

# **Projet Gestion des terrains de foot**

---

Table des matières

1	Analyse préliminaire .....	3
1.1	Introduction .....	3
1.2	Objectifs.....	3
1.3	Planification initiale .....	3
2	Analyse / Conception.....	3
2.1	Concept .....	3
2.2	Stratégie de test.....	3
2.3	Risques techniques .....	4
2.4	Planification .....	4
2.5	Dossier de conception .....	4
3	Réalisation.....	12
3.1	Dossier de réalisation .....	12
3.2	Description des tests effectués .....	18
3.3	Erreurs restantes .....	19
3.4	Liste des documents fournis .....	19
4	Conclusions .....	19
5	Annexes.....	20
5.1	Résumé du rapport du TPI / version succincte de la documentation .....	20
5.2	Sources – Bibliographie.....	20
5.3	Journal de travail .....	20
5.4	Manuel d'Installation .....	20
5.5	Manuel d'Utilisation.....	20
5.6	Archives du projet.....	20

NOTE L'INTENTION DES UTILISATEURS DE CE CANEVAS:

*Toutes les parties en italiques sont là pour aider à comprendre ce qu'il faut mettre dans cette partie du document. Elles n'ont donc aucune raison d'être dans le document final.*

*De plus, en fonction du type de projet, il est tout à fait possible que certains chapitres ou paragraphes n'aient aucun sens. Dans ce cas il est recommandé de les retirer du document pour éviter de l'alourdir inutilement.*

## 1 Analyse préliminaire

### 1.1 Introduction

Une organisation nous mandate pour organiser la planification du tournoi de foot d'un club. Le site doit se constituer doit pouvoir accueillir la partie publique, accessible à tous, une partie « entraîneurs », qui gère la planification de ses propres matchs, et une partie administrateur, qui peut gérer les droits des utilisateurs.

### 1.2 Objectifs

L'organisation est fictive. L'objectif ici est de vérifier les compétences apprises pendant la formation d'informaticien d'entreprise.  
Cela se traduit par la conception d'un site Web dynamique en HTML, CSS, PHP relié à une base de données SQL.

### 1.3 Planification initiale

*Ce chapitre montre la planification du projet. Celui-ci peut être découpé en tâches qui seront planifiées. Il s'agit de la première planification du projet, celle-ci devra être revue après l'analyse. Cette planification sera présentée sous la forme d'un diagramme.*

*Ces éléments peuvent être repris des spécifications de départ.*

## 2 Analyse / Conception

### 2.1 Concept

*Le concept complet avec toutes ses annexes:*

*Par exemple :*

- *Multimédia: carte de site, maquettes papier, story board préliminaire, ...*
- *Bases de données: interfaces graphiques, modèle conceptuel.*
- *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle...*
- *...*

### 2.2 Stratégie de test

*Décrire la stratégie globale de test:*

- *types de tests et ordre dans lequel ils seront effectués.*

- les moyens à mettre en œuvre.
- couverture des tests (tests exhaustifs ou non, si non, pourquoi ?).
- données de test à prévoir (données réelles ?).
- les testeurs extérieurs éventuels.

### 2.3 Risques techniques

- risques techniques (complexité, manque de compétences, ...).

*Décrire aussi quelles solutions ont été appliquées pour réduire les risques (priorités, formation, actions, ...).*

### 2.4 Planification

*Révision de la planification initiale du projet :*

- planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.
- partage des tâches en cas de travail à plusieurs.

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

*Pour planifier un tel projet,*

### 2.5 Dossier de conception

*Fournir tous les document de conception:*

- le choix du matériel HW
- le choix des systèmes d'exploitation pour la réalisation et l'utilisation
- le choix des outils logiciels pour la réalisation et l'utilisation
- site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...
- bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.
- programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...

### 2.5.1 Programmes utilisés



Figure 1 : logo de WampServer

#### Wamp.

Pour concevoir un site internet dynamique, il faut un environnement de test.

Un moyen très simple consiste à faire tourner un serveur PHP/SQL en local. Wamp est une programme simple d'utilisation et gratuit. Il permet d'importer/exporter une base de données, d'insérer des données, les tester, et d'obtenir des requêtes SQL de manière graphiques.

Wamp créé un répertoire dans le disque C. Et tous les sites internet qui y figurent peuvent être reliés aux bases de données importées. C'est ensuite l'occasion de tester chacune des requêtes et des fonctionnalités que l'on souhaite implémenter.

