**Dossier d’analyse et de conception**

**SAE.S5.C.01**

Proposer une solution optimisée à partir de données

Sommaire

Table des matières

[Objectifs 4](#_Toc150853115)

[Énumérations des fonctions 4](#_Toc150853116)

[Fonctions administrateur : Administrer le site 4](#_Toc150853117)

[Fonctions utilisateurs : Chercher et regarder des séries 4](#_Toc150853118)

[Description des grandes fonctionnalités 5](#_Toc150853119)

[Cas d’utilisation : Regarder une série 5](#_Toc150853120)

[Diagramme de séquence : Regarder une série 6](#_Toc150853121)

[Cas d’utilisation : La recommandation de séries 7](#_Toc150853122)

[Diagramme de séquence : Ajouter une série 8](#_Toc150853123)

[Architecture générale du site 8](#_Toc150853124)

[Arborescence du site 8](#_Toc150853125)

[Administrateur 8](#_Toc150853126)

[Utilisateur 10](#_Toc150853127)

[Description des interfaces et maquettes d’écran 11](#_Toc150853128)

[Page de connexion 11](#_Toc150853129)

[Accueil 12](#_Toc150853130)

[Gestion administrateur 12](#_Toc150853131)

[Créer compte administrateur 13](#_Toc150853132)

[Gestion utilisateur 13](#_Toc150853133)

[Modèles de données 14](#_Toc150853134)

[Schémas MongoDB 14](#_Toc150853135)

[Collection Series\_Mots 14](#_Toc150853136)

[Description : 14](#_Toc150853137)

[Collection Utilisateur 15](#_Toc150853138)

[Description : 16](#_Toc150853139)

[Collection Mots\_Series 16](#_Toc150853140)

[Description : 17](#_Toc150853141)

[Informations supplémentaires 17](#_Toc150853142)

# Objectifs

La demande consiste à réaliser un site web vitrine pour la diffusion de séries TV. Nous devons mettre en place une fonction de recherche basée sur les textes de sous-titres fournis par le client. De plus, le site devrait être en mesure de recommander des séries pertinentes en fonction de l’activité de l’utilisateur.

# Énumérations des fonctions

Les fonctions principales attendues se divisent en 2 catégories principales : Administrateur et Utilisateur

## Fonctions administrateur : Administrer le site

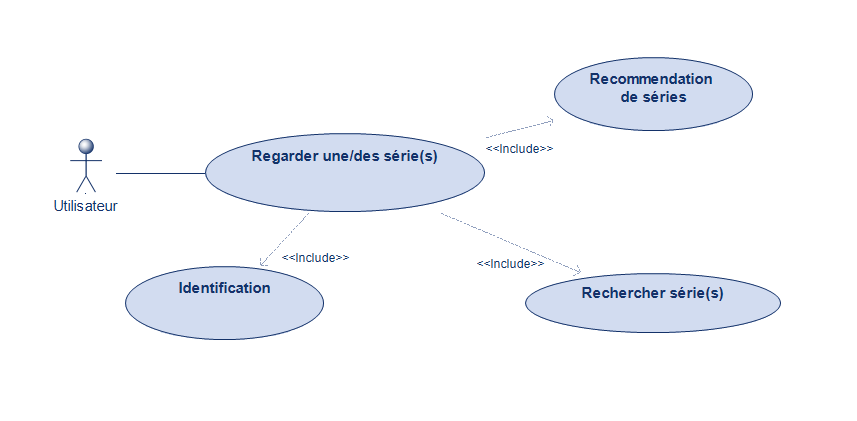
* Se connecter/déconnecter
* Créer un nouveau compte administrateur
* Supprimer un utilisateur

## Fonctions utilisateurs : Chercher et regarder des séries

* Consulter des séries
* Rechercher des séries
* Créer un compte utilisateur
* Se connecter/déconnecter

# Description des grandes fonctionnalités

## Cas d’utilisation : Regarder une série



**Rôle :** Ce cas d’utilisation est déclenché à chaque fois qu’un utilisateur se connecte sur la plateforme de streaming du client.

