

L2 – Bases de données

# Projet Sortir à Saint- Georges des Groseillers

Phase 3 (finale)

EL JIDE YASSINE - ANASS OUSAID

Groupe TP 12



## Sommaire

---

### TABLE DES MATIÈRES

---

Manuel d'utilisateur.....	3
Documentation technique.....	5
Architecture du Site Web et Structure de la base de données .....	5
Requête SQL majeures .....	7
Gestion de sessions .....	8
Répartition du travail .....	8

# Manuel d'utilisateur

Le site SGG City est un site de la ville de Saint Georges des Groseillers qui offre à ses habitants la possibilité de partager des bons souvenirs des événements à qui ils ont déjà assisté où été invité et de plus il leur donne la possibilité de proposer leurs propres idées des événements.

Lors de la connexion de l'habitant au site celui-ci est dirigé vers la page d'accueil où il trouvera toutes les photos des 5 événements récents des habitants de la ville avec une possibilité de voir les détails de chaque événement et de plus de ça il aura accès aux événements proches qui se dérouleront le mois prochain avec possibilité de voir les détails des événements.

Pour accéder aux détails de chaque événement soit récent soit proches et d'autre fonctionnalités du sites Web comme (la recherche, suggestions, profil, créer un événement) l'utilisateur doit avoir un compte sur le site.

Pour cela dans la navigation, Il existe deux boutons un pour l'inscription et l'autre pour la connexion.

Quand l'utilisateur remplit tous les champs nécessaires sans erreur pour l'inscription ce dernier est redirigé vers la page de connexion pour qu'il se connecte avec son nouveau compte et tant que l'utilisateur ne rentre pas les informations correctement, on lui affiche un message d'erreur.

Au moment de la connexion de l'habitant ce dernier a accès à toutes les fonctionnalités du sites Web :

- **Page de détails de chaque événement soit récent soit proche :** Pour chaque événement situé dans la page d'accueil l'habitant pourra voir ses détails qui contiendra les informations suivantes : le lieu, la date de début et fin, la limite d'âge et sa description ainsi que les photos, les notes et les commentaires des habitants ayant déjà participé à cet événement, **au cas où** l'événement n'a pas encore eu lieu ; y aura un bouton pour l'inscription à cet événement.
- **Page d'inscription de chaque événement :** Après avoir décider de s'inscrire à un événement l'habitant peut choisir une date spécifique qui est entre la date de début et fin de l'événement ; éventuellement l'utilisateur peut s'inscrire au même événement à plusieurs dates.
- **Page de recherche :** Ici les habitants peuvent chercher des événements selon les catégories de ces derniers ou bien leurs tags, ainsi ils auront les événements dédiés avec chacun une possibilité de voir ses propres détails.
- **Page de création d'un événement :** Au cas où l'utilisateur souhaite créer son propre événement il pourra faire en se dirigeant vers cette page en cliquant sur le bouton créer un événement ainsi il aura un formulaire à remplir avec toutes les informations nécessaires.



