Université de Rouen, UFR Sciences et Techniques
Master 1 IGIS – Langages Web
SciMS, un mini CMS pour les articles scientifiques

Boubekri, Abdelmalek abdelmalek.boubekri@etu.univ-rouen.fr
Rabhi, Lounis lounis.rabhi@etu.univ-rouen.fr

24 décembre 2016

Table des matières

Liste des symboles

AJAX Asynchronous JavaScript And Xml
CRUD Create Read Update and Delete
CSS Cascading Style Sheets
DOM Document Object Model
HTML HyperText Markup Language
MVC Model View Controller
MVT Model View Template
ORM Object-Relational Mappers
SGBD description
URL Uniform Resource Locator

Chapitre 1

Spécification et analyse du besoin

1.1 Objet

Dans le cadre du mini-projet de langages web, il nous a été demandé de créer un mini CMS pour la rédaction d'articles scientifiques, qui auront un rendu similaire à LaTeX.

1.2 Défintion des utilisateurs

Administrateur peut seulement visualiser le contenu des catégories et des articles.

Rédacteur peut créer , modifier ou supprimer les articles dont il est propriétaire.

Anonyme peut ajouter, supprimer, modifier des utilisateurs et des catégories.

1.3 Cas d'utilisation

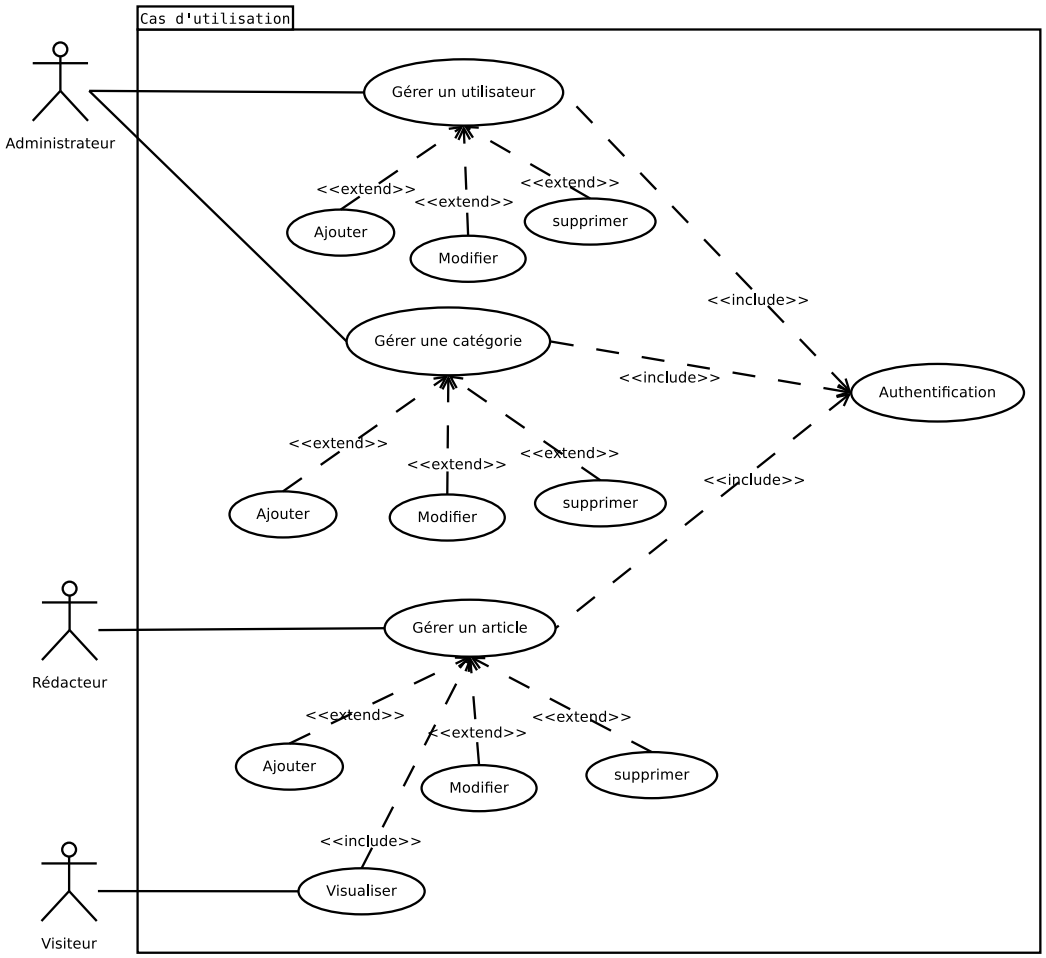

FIG. 1.1 – *Diagramme de cas d'utilisation*

Chapitre 2

Technologies utilisées

2.1 Django

Django est un framework qui s'inspire du principe MVT (la vue est gérée par un template) composé de trois parties distinctes :

- Un langage de templates flexible qui permet de générer du HTML, XML ou tout autre format texte ;
- Un contrôleur fourni sous la forme d'un « remapping » d'URL à base d'expressions régulières ;
- Une API d'accès aux données est automatiquement générée par le framework compatible CRUD. Inutile d'écrire des requêtes SQL associées à des formulaires, les requêtes SQL sont générées automatiquement par l'ORM.