**Début :** Quand l’utilisateur s’authentifie sur le site.

**Fin :** Quand l’utilisateur ferme le site.

**Précondition :** Nécessite un compte utilisateur

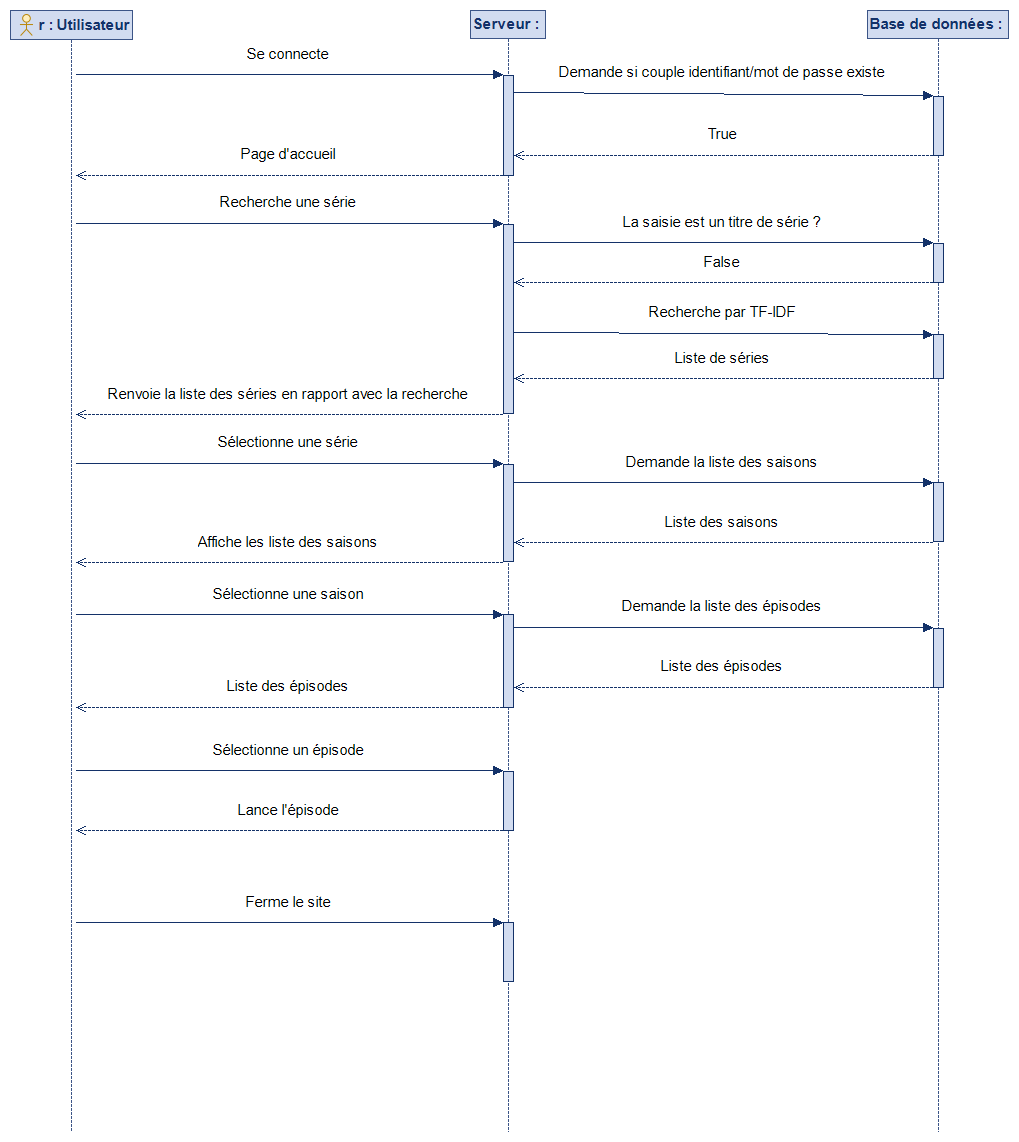
**Postcondition :** Néant

**Données en entrée :** Néant

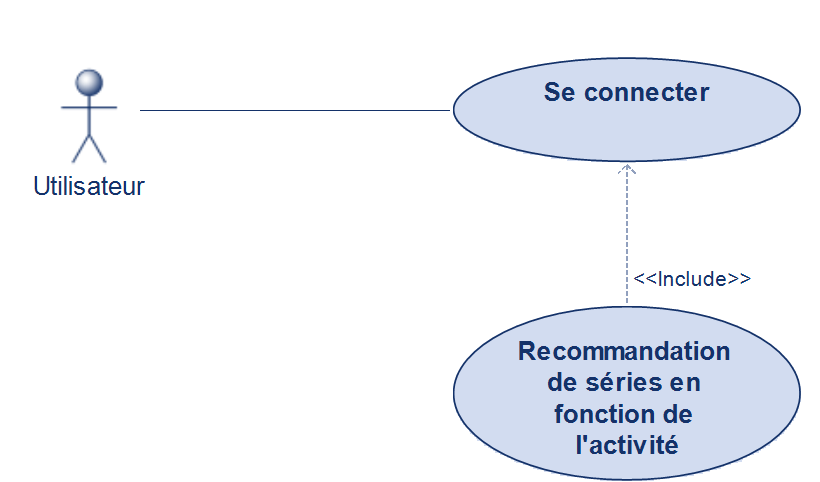
**Données en sortie :** Néant

**Exception :** Problème de connexion avec le serveur.

## Diagramme de séquence : Regarder une série



## Cas d’utilisation : La recommandation de séries



**Rôle :** Ce cas d’utilisation est déclenché lorsqu’un utilisateur se connecte sur son compte.

**Début :** Quand l’utilisateur s’authentifie sur le site.

**Fin :** Quand l’utilisateur ferme le site

**Précondition :** Nécessite d’avoir un compte, de se connecter et d’avoir eu une activité sur le compte.

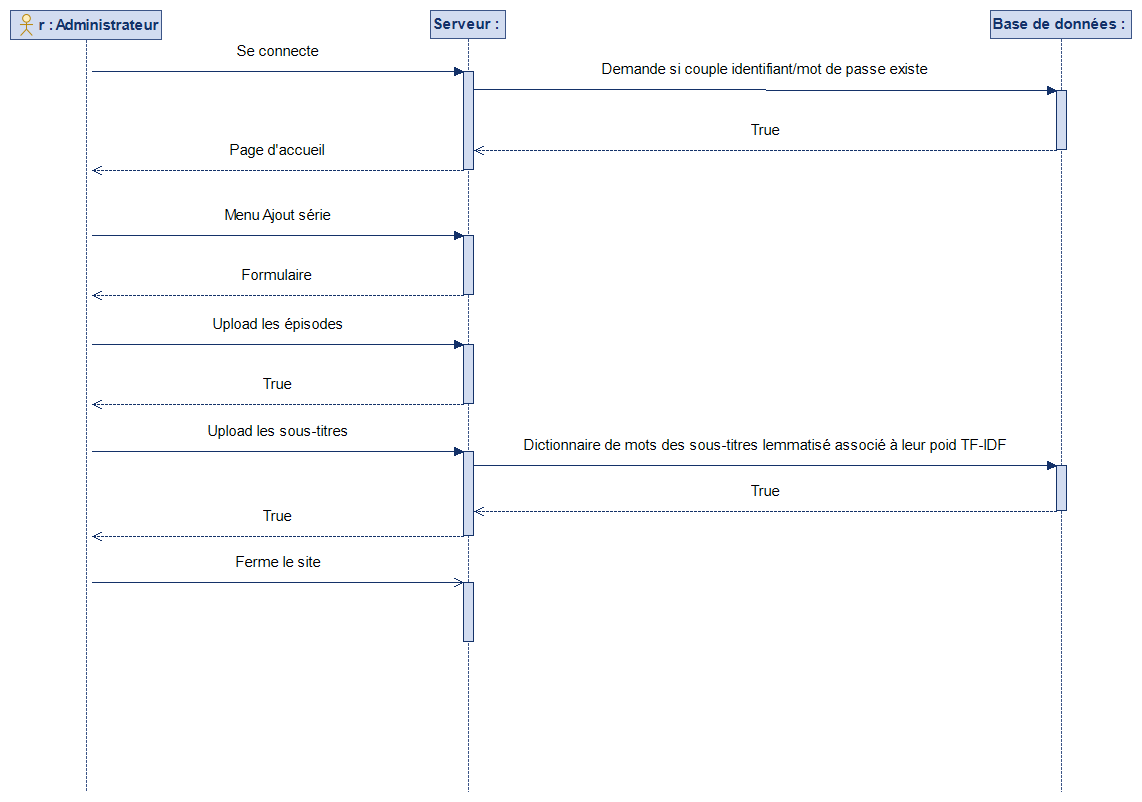
**Postcondition :** Néant

**Données en entrée :** Néant

**Données en sortie :** Néant

**Exception :** Problème de connexion avec le serveur.

## Diagramme de séquence : Ajouter une série

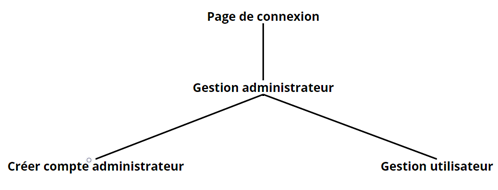


# Architecture générale du site

Le site sera hébergé sur le serveur de l’IUT. Toutes les informations relatives aux données des utilisateurs/administrateurs et des mots liés à l’algorithme de recherche seront stockées dans la base de données. Les épisodes des séries seront stockés sur le serveur de l’iut.

