

Cette Saé a été réalisée par Auxilien Katia, Da Veiga Sanches Carvalho Flavio et Samedi Achok.

Présentation du site fictif

L'association fictive créée se nomme GameAbility ("*Capacité de jeu*"). Elle a pour but de donner l'accès aux jeux-vidéos à tout le monde : et ce, sans exception. En effet, cette association cherche à développer des dispositifs (technologiques...) aidant les personnes handicapées (mentaux, moteur ou cognitif) à jouer aux jeux-vidéos de leurs choix. Les différents projets sont financés via des dons et des revenus d'évènements (Convention, LAN, Tournoi...). L'objectif de l'association a été inspiré d'autres associations qui existent réellement telles que CapGames.

Présentation du site web

Partie Front-end

Concernant la partie HTML et CSS, nous avons repris le rendu d'un devoir maison réalisé avec Monsieur Réda Laroussi que nous avons réajusté pour convenir à la charte graphique de notre association fictive.

La bannière a été créée via l'application [Canva](#), par Katia Auxilien. L'icône affichée sur l'onglet correspond à une icône du site [Flaticon](#) qui propose un grand nombre d'éléments graphiques.

Création de la base de données.

Voici les requêtes SQL envoyées à la base de données pour créer les tables dont nous avons eu besoin :

```
CREATE TABLE Participants(  
    idEvenement int(11),  
    idParticipant int(11),  
    FOREIGN KEY (idEvenement) REFERENCES Evenements_sae(idEvent),  
    FOREIGN KEY (idParticipant) REFERENCES User_sae(idUser),  
    role VARCHAR(20) NOT NULL DEFAULT 'participant' CHECK (role IN ('organisateur', 'regulateur', 'gestionnaire', 'participant'))  
);
```

```
Create Table Evenements_sae(  
    idEvent INT PRIMARY KEY AUTO_INCREMENT,  
    idCreateur INT NOT NULL,  
    FOREIGN KEY (idCreateur) REFERENCES User_sae(idUser),  
    dateEvenement Date NOT NULL,  
    titre VARCHAR(30) NOT NULL,  
    description VARCHAR(2500) NOT NULL,  
    type VARCHAR(30) NOT NULL check (type IN ('LAN', 'convention', 'competition', 'stream'))  
);
```

```
Create Table User_sae(
  idUser INT PRIMARY KEY AUTO_INCREMENT,
  identifiant varchar(30) UNIQUE NOT NULL,
  motdepasse varchar(255) NOT NULL,
  addrmail varchar(320) UNIQUE NOT NULL,
  datedenaissance DATE NOT NULL,
  nom varchar(60) DEFAULT NULL,
  prenom varchar(60) DEFAULT NULL,
  genre VARCHAR(20) NOT NULL CHECK (genre IN ('homme','femme','autre')),
  situation VARCHAR(20) NOT NULL CHECK (situation IN ('handicap','valide','NSPP')),
  handicap VARCHAR(20) DEFAULT NULL CHECK (handicap IN ('moteur','sensoriel','mental','cognitif')),
  roleSite VARCHAR(20) DEFAULT 'utilisateur' NOT NULL CHECK (roleSite IN ('utilisateur','gestionnaire','moderateur')),
  imgUrl varchar(255) DEFAULT NULL
);
```

```
CREATE TABLE Commentaire_sae (
  idCommentaire INT PRIMARY KEY AUTO_INCREMENT,
  idEvenement INT NOT NULL,
  FOREIGN KEY (idEvenement) REFERENCES Evenements_sae(idEvent),
  idUser INT NOT NULL,
  FOREIGN KEY (idUser) REFERENCES User_sae(idUser),
  contenu varchar(2000)
);
```

Nous voulions utiliser CREATE DOMAIN, pour vérifier les valeurs possibles d'un attribut sur *mariadb* mais ce système n'accepte pas ce type de syntaxe. De même pour les clé étrangères dont il a fallu adapter la syntaxe.

Partie Back-end.

Dans un premier temps, nous voulions nous baser sur une partie du système Modèle Vue Contrôleur, ainsi pour la partie connexion et inscription, nous avons mis en place ce système, en revanche par la suite, nous avons rencontré des problèmes que nous n'avons pas réussi à résoudre, c'est pour cela que les parties profil et créer événements ne sont pas basés sur ce système.