Le modèle MVT est tout à fait équivalent au modèle MVC, tel que la **View** a été remplacée par le **Template** et le **Controller** est remplacé par la **View**.

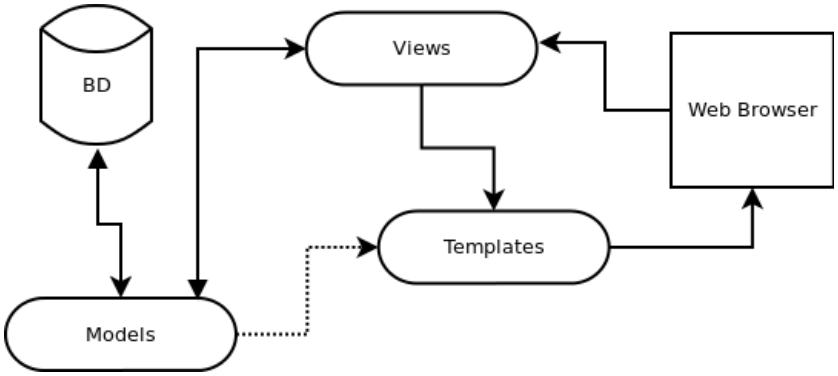

FIG. 2.1 – *Modèle MVT*

2.2 choix technique

2.2.1 Langage pyhton

langage puissant fournissant une multitude d'outils et de bibliothèques et omniprésent dans la sécurité.

2.2.2 Framework Django

Le choix s'est porté sur le framework Django, pour les raisons suivantes:

- Contraignant à développer en utilisant les bonnes pratiques telles que le modèle MVT;
- gagné en popularité ces dernières années;
- Améliorer ses compétences en python pour le web, qu'on n'a pas vu cette technologie en cours;
- Un serveur «causant» qui facilite le débogage, permet de monitorer les requêtes HTTP et les sorties système (print);
- Idéal pour un projet collaboratif;
- Compatible avec la plupart des technologies.

Django utilise également **SQLite** comme base de données, qui a l'avantage d'être non volumineuse, et ne nécessite pas la mise en place un serveur.

2.2.3 Bootstrap

- Il est responsive;
- Fonctionne sur «tous» les navigateurs;
- Embarque un arsenal de composants prêts à l'usage;
- Possède également des composants JavaScript

2.3 Architecture de l'application

Au lieu de mettre une structure article/section/sous-section/paragraphe dans la base de données, on a plutôt opté pour la même philosophie que LaTeX, c'est à dire pas de structure préalable aux documents,Par exemple: on peut avoir une sous-section (0.1) sans section mère, ou une figure sans conteneur. Cela permet une meilleure flexibilité et liberté et par conséquent avoir une meilleure expérience utilisateur lors de la rédaction d'articles.