## Arborescence du site

### Administrateur



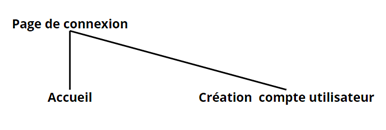
**Page de connexion :** La connexion d’un administrateur nécessite au préalable la création de son compte par un autre administrateur. Une fois le compte créé il peut se connecter avec son identifiant et son mot de passe. Son mot de passe est envoyé au contrôleur qui effectue la connexion avec la base de données et envoie le mot de passe sous forme de hash. Il retourne au client un refus ou une validation de connexion.

**Gestion administrateur :** Cette page fait office de hub pour les administrateurs. Ils peuvent avoir accès aux pages “gestion des utilisateurs” et “Ajouter un administrateur”. De plus, ils ont accès à une barre de recherche des séries. La barre de recherche fonctionne en mots clés. Une fois la recherche effectuée, chaque mot sera lemmatisé. Sur notre base de données on regardera les correspondances trouvées avec les textes des séries. Une liste de séries sera composée par poids décroissant associés aux correspondances des mots trouvées. Il faut savoir que chaque série se voit associée à une liste de mots récupérés par les fichiers sous-titres anglais et français.   
Finalement, la liste de série sera renvoyée graphiquement à l’interface client.

