

# Projet Applications Web

## Rapport Final

Kamal Hammi  
Salahdine Ouhammiali  
Mohamed Hamza Kadri  
Mahmoud Laanaiya

Département Sciences du Numérique - Deuxième année  
2021-2022

## 1 Theme de site

le site permet à une entreprise de gérer ses annonces et ses publications sur facebook. Il y a deux façons d'utiliser le site la première est de de l'utiliser comme un utilisateur normal et la deuxième et de l'utiliser comme un manager. Un manager s'occupe de poster les publications d'une campagne de l'entreprise, et de faire un relevé de reaction sur ces publications (nombre de likes, commentaires etc.) afin de décider les groupes qui réagissent le plus et les campagnes qui intéressent les clients. Pour ce qui est de l'utilisateur il peut à travers notre application se connecter, s'inscrire et afficher plusieurs informations.

## 2 Fonctionnalités

Le site permet à l'utilisateur et au manager de réaliser plusieurs opérations, dans cette partie nous allons décrire une à une ses opérations.

### 2.1 Fonctionnalités Utilisateur

Un utilisateur peut à travers le site :

- Se connecter, ou s'inscrire s'il n'est pas encore enregistré.
- Voir les campagnes, les posts ou les groupes.
- Afficher le relevé des réactions d'une campagne.

### 2.2 Fonctionnalités Manager

Un manager (en plus des permissions données à l'utilisateur) peut grâce au site :

- Créer une nouvelle campagne.
- Ajouter les informations d'un nouveau groupe.
- Créer un nouveau post.
- Ajouter un autre manager.
- Associer un manager à un groupe.
- Associer un post à une campagne.

## 3 Description Technique

### 3.1 Entities

- Dans ce projet nous avons eu besoin d'implémenter plusieurs entités pour effectuer les opérations dont nous avons besoin pour atteindre le résultat voulu. Nous allons les décrire une par une dans les parties qui suivent.

#### 3.1.1 Campagne

- **les attributs de cette entité :**

- id\_campagne (@Id integer généré automatiquement), nom (string), date de debut (string), date de fin (string), objectifs (string), contexte (string), posts (collection Post OneToMany , mapped by campagne)  
on a géré ici le fait qu'une campagne peut contenir plusieurs posts.  
- groupes (collection Groupe ManyToMany) ici on a supposé qu'une campagne a plusieurs groupes et qu'un groupe lui aussi a plusieurs campagnes.

#### 3.1.2 Groupe

- **les attributs de cette entité :**

- id\_groupe (@Id integer généré automatiquement), nom (string), nombre de membres (integer), theme (string), type (string), reglementation (string), url de groupe (string), frequence des posts (integer), format du groupe (string), lieu (string), campagnes (collection Campagne ManyToMany, mapped by groupes),  
ici on gère le fait qu'un groupe peut avoir plusieurs campagnes,(voir groupes dans Campagne.java).  
- posts (collection Post OneToMany , mapped by groupe), en suivant le même raisonnement on décrit la relation entre Post et Groupe.  
- managers (collection Manager ManyToMany), relation entre Manager(entity) et Groupe.

#### 3.1.3 Post

- **les attributs de cette entité :**

- id\_post (@Id integer généré automatiquement), titre (string), url (string), contenu (string), campagne (Campagne), utilisateur (Utilisateur), groupe (Groupe),releves (collection Releve\_post OneToMany, mapped by post)

#### 3.1.4 Relevepost

- **les attributs de cette entité :**

- id\_releve (@Id integer généré automatiquement), date de relevé (string), nombre de likes (integer), nombre de commentaires (integer), nombre de shares (integer), post (Post)

#### 3.1.5 Utilisateur

- **les attributs de cette entité :**

- id\_utilisateur(@Id integer généré automatiquement), nom (String), prenom (String), login (String), mot\_de\_passe (String)

#### 3.1.6 Manager

- **les attributs de cette entité :**

- login\_Facebook (string), mdp\_Facebook (string), groupes (collection Groupe ManyToMany)  
On remarque ici moins d'attribus car Manager hérite d'Utilisateur.

### 3.2 Facade

- Dans cette partie nous allons décrire les principales opérations que la façade doit implémenter :
- void ajouterCampagne (string nom, string date\_debut, string date\_fin, string objectifs, string contexte).
- void ajouterGroupe (string nom (string), et d'autres opérations d'ajouter pour les autres entités.
- On a aussi des fonctions qui permettent de lister, par exemple lister les groupes, les campagnes, etc ..
- On a ajouté aussi des méthodes get() (getters) pour récupérer des éléments , et des opérations d'association entre différents éléments.
- On peut noter aussi qu'on a utilisé un EntityManager em pour manipuler la base de données et gérer les résultats des différentes méthodes et fonctions.

### 3.3 Contrôler

- Dans cette classe (WebServlet) nous avons géré les différentes requêtes et réponses qu'on peut recevoir en utilisant le site.

### 3.4 Schéma de données

- Nous allons donner quelques explications sur les différentes vues que nous avons créé.
- Il ya des .jsp et des .html selon ce qu'on veut représenter, les détails des vues sont dans le dossier WebContent.
- Dans la suite vous trouverez plusieurs extraits des pages de notre site. Vous trouverez les captures d'écrans des pages du site à la fin du document ainsi que le schéma de données à la fin.

## 4 Conclusion

- Nous avons trouvé ce projet très intéressant car il nous a poussé à chercher et à essayer d'améliorer les fonctionnalités, les vues, les styles des pages. Nous avons d'autres fonctionnalités en tête que nous voulions ajouter mais nous n'avons pas eu le temps de le finir mais globalement nous sommes satisfaits des résultats.

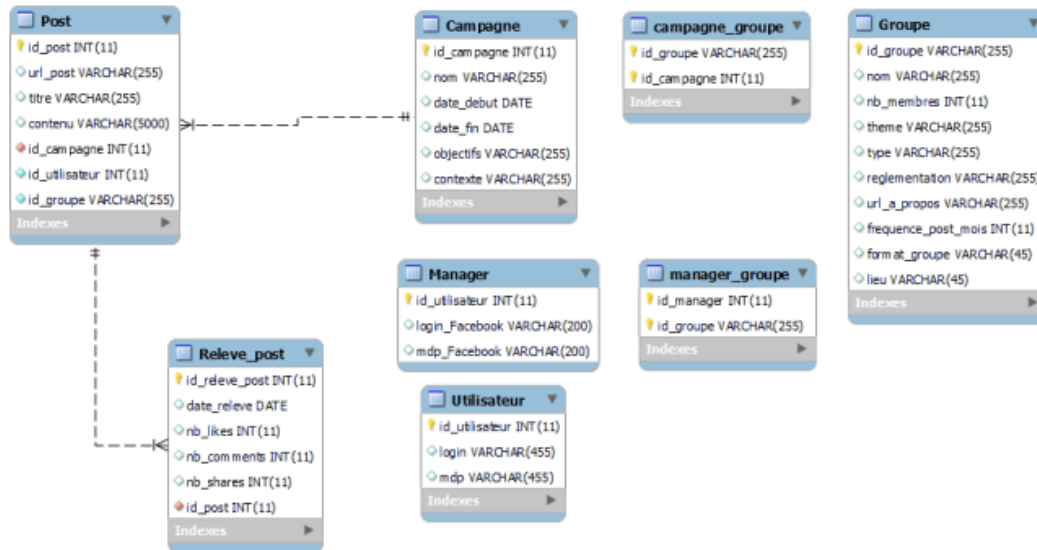


FIGURE 1 – Schéma de données du site