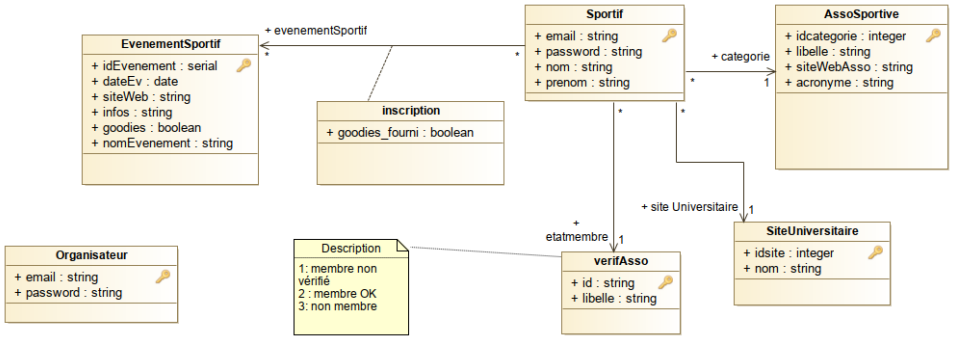
IS 2A - 3ème année - Projet tutoré base de données
Gestion d'événements sportifs - Partie Web

1 Introduction

Durant cette seconde partie du projet, une seule base de données commune 'sportuniv' sera utilisée, conforme au schéma UML suivant : L'attibut `infos` de `EvenementSportif` est un champ texte libre



qui permet de fournir différentes informations (quel est le goodies, où peut-on aller le chercher, etc). Le nom du site universitaire (ou campus) n'a pas été retenu comme identifiant car ce nom change assez régulièrement. Le modèle relationnel est celui-ci :

```
create table evenementsportif (
   idevenement serial primary key, nomevenement varchar(200) not null,
   dateev date not null, siteweb varchar(300) not null,
   infos text, goodies boolean not null default false
 ) ;

create table assosportive (
  idcategorie integer primary key, libelle varchar(200) not null unique,
  acronyme varchar(30) not null unique, sitewebasso varchar(300) not null unique
  ) ;

insert into assosportive  values
  (1, 'Association␣Sportive␣des␣Personnels␣de␣l''Université␣de␣Lille',
   'ASP␣ULille', 'http://www.aspulille.fr'),
  (2, 'Association␣Sportive␣des␣Etudiants␣de␣l''Université␣de␣Lille' ,
   'ASE␣ULille', 'https://ase-ulille.blogspot.com');

create table siteuniversitaire (
  idsite integer primary key, nom varchar(80) not null unique
) ;

insert into siteuniversitaire values
  (1,'Campus␣Santé'),(2,'Campus␣Cité␣Scientifique'), (3,'␣Campus␣Moulins-Ronchin'),
  (4,'Campus␣Flers-Chateau'), (5, 'Campus␣Pont␣de␣Bois'),
  (6, 'Campus␣Roubaix-Tourcoing'),(7,'Campus␣Lille') ;

create table verifasso (
  id integer primary key, libelle varchar(80) not null unique
) ;

insert into verifasso values
 (1,'membre␣non␣vérifié') , (2, 'membre␣OK'), (3,'non␣membre') ;
```

```
create table sportif(
  email varchar(300) primary key, password varchar(10) not null,
  nom varchar(200) not null, prenom varchar(200) not null,
  refcategorie integer not null references assosportive,
  refsiteuniversitaire integer not null references siteuniversitaire,
  etatmembre integer not null references verifasso default 1,
) ;

create table inscription (
  refevenement integer references evenementsportif,
  refsportif varchar(300) references sportif,
  goodies_fourni boolean not null default false,
  primary key (refevenement, refsportif)
) ;

create table organisateur (
  email varchar(300)  primary key, password varchar(10) not null
) ;

insert into organisateur values ('bob@gmail.com', 'secret') ;
```

2 Les fonctionnalités du serveur web

Rappel, le serveur web doit permettre :

à tous :

* d'exposer la liste des prochains événements sportifs avec leurs descriptifs
* la possibilité de s'enregistrer (action préalable pour s'inscrire aux événements)
* la possibilité de s'inscrire à un ou plusieurs événements
* la possibilité à un adhérent de consulter ses inscriptions (et uniquement les siennes), et d'éventuellement de les annuler.
- d'afficher un bilan statistique des inscriptions pour chaque événement : nombre d'étudiants inscrits, nombre de personnels inscrits, puis du nombre total de participants pour l'année civile en cours.

aux seuls organisateurs :

- d'ajouter un événement sportif,
- de consulter les inscriptions des événements,
- de valider ou pas l'appartenance de l'usager à l'une des deux associations
- de valider la fourniture du goodies.

Important : les fonctionnalités étiquetées par * sont requises. Ne faire les autres que si vous les avez terminées.
Le rendu visuel graphique est secondaire, la facilité d'utilisation est à privilégier.

3 Organisation du projet

3.1 Sécurisation de vos pages webs

Pour sécuriser votre espace web (`~/public_html`), lancez la commande suivante dans un terminal :
`cd ;  super set_public_html on`
remarque : avec cette commande, votre groupe Unix (apgis3) ne pourra pas accéder à vos pages, l'option « off » de cette commande permet de revenir en arrière.

3.2 Exemple de page web sécurisée par login/password

Cet exemple montre comment forcer l'internaute à se connecter pour accéder à une page web. Toute page web (html ou page web) que l'on veut sécuriser aura un prologue conforme (ajout des 3 premières lignes) à cet extrait :

```
<?php
    session_start();
    include("./securiteSimple.php");
?>
<!DOCTYPE html>
<html> ...
\end{document}
```

Le code du fichier `securiteSimple.php` est le suivant :

```
1  <?php
2  define('APPLICATION_NAME','Demo Securite');
3  define('TEXT_TRY','Veuillez vous identifier');
4  define('TEXT_FAIL','Identifiants incorrects');
5
6  if(!isset($_SERVER['PHP_AUTH_USER'])){
7    header('WWW-Authenticate: Basic realm="'.APPLICATION_NAME.'"');
8    header('HTTP/1.0 401 Unauthorized');
9    echo TEXT_TRY;
10   exit;
11   }
12
13 $user=$_SERVER['PHP_AUTH_USER'];
14 $pass=$_SERVER['PHP_AUTH_PW'];
15
16 if ($user != 'bob' or $pass != 'secret') {
17   header('WWW-Authenticate: Basic realm="'.APPLICATION_NAME.'"');
18   header('HTTP/1.0 401 Unauthorized');
19   echo TEXT_FAIL;
20   exit;
21   }
22 ?>
```

Tout document web qui sera sécurisé par cet exemple ne pourra être visible que si l'internaute saisit 'bob' pour login et 'secret' pour le mot de passe (ligne 16 de `securiteSimple.php`). Naturellement, cet exemple PHP peut être adapté pour aller vérifier des logins/password dans une base de données.
Les variables PHP `$user` et `$pass` sont utilisables dans la page web sécurisée.