**Créer compte administrateur :** Lors de la validation du formulaire, les informations saisies sont envoyées au contrôleur. Le contrôleur fait un hash du mot de passe. Il effectue une connexion avec la base de données. Il exécute une requête d’insertion de données composé d’un mot de passe hashé, d’un identifiant et d’un rôle administrateur.

**Gestion utilisateur :** L’en-tête de la page propose les mêmes fonctionnalités que la page “gestion administrateur”. Cette page se compose de la liste de tous les utilisateurs (identifiant et rôle). On laissera la possibilité à l’administrateur de supprimer un utilisateur. Lors de la suppression, une requête est envoyée au contrôleur, le contrôleur crée une connexion et envoie une requête de suppression d’utilisateur dans la base de données.

### Utilisateur



**La connexion d’un utilisateur** : l’utilisateur devra dans un premier temps rentrer ses identifiants, s’il a déjà créé un compte sinon il cliquera sur le bouton de création du compte et sera redirigé sur la page de création de compte. Sur cette page lui seront demandées plusieurs informations obligatoires comme un pseudo, un mot de passe, une adresse mail et des informations de carte bleue. Toutes ses informations seront stockées en brut excepté toutes les informations relatives à sa carte bleue et le mot de passe qui passeront par une fonction de hachage avant d’être stockées.