Inscription/Connexion:

La première page affichée est celle de connexion. Si aucun compte n'a été créé, il suffit de cliquer sur *Créez-en un* qui nous amène au formulaire d'inscription. Ce formulaire indique des champs obligatoires (marqués par un *). Une fois le formulaire validé et envoyé, une fonction est lancée: celle-ci crée un tuple dans la table *User_sae*. De plus, nous sommes redirigés dans la page de connexion.

Gestion des informations et modification du profil:

La page profil regroupe l'ensemble des informations de l'utilisateur connecté situé dans la table *User_sae* (à l'exception de son *idUser*).

Sur cette page, la photo de profil ,mot de passe, le nom et le prénom sont modifiables. Pour la photo de profil, il faut l'url de l'image voulue. Il faut toutefois rafraichir nous-même afin de voir les données mises à jour.

Affichage et filtrage/tri des événements:

La page événement regroupe l'ensemble des informations de chaque événement situé dans la table *Evenements_sae* (à l'exception de son *idEvent*).

Chaque événement contient sa propre fiche (accessible en cliquant sur le nom de l'événement).

Concernant le tri des événements, les événements peuvent avoir: aucun tri, un tri par date décroissant, un tri par titre croissant ou un tri par créateur croissant. Il se présente sous la forme d'une liste déroulante.

Pour le filtrage, ils peuvent être combinés avec un tri et avec eux même. Les filtrage concernent les types d'événement. Il se présente sous forme de checkbox à cocher (ou non) à votre convenance.

Le fait de cliquer sur le bouton Filtrer permet de rafraîchir la page ainsi que de modifier la requête (select) de sélection des événements selon les contraintes appliquées.

Création d'un événement:

Le bouton permettant d'accéder au formulaire de création d'événement n'est accessible que si et seulement si l'utilisateur connecté est un gestionnaire.

Le formulaire impose l'existence d'un titre d'une date et d'un type d'événement afin de pouvoir en créer un. Une fois créé, on est redirigé vers la page d'affichage des événements (où l'on peut voir celui qu'on vient de créer). L'envoi du formulaire appelle une fonction qui récupère les données du formulaire (de type GET) insère un tuple dans la table *Evenements_sae*.

Commentaire d'un événement:

Le bouton permettant d'accéder à la page de création d'un commentaire n'est accessible que si et seulement si l'utilisateur est connecté. Lorsque l'utilisateur n'est pas connecté, il est toujours possible de voir les commentaires d'un événement en accédant à la fiche d'un événement. Il est possible que la publication d'un commentaire ne fonctionne pas.

Pour vous faciliter l'utilisation du site en tant que gestionnaire, nous vous avons préparé un compte. Voici les informations nécessaires:

- **Identifiant:** Luc_Skywalker
- **Mot de passe:** DartoisL

Diagrammes

Diagramme séquence inscription à un événement du site web

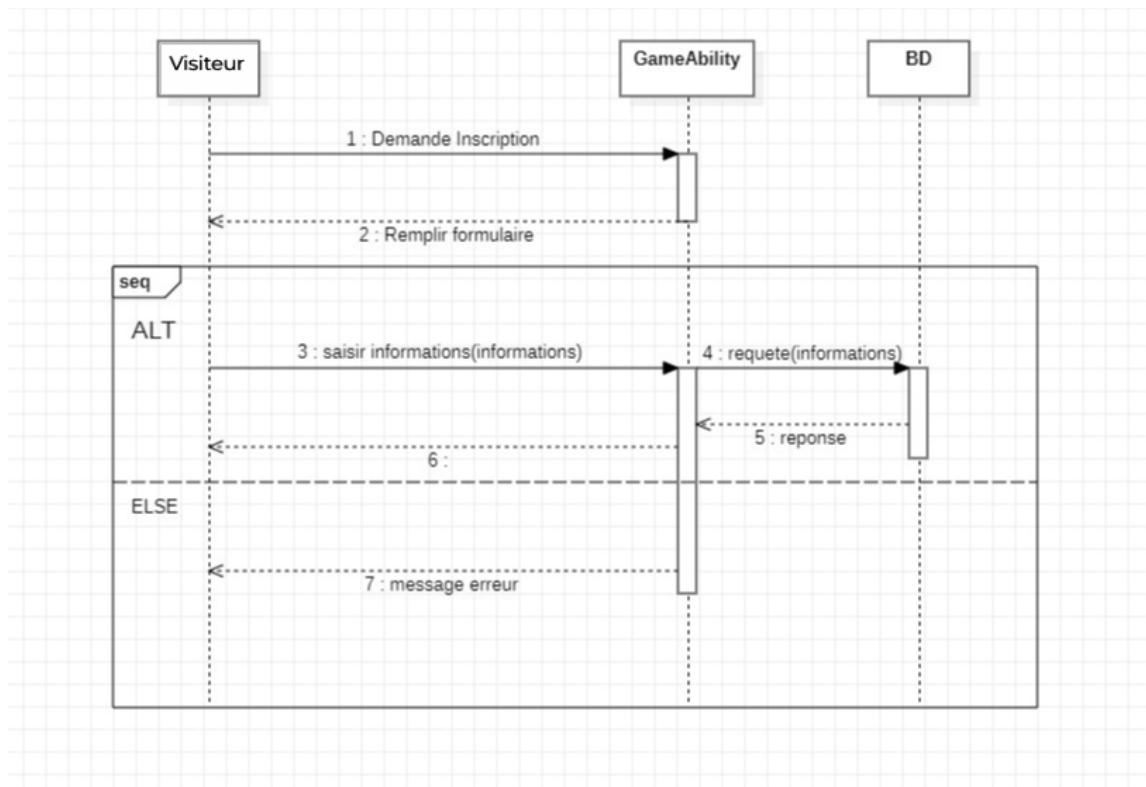


Diagramme séquence inscription au site web GameAbility.

