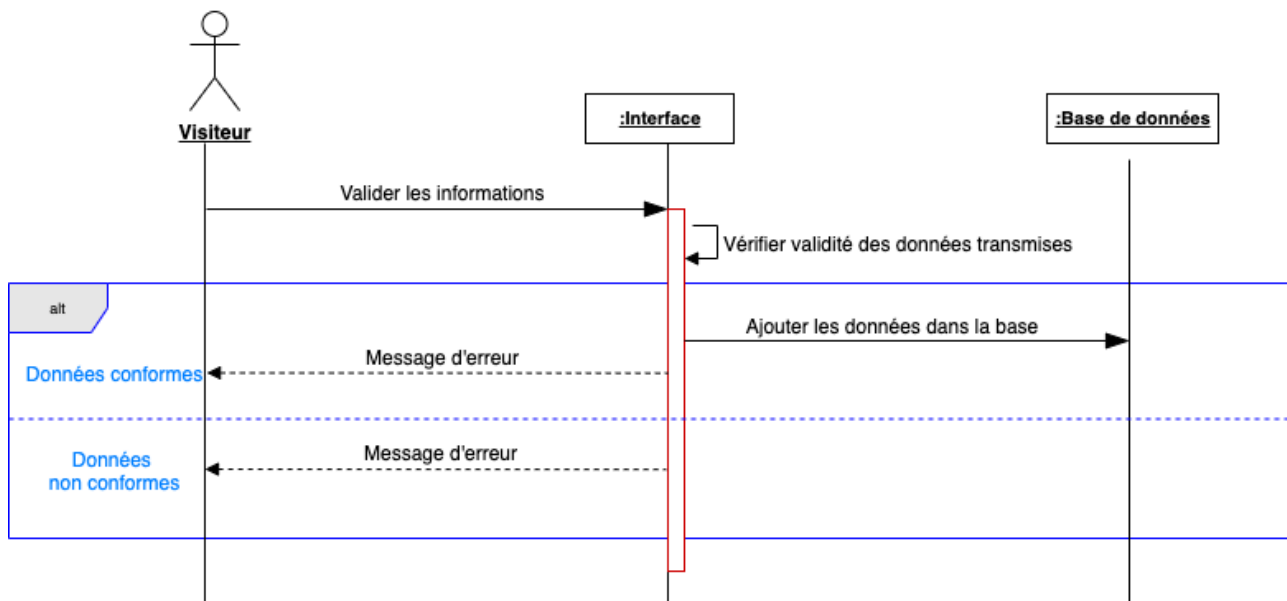


Ce diagramme détaille le processus d'authentification et la gestion des erreurs.

- **l'utilisateur saisit son login et son mot de passe**: le système vérifie la conformité des informations saisie et interroge la base de données. Il vérifie l'existence du login et la compatibilité avec le mot de passe saisi.
- **la connexion est réussie**: la session s'ouvre et une page s'affiche selon le rôle de l'utilisateur. L'administrateur accédera directement à la page administration.
- **la connexion a échouée**: un message d'erreur s'affiche et l'utilisateur est invité à rejoindre la page de connexion.

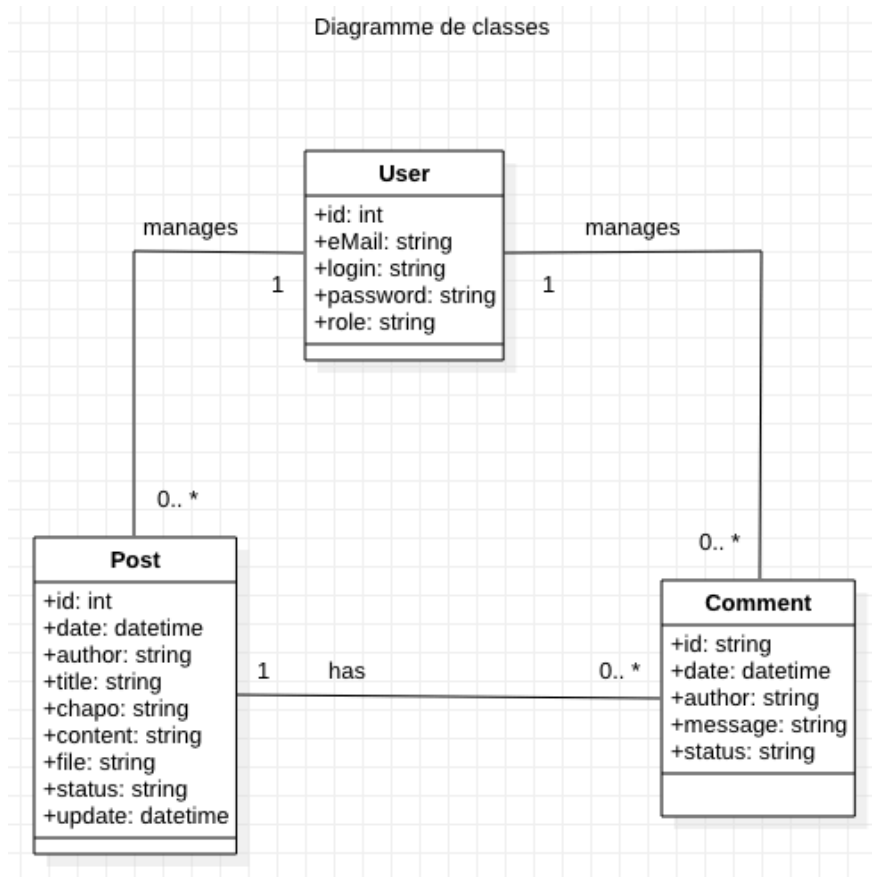
### 2.3.6. Création d'un compte utilisateur

Diagramme de séquence - Création d'un compte utilisateur



- le visiteur renseigne les informations nécessaires à la création d'un compte utilisateur: le système vérifie l'existence et la conformité des données.
- les données sont conformes: un message de validation s'affiche et l'utilisateur est invité à rejoindre la page d'accueil.
- la connexion a échoué: un message d'erreur s'affiche et l'utilisateur est invité à rejoindre la page de connexion.