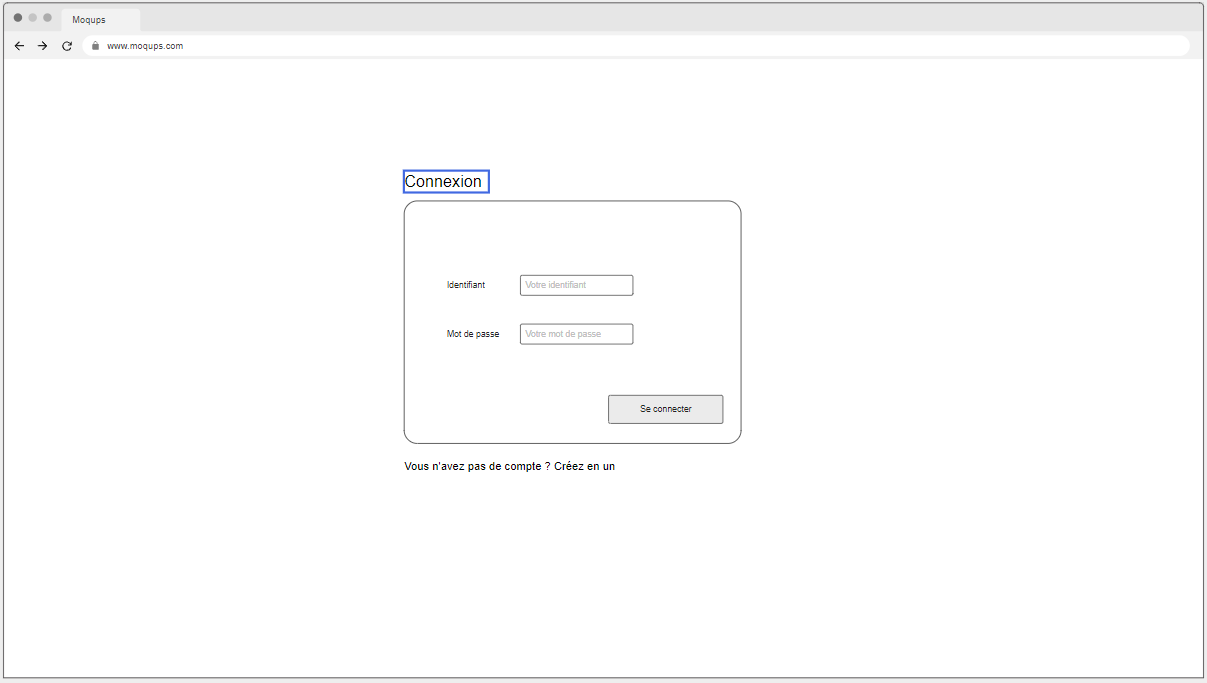
**Page d'accueil** : Une fois le compte créé ou la connexion effectué l’utilisateur sera redirigé vers la page d'accueil où une proposition de séries sera faites lors de la première connexion la proposition ne pourra pas être basé sur les goûts de l’utilisateur une proposition de séries aléatoires sera donc faites puis une proposition de séries sera faites en fonction des séries regardé, la proposition sera basé sur la dernière série regardé.