2.3.1 Structure de l'application SciMS

SciMS/ contient les réglages et les URLs du site web de la racine ainsi que le seveur WSGI.

articles/ contient les différents composants de application et le dossier static contenant trois sous dossiers :

- js/ contient les fichiers javascripts qu'on a développé ainsi que les fichiers utilisés par bootstrap.
- css/ contient notre fichier de style et la bibliothèque Bootstrap.
- images/ contient les images.

templates / contient les templates propres à l'application.

Chapitre 3

Développement

3.1 Les modèles

Les modèles utilisés sont:

```
class Category(models.Model):
 cat_name = models.CharField(max_length=128)

class Keyword(models.Model):
 word = models.CharField(max_length=20)

class Article(models.Model):
 pub_date = models.DateTimeField(auto_now_add=True)
 title = models.CharField(max_length=200)
 abstract = models.TextField()
 content = models.TextField()
 category = models.ForeignKey(Category, on_delete=models.
    CASCADE)
 author = models.ForeignKey(User, on_delete=models.CASCADE)
 keyword = models.ManyToManyField(Keyword)
```

3.2 Les Expressions régulières

rxp.py regroupant les expression régulières permettant d'embarquer les formules, sections sous-section et figures:

```
import re

def toHTML(str):
 section = re.compile(r"^\\(section)\{([a-zA-Z0-9_ ]*)\}", re
    .MULTILINE)
```

```
subsection = re.compile(r"^\\(subsection)\{([a-zA-Z0-9_ ]*)\}", re.MULTILINE)
mathform =re.compile(r"(\[.*\])", re.MULTILINE)
tofcontents =re.compile(r"\\(tableofcontents)", re.MULTILINE)
image =re.compile(r"\{%(.*)%\}", re.MULTILINE)

sect = "<h2 class=\"section\">\\2</h2>"
subsect = "<h3 class=\"subsection\">\\2</h3>"
math = "<math class=\"formule\" display=\"block\" xmlns=\"http://www.w3.org/1998/Math/MathML\">
  <mrow>
    <mi>d</mi>
    <mo>=</mo>2
  </mrow>
</math>"
contents = "<div id=\"tofcontents\"><h4>Table des matieres</h4></div>"
img="<img src=\"\\1\">"

str = section.sub(sect, str)
str = subsection.sub(subsect, str)
str = mathform.sub(math, str)
str = image.sub(img, str)
return tofcontents.sub(contents,str)
```

3.3 La génération de la table des matières

Le script Javascript permettant de générer la table des matières:

```
function sumGen(){
  var article = document.getElementById("artContent");
  var summary = document.getElementById("tofcontents");
  if(summary != null){
  cpt = 1;

  mynav = document.createElement("nav");
  mynav.appendChild(genTheSum(article));
  summary.appendChild(mynav);
  }
}

function genTheSum(myarticle){
  list = document.createElement("ul");
  for (var i = 0; i < myarticle.childNodes.length; i++){
    if (myarticle.childNodes[i].tagName == "H2"){
    var li = document.createElement("li");
```

```
18        var anch = document.createElement("a");
19
20        anch.setAttribute("href", "#"+cpt);
21        anch.appendChild(document.createTextNode(myarticle.
              childNodes[i].textContent));
22
23        li.appendChild(anch);
24        list.appendChild(li);
25        myarticle.childNodes[i].setAttribute("id", cpt);
26        cpt++;
27        }
28      }
29      return list;
30    }
31    window.onload = sumGen;
```