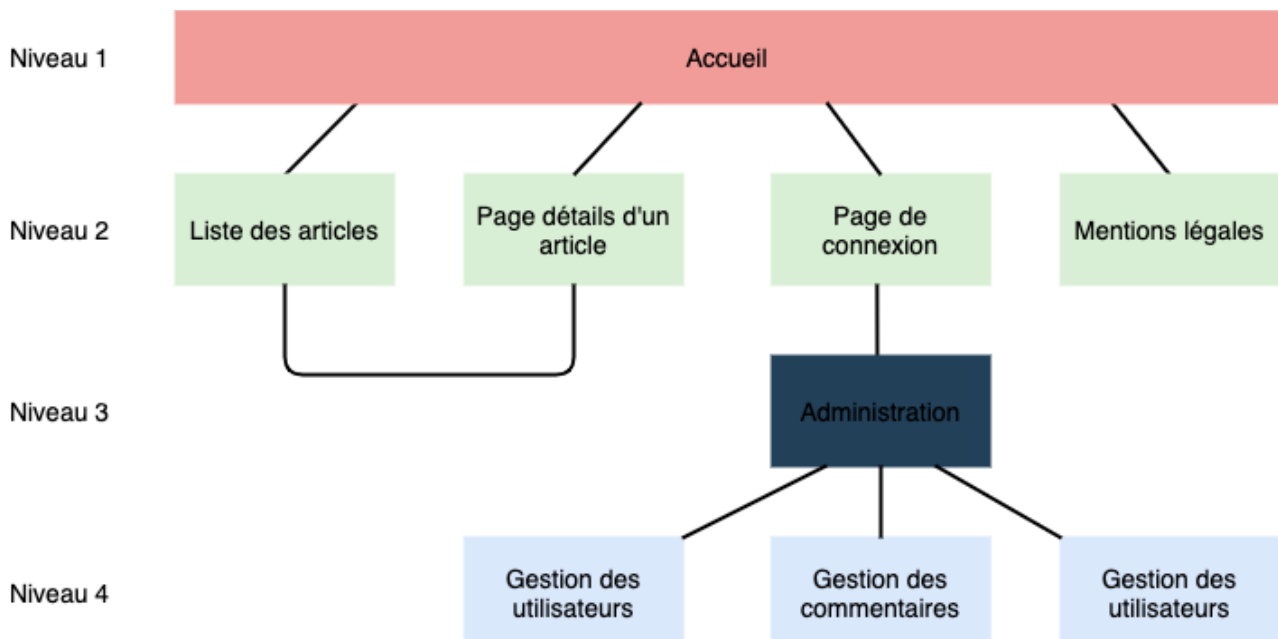
## 2.4. Diagramme de classes



Trois classes interagissent entre elles:

- un **utilisateur** peut gérer un ou plusieurs posts ou commentaires en fonction de son rôle.
- un **post** peut détenir ou non des **commentaires**.

## 2.5. Arborescence du blog



## 3. SPÉCIFICATIONS TECHNIQUES

### 3.1. Développement du site

Le développement du blog sera effectué from scratch dans le langage HTML 5, CSS3 et PHP version 7.3.8 (serveur PHP: Apache version 2.2.34).

Pour la gestion du projet, un dossier nommé « *BlogP5* » sera créé sur GitHub.

### 3.2. Les besoins techniques

La base de données est hébergée sur un serveur MySQL version 5.6.

Composer et son autoloader seront installés sur le projet en vue de la programmation en MVC (Model-View-Controller).

### 3.3. Le formulaire de contact

Un formulaire de contact sera créé afin que les visiteurs et utilisateurs du blog puissent contacter l'administrateur. Les informations de ce formulaire seront transmises à l'adresse mail de l'administrateur: [edwige.gentymail@gmail.com](mailto:edwige.gentymail@gmail.com) grâce à PHPmailer.

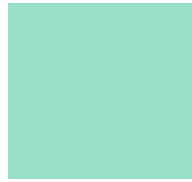
### 3.4. Responsive

Le blog sera responsive c'est à dire qu'il sera lisible sur différents supports: ordinateur, tablette, et mobile.

## 4. CHARTE GRAPHIQUE

### 4.1. Polices et couleurs du blog

#### Les couleurs:



#### Les polices:

- Titre principal:

**PERMANENT MARKER**

- Sous-titres:

*Pacifico*

- Texte:

Arial

### 4.2. Visuel

Le framework Bootstrap version 4.5.3 sera utilisée pour la création du visuel. Il est notamment conçu pour adapter le visuel en fonction de la taille de l'écran sur lequel le blog sera visité (responsive).