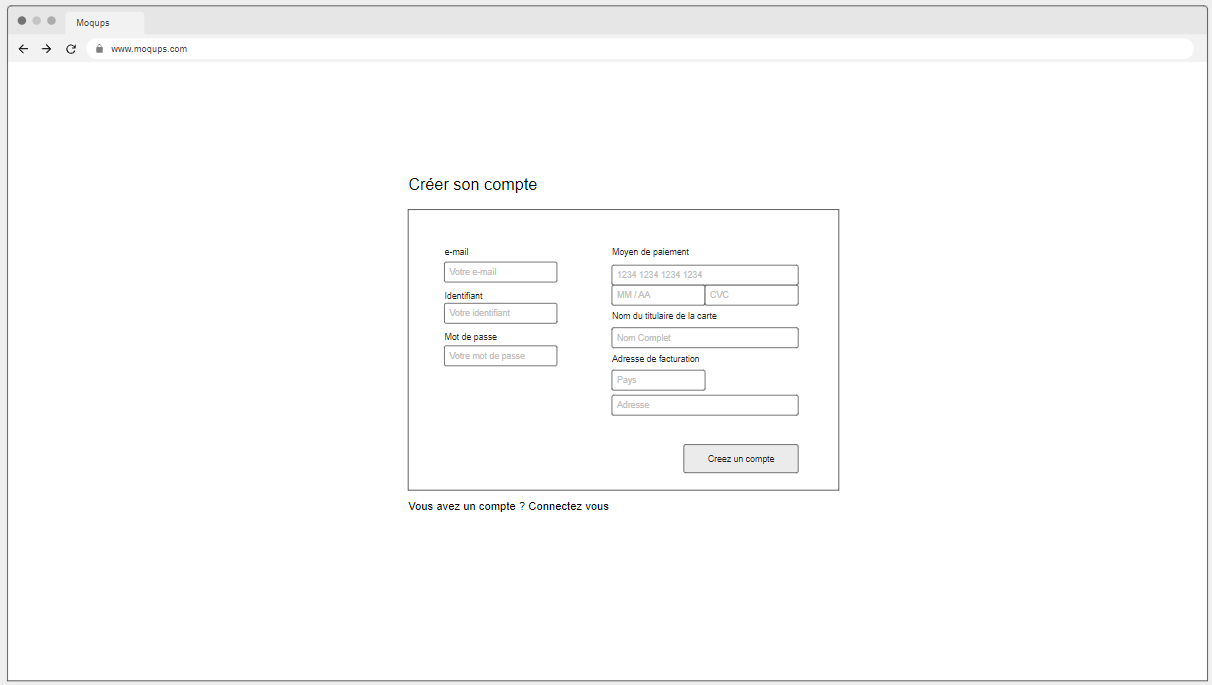
Sur cette page d'accueil il sera aussi possible de chercher une série dans une barre de recherche. Dans un premier temps, les mots écrits par l’utilisateur permettront de chercher le titre d’une série si la recherche par titre ne renvoie rien, une seconde recherche sera effectuée par mot clés, cette recherche renvoie toutes les séries contenant les mots entrés par l’utilisateur par ordre décroissant.

# 

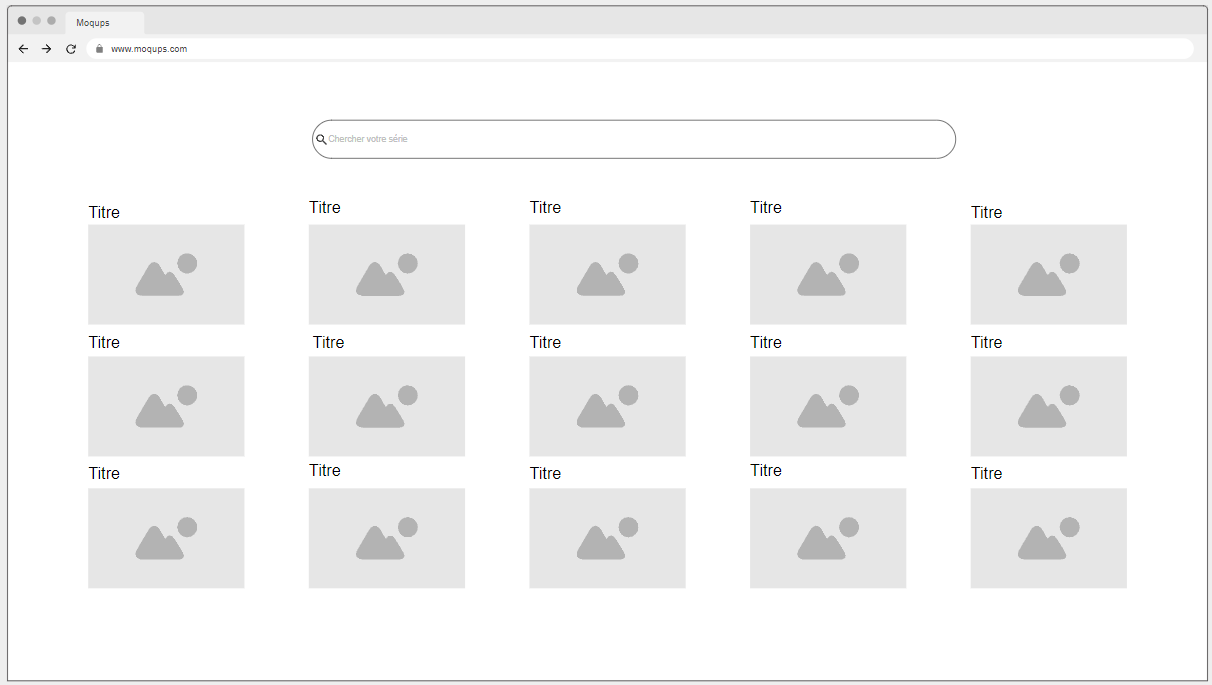
# Description des interfaces et maquettes d’écran