3.4 Captures d'écran

3.4.1 Page d'accueil

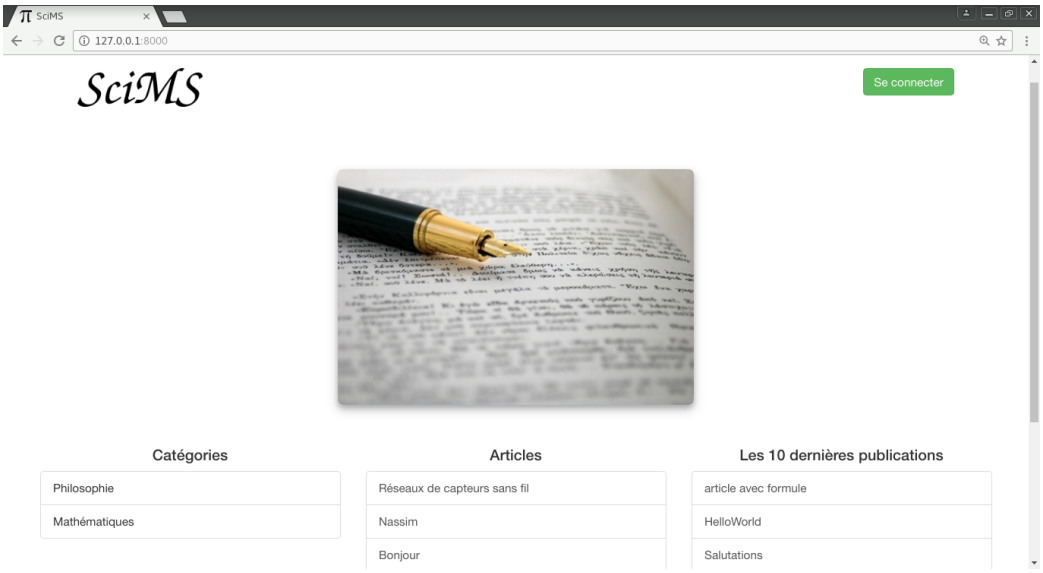

FIG. 3.1 – *Page d'accueil*

3.4.2 Page d'administration

Interface fournie par le framework, accessible pour administrateur.