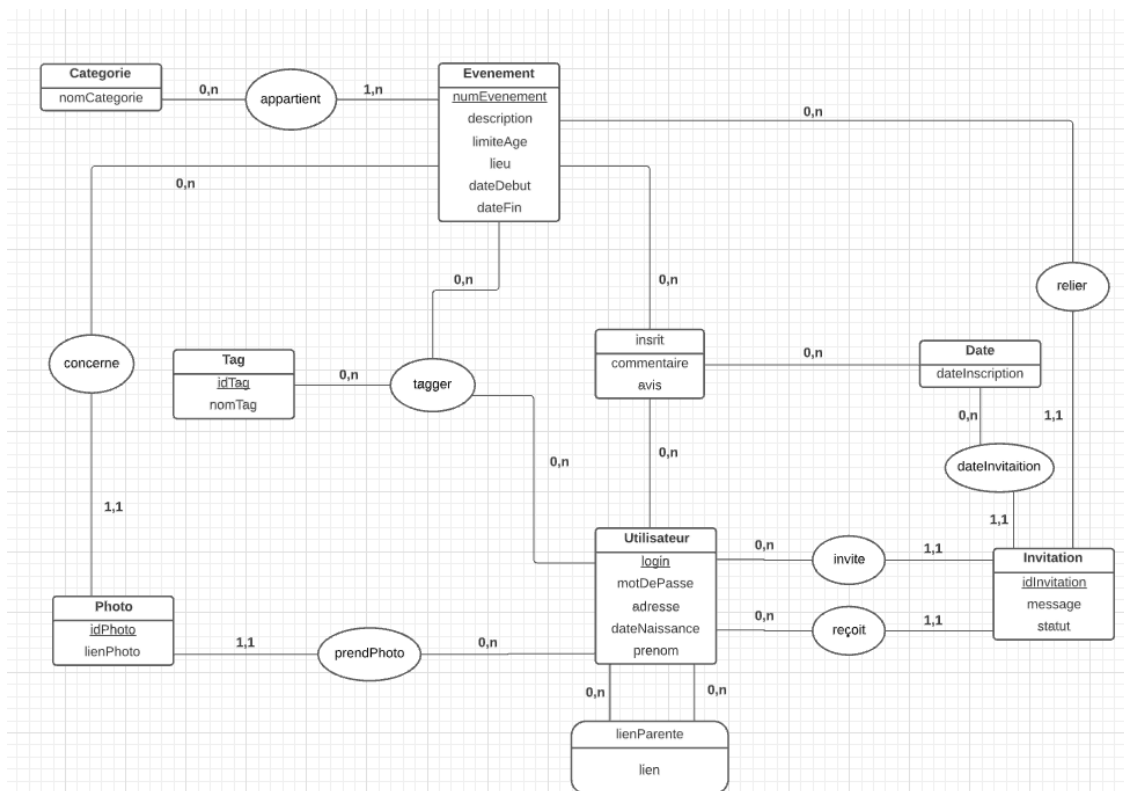
- **Page du profil :** En se connectant au site Web, l'habitant aura accès à sa page du profil, ici il pourra ajouter une note, un commentaire et une photo pour chaque événement où il a déjà participé, ainsi il verra tous les événements où il est inscrit avec une possibilité d'inviter des gens à ce dernier en cliquant sur un bouton Invitez vos proches, éventuellement y aura un pop-up demandera à l'utilisateur le destinataire et le message de l'invitation ; quand l'invitation est envoyée l'utilisateur peut consulter toutes les invitations envoyées avec chacune son statut (En cours – Accepté – Refusé) et les invitations reçues d'autre part avec la possibilité de l'accepter ou refuser bien sûr.
- **Page des suggestions des événements :** Cette page est dédiée à donner des suggestions personnalisées à chaque habitant de la ville en se basant sur les événements où il est déjà inscrit et qui a bien aimé.

# Architecture du site Web et Structure de la base de données.

La page d'accueil est le point d'accès au site Web ainsi que l'utilisateur a droit à toutes les autres fonctionnalités d'autres pages quand il est connecté. Cela est fait avec les sessions.

## Base de données :

### Schéma E-A :



En ce qui concerne le schéma entité-association, on avait rien modifié pareil pour le schéma relationnel y avait pas de modifications ajoutées.



## Schéma Relationnel :

**Evenement** (numEvenement, description, debutAge, limiteAge, lieu, dateDebut, dateFin)

**Appartient** (numEvenement, nomCategorie)

FK : numEvenement dans Appartient fait référence à numEvenement dans Evenement  
nomCategorie dans Appartient fait référence à nomCategorie dans Categorie

**Tag** (idTag, nomTag)

**Utilisateur** (login, prenom, motDePasse, adresse, dateNaissance)

**Tagger** (numEvenement, idTag, login)

FK : idTag dans tagger fait référence à idTag dans Tag  
login dans tagger fait référence à login dans Utilisateur  
numEvenement dans tagger fait référence à numEvenement dans Evenement

**Photo** (idPhoto, lienPhoto, numEvenement, login)

FK : numEvenement dans Photo fait référence à numEvenement dans Evenement  
Login dans Photo fait référence à login dans Utilisateur