## Page de connexion

Création compte utilisateur



## Accueil

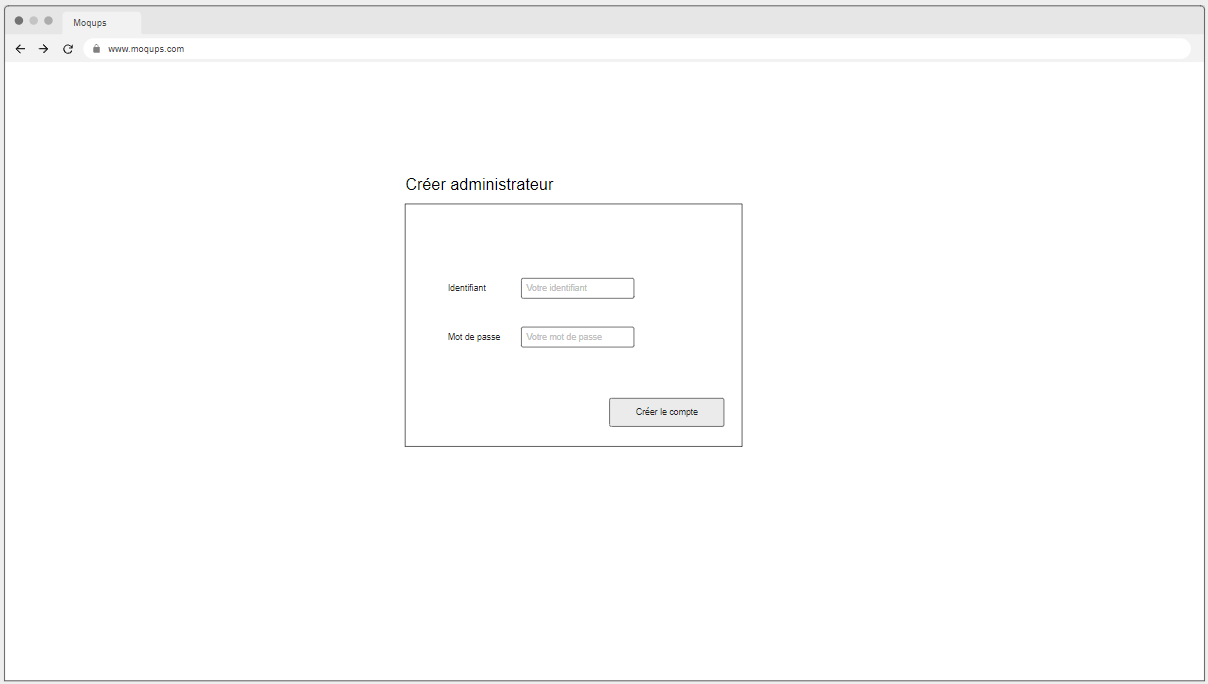


## Gestion administrateur

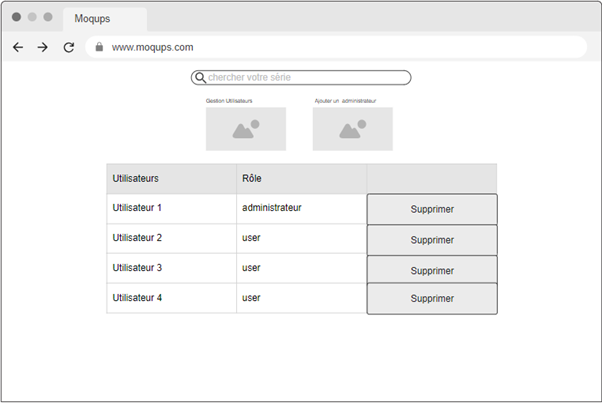
Une image contenant texte, capture d’écran, conception