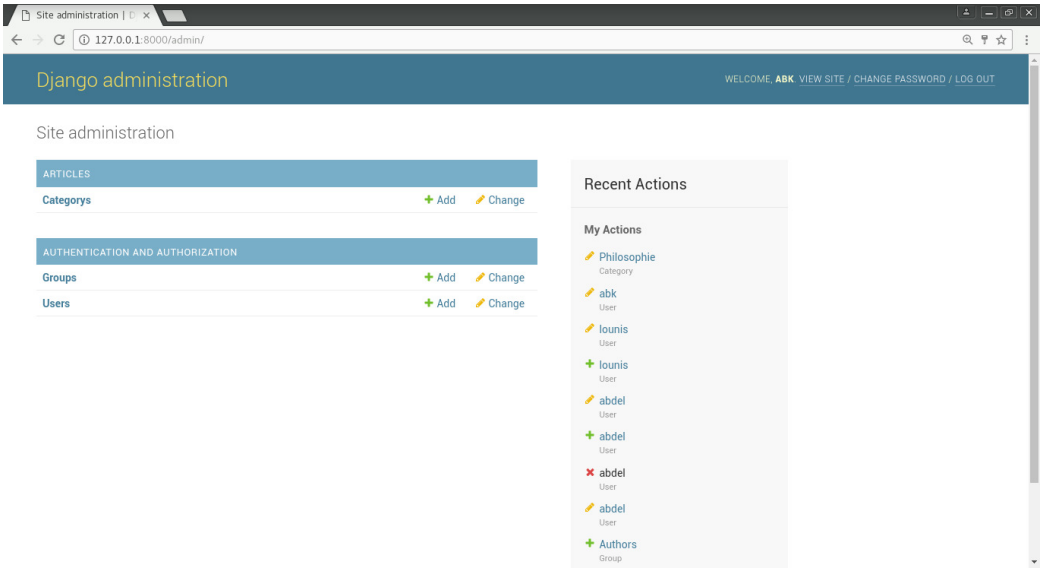

FIG. 3.2 – *Page d'administration*

3.4.3 Page de rédaction

FIG. 3.3 – *Page de rédaction*

3.4.4 Rendu d'un article

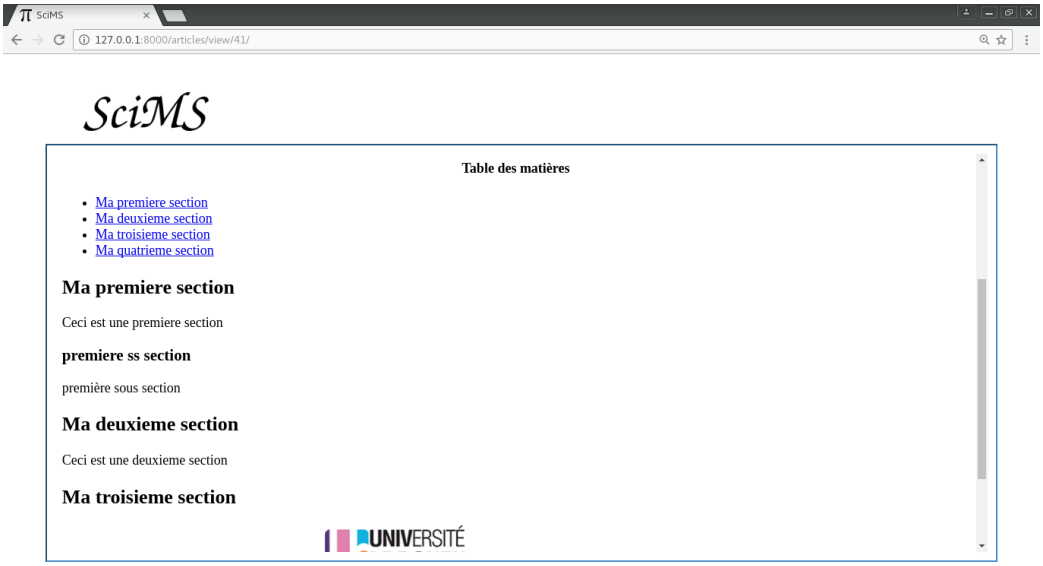

FIG. 3.4 – *Aperçu d'un article pour un anonyme*

3.5 Fonctionnalités implémentées

- Remplit toutes les fonctionnalités de base (ajout, modification et suppression) des articles, utilisateurs et catégories.
- Site web responsive.
- Régule les accès selon le type d'utilisateur.
- Génération des tables des matières aux articles.
- Mise à disposition d'un assistant pour la rédaction.

3.6 Fonctionnalités non-implémentées

- -La disposition de la page d'accueil ne réponds pas aux exigences (barre latérale gauche).
- Ne permets pas le téléchargement d'images sur le serveur, cependant l'affichage est possible en introduisant l'URL d'une image.
- Numérotation des formules.

Chapitre 4

Manuel technique

Dépendances requises:
Python >=3.4
Python3-pip
Django >=1.10

4.1 Installation

1- Installer les dépendances

```
# apt-get install python3
# apt-get install python3-pip
# pip3 install django
```

2- Télécharger l'archive FileX
3- Extraire l'archive, par défaut le chemin le chemin des templates a été défini dans SciMS/settings.py est /var/www/SciMS/templates/ , il faut le modifier par le chemin correspondant au votre.

4.2 Utilisation

Lancer le serveur avec la commande:

```
$ python3 manage.py runserver
```

Vous pouvez accéder à l'application via un navigateur;
Un compte administrateur est mis à disposition, identifiants: **abk:malekbbkr**
deux comptes rédacteur, identifiants: **abdel:django123** et **farouk:far123456**