**Categorie** (nomCategorie)

**Datee** (dateInscription)

**Inscrit** (numEvenement, login, dateInscription, commentaire, avis)

FK : numEvenement dans Inscrit fait référence à numEvenement dans Evenement  
login dans Inscrit fait référence à login dans Utilisateur  
dateInscription dans Inscrit fait référence à dateInscription dans Datee

**LienParente** (utilisateur1, utilisateur2, lien)

FK : utilisateur1 dans lienParente fait référence à login dans Utilisateur  
utilisateur2 dans lienParente fait référence à login dans Utilisateur

**Invitation** (idInvitation, message, statut, loginExpediteur, loginDestinataire, dateInscription, numEvenement)

FK : loginExpediteur dans Invitation fait référence à login dans Utilisateur  
loginDestinataire dans Invitation fait référence à login dans Utilisateur  
dateInscription dans Invitation fait référence à dateInscription dans Datee  
NumEvenement dans Invitation fait référence à numEvenement dans Evenement

# Requête SQL majeures

## Généralités :

Pour les affichages, partout où on avait besoins d'afficher soit des événements avec un spécifique identifiant, ou bien ceux où l'utilisateur est inscrit ou bien afficher les invitations reçues où envoyées les requêtes SQL étaient de la forme « Select .... From .... Where login = '{\$\_SESSION['login']}' » ou bien « Select .... From evenement Where numevenement = {\$numevenement} ».

Partout on avait besoin d'insérer les informations à la base de données par exemple L'inscription d'un utilisateur, l'inscription à un événement spécifique, envoyer des invitations ou bien la création d'un nouvel événement ainsi que la mise à jour des informations de la base par exemple en ce qui concerne les avis et les commentaires les requêtes SQL étaient de la forme « Insert Into .... Values .... Where ... » ou bien « Update .... Set ... Where ... »

## Requêtes SQL majeures :

- **Pour la fonctionnalité de recherche :** La requête SQL était de la forme « SELECT description, numevenement from appartient natural join evenement where lower(nomcategorie) LIKE '%{\$categorie} % ».
- **Pour la fonctionnalité de suggestions :** Dans cette requête on se basait sur les événements où l'utilisateur est déjà inscrit et auxquels il a donné un avis supérieur ou égal à 4 et ce choix est fait selon les catégories et les tags aussi la requête était de la forme :

```
SELECT DISTINCT numevenement, description
FROM (
    SELECT numevenement, description
    FROM appartient NATURAL JOIN inscrit NATURAL JOIN evenement
    WHERE nomcategorie IN (
        SELECT nomcategorie
        FROM appartient NATURAL JOIN inscrit
        WHERE inscrit.login = '{$_SESSION['login']}'
        AND avis >= 4
        AND datefin > '{$now}'
    )
) AS P1 UNION (
    SELECT numevenement, description
    FROM tag NATURAL JOIN inscrit NATURAL JOIN tagger NATURAL JOIN evenement
    WHERE tag IN (
        SELECT tag
        FROM tagger NATURAL JOIN tag NATURAL JOIN inscrit
        WHERE inscrit.login = '{$_SESSION['login']}'
        AND avis >= 4
        AND datefin > '{$now}'
    )
)
```

## Gestion de sessions

Une session est démarrée quand l'utilisateur entre sur le site et est conservée entre toutes les pages. Quand l'utilisateur s'authentifie, une variable de session login est initialisée, ce qui permet de savoir que l'utilisateur s'est bien connecté. La plupart des pages vérifient que cette variable de session soit déclarée pour permettre l'accès à la page, et l'utilise pour savoir le login de la personne connectée dans plusieurs requêtes notamment les invitations les suggestions....

Le code faisant cela étant répété partout, il est contenu dans un fichier isConnected.php, et est inclus au début de toutes les pages protégées.

L'ID de sessions est régénéré sur chaque page pour des raisons de sécurité.

L'utilisateur peut également se déconnecter, ce qui aura la double action de supprimer la variable de session pseudo puis de détruire la session, et cette fonctionnalité est implémenté dans le fichier logout.php

## Répartition du travail

En ce qui concerne la répartition du travail, selon les créneaux de disponibilités de chacun on organise des appels discords sinon on se rencontre à la bibliothèque pour partager les idées et les tâches, et travailler ensemble au cas où un de nos deux est bloqué sur un point particulier.