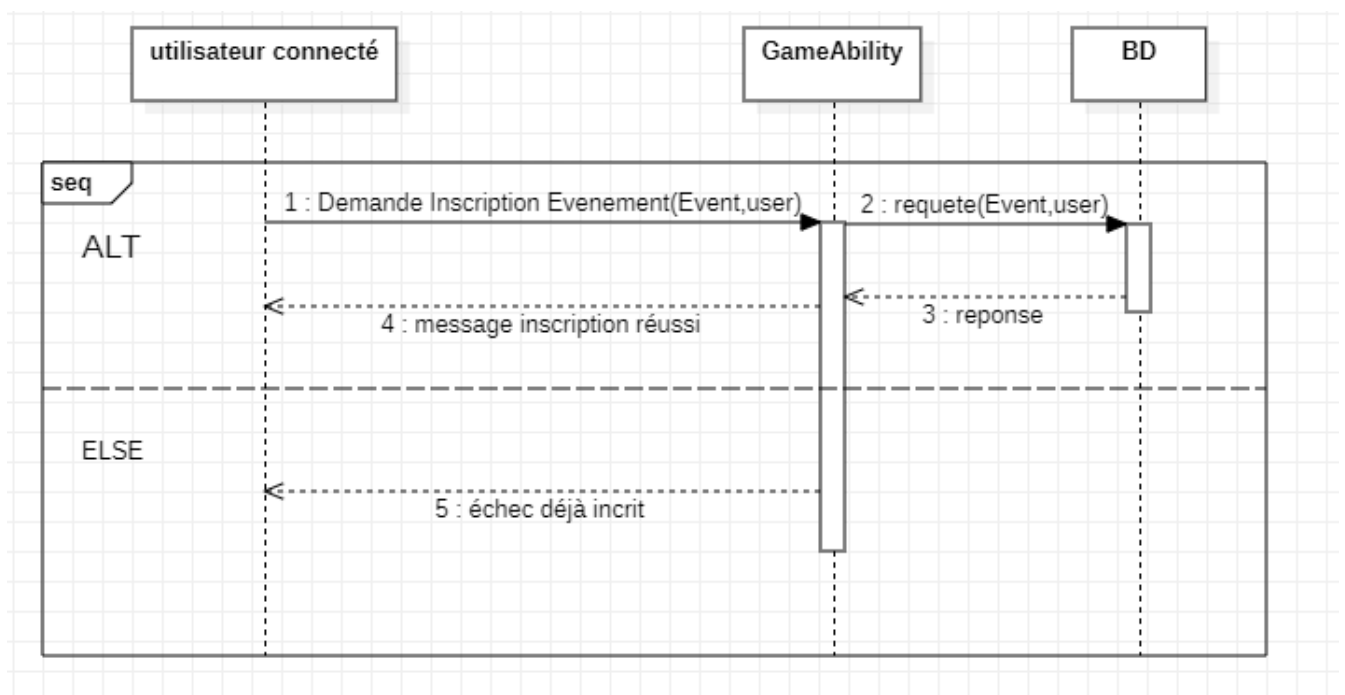


Diagramme de cas d'usage du site web GameAbility.

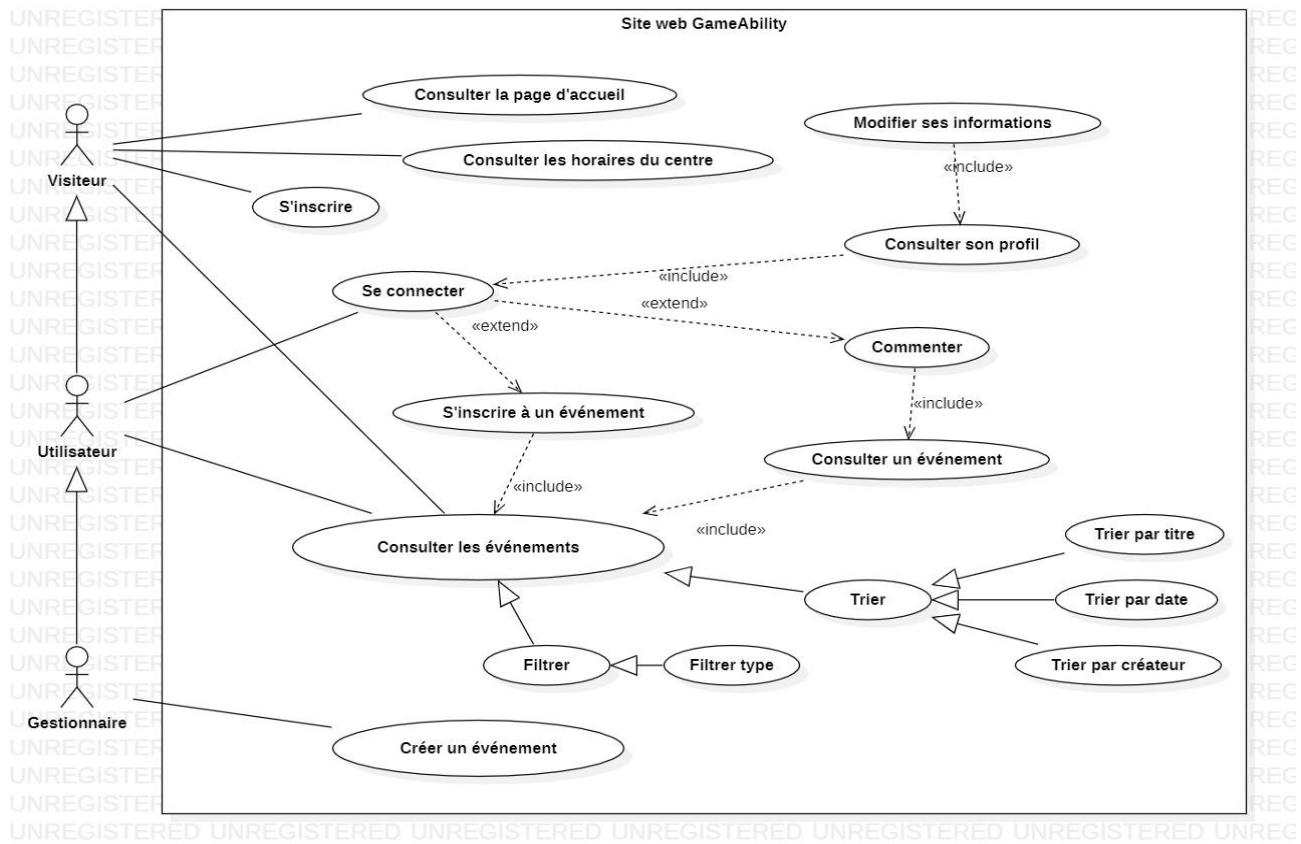
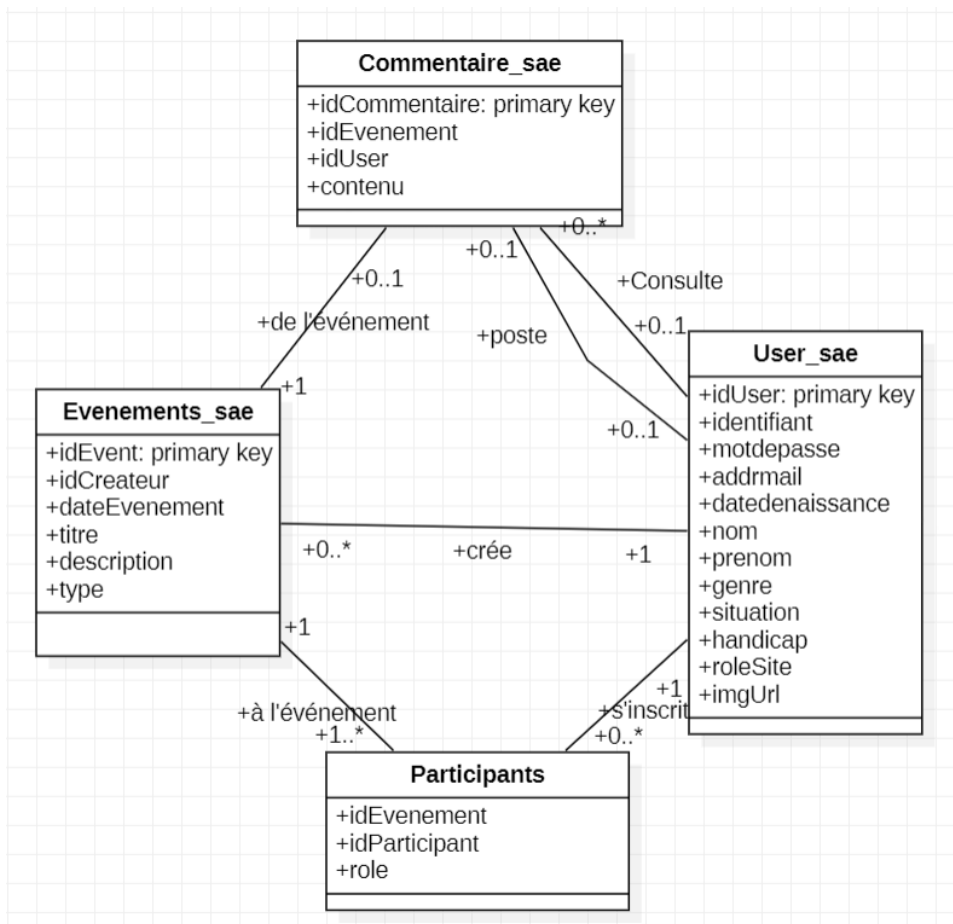


Schéma de base de données



Difficultés rencontrées:

- Modification du profil (gestion des erreurs) Les informations ne s'actualisent pas avec les modifications effectuées, sauf en cas de rafraîchissement.
- Création d'évènement : il manquait une apostrophe, fiche des événements, et l'ajout d'un commentaire depuis une fiche d'événement (ne fonctionnant pas)
- Filtrage: ignorance de l'existence de la fonction *implode()* ainsi que la clause *where 1*
- Communication entre les pages
- Adaptation à la syntaxe de *Mariadb*

Répartition et temps de travail

Katia :

Front-end, Connexion, inscription, profil (modification des informations) évènement (fiche pour chaque événement), commentaire (affichage) pour environ 26 h de travail.

Flavio :

Front-end, Profil (affichage des informations), événements (inscription à un événement,affichage sans tri/filtre), commentaire (création et affichage) pour 24 h de travail.

Achok :

Front-end, événements (création d'un événement), filtre et tri des événements pour 20h de travail.

Voici le lien vers l'URL où le site à été déployé:

<https://dwarves.iut-fbleau.fr/~auxilien/Sae/index.php>

Voici le lien du GIT:

https://dwarves.iut-fbleau.fr/gitiut/samedi/SAe_PHP.git