Description générée automatiquement

## Créer compte administrateur



## Gestion utilisateur



# Modèles de données

## Schémas MongoDB

### Collection Series\_Mots

{

$jsonSchema: {

bsonType: 'object',

title: 'Series\_Mots',

required: ['nom','mots'],

properties: {

nom : { bsonType: 'string'},

mots : {

bsonType: 'array',

items : {

bsonType: 'object',

required: [ 'mot', 'poid'],

properties: {

mot : {bsonType: 'string' },

poid: {bsonType: 'double'}

}

}

}

}

}

}

### Description :

Cette collection permet de stocker :   
- Le nom de la série (“nom”)  
- La liste des mots lemmatisé à partir des fichiers sous-titres (“mots”). Chaque élément de la liste est composé d’un mot (“mot”) et de son poids (“poids”) calculé par l’algorithme TF-IDF

Les champs nom (référence au titre de la série) et mots sont obligatoires car ils sont nécessaires au référencement de la série pour nos fonctions de recherche et de recommandation.

### Collection Utilisateur

{$jsonSchema: {

bsonType: 'object',

title: 'Utilisateur',

required: [

'identifiant',

'mdp',

'role'

],

properties: {

identifiant: {

bsonType: 'string'

},

mdp: {

bsonType: 'string'

},

role: {

bsonType: 'array',

'enum': [

'administration',

'utilisateur'

]

},

date\_exp: {

bsonType: 'string',

pattern : '^[0-9]{2}\/[0-9]{2}$'

},

email: {

bsonType: 'string',

pattern : '^[0-9]{3}$'

},

cvc: {

bsonType: 'string',

pattern : '^[0-9]{2}\/[0-9]{2}$'

},

num\_carte\_bank: {

bsonType: 'string',

pattern : '^[0-9]{16}$'

},

adresse: {

bsonType: 'string'

},

pays: {

bsonType: 'string'

}

}

}}

### Description :

Cette collection permet de stocker :   
- L’identifiant d’un utilisateur (“identifiant’)

- Son mot de passe (“mdp”) qui sera sous sa forme hashé

- Son rôle (“rôle”), s’il est un administrateur ou un utilisateur

Si le compte correspond à un utilisateur alors nous stockerons :   
- Son email

- Son adresse

- Son pays de résidence

- Le numéro carte bancaire

- La date d'expiration de sa carte bancaire

- Son CVC

À noter que l’identifiant utilisateur doit-être unique.

### Collection Mots\_Series

{

$jsonSchema: {

bsonType: 'object',

title: ‘Mots\_Series’,

required: ['mot', 'series'],

properties: {

mot: {

bsonType: 'string'

},

series: {

bsonType: 'array',

items: {

bsonType: 'object',

required: ['serie', 'TF-IDF'],

properties: {

serie: {

bsonType: 'string'

},

'TF-IDF': {

bsonType: 'double'

}

}

}

}

}

}

}

### Description :

Cette collection permet de stocker :   
- Un mot (“mot”)  
- La liste des séries à partir des fichiers sous-titres (“series”). Chaque élément de la liste est composé d’une série (“série”) et de du poid du mot dans la série (“poid”) calculé par l’algorithme TF-IDF

Nous avons fait le choix d’incrémenter chaque mot avec des séries dans le but de limiter le nombre de données insérées.

## Informations supplémentaires

Nous avons décidé d’utiliser une base de données orientée NOSQL avec MongoDB car nous traitons une quantité importante d’informations notamment par notre système de recherche et de recommandation. Nous cherchons à avoir les meilleures performances possibles lors du traitement d’informations. Les bases de données NOSQL nous semblaient donc être la meilleure solution.

Notre base de données comportera trois collections MongoDB. La première collection comporte toutes les informations relatives à un utilisateur. Notre deuxième collection comportera toutes les informations relatives à une série. Et la dernière comporte les informations du TF-IDF relatif à chaque mot.