*Wamp intègre de base l'outil PhpMyAdmin. C'est un outil qui permet de gérer la base de données. Ici, la base db\_fields a été importée, et tourne sans erreur.*

#### Capture à Changer



*Ici depuis le site internet, quand on essaie de se connecter, le site renvoie un message qui montre que l'utilisateur est connecté. Mais surtout que le lien entre la base de données et le site internet fonctionne.*

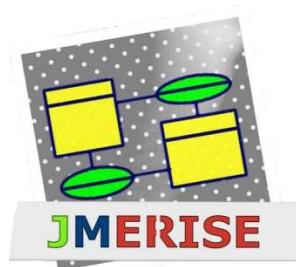


Figure 2 : incone de jMerise

#### JMerise par jfreesoftware

JMerise est un programme qui permet de traduire le dessin de la base de données que l'on a fait à la main, pour le faire comprendre à l'ordinateur. JMerise se débrouille ensuite pour créer un fichier SQL qui permet de créer la base de données.

L'avantage de ce moyen, c'est la simplicité d'utilisation et la rapidité d'exécution. D'autres programmes comme WorkBench peuvent donner le même rendu. Mais le but ici est de gagner du temps, c'est pourquoi cette solution a été privilégiée.

Ce programme n'est plus gratuit. Mais une somme de sept francs est acceptable pour mener à bien un projet de plusieurs semaines.



Figure 3 : icône de PhpStorm

### PhpStorm

PhpStorm est un excellent éditeur pour créer du contenu Web. Les fonctionnalités comme le « refactor » (=le renommage sans erreur), le débogueur, la sauvegarde automatique sont d'autant d'avantages dont on ne saurait se priver.

Ce logiciel n'est pas gratuit, je remercie donc le CPNV de me permettre de m'en servir.



Bienvenue dans phpMyAdmin

### PHPMyAdmin

phpMyAdmin est un logiciel libre écrit en PHP, destiné à gérer l'administration de MySQL sur le Web. phpMyAdmin prend en charge un large éventail d'opérations sur MySQL et MariaDB. Les opérations fréquemment utilisées (gestion des bases de données, des tables, des colonnes, des relations, des index, des utilisateurs, des permissions, etc.) peuvent être effectuées via l'interface utilisateur, tout en ayant la possibilité d'exécuter directement n'importe quelle instruction SQL.<sup>1</sup>

<sup>1</sup> Source, site officiel, <https://www.phpmyadmin.net>, consulté le 16 mai 2022



Figure 4 : interface de PhpMyAdmin



Figure 5 : Logo de SwissCenter

### Swiss Center

SwissCenter est une division fonctionnelle du groupe OpenBusiness SA, actif depuis 1994 dans le domaine des technologies de l'information et de l'informatique de pointe<sup>2</sup>. Cette entreprise nous permet de d'héberger le site Web et de tester les fonctionnalités qui n'étaient pas encore possible, par exemple l'envoi des mails automatiques.

Ceci permettra ensuite de rédiger un manuel d'installation et d'utilisation pour l'administrateur, comme demandé dans le cahier des charges.



Figure 6 : un des logos de Bootstrap

### Bootstrap

Bootstrap est une collection d'outils qui permet de créer un site responsive par exemple. Le code HTML et CSS et JavaScript sont déjà édités pour la maquette sur site.

<sup>2</sup> Source, site officiel, <https://www.swisscenter.com/fr/about>, consulté le 16 mai 2022

Le site de base qui a été retenu se trouve sur <https://themewagon.com>. Le site se nomme Shopper et se trouve sur le site suivant :

<https://technext.github.io/shoppers/contact.html>

Ce site est libre d'utilisation pour une utilisation non commerciale. Ce qui est le cas dans ce projet.

### Gestion de projet

Pour ce projet, où les tâches sont plus ou moins bien définies, j'ai choisi une méthode de « waterfall », une méthode traditionnelle donc. Je peux me permettre cette méthode un peu rigide, parce qu'il ne s'agit pas d'un travail de groupe, et il n'y aura pas de changement des impératifs du cahier des charges.

Cette méthode est simple et rapide à mettre en place, surtout avec un temps limité pour concevoir un planning. Elle est logique et structurée, ce qui permet de s'apercevoir rapidement d'un éventuel retard, et pour prévenir le Chef de projet ou les Experts d'un problème.

### Asana

Asana est une plateforme de gestion de projet qui permet de contrôler l'avancée du travail avec toutes sortes de colonnes de données.

▼ Partie "public"

✓ Page News	10 mai – Aujourd'hui	2,5	2	100 %	
✓ Inscription	4 – 5 mai	2,0	2	90 %	A surveiller
✓ Connexion	3 – 4 mai	2,0	2	100 %	
⊙ Vérification du mail	4 – 5 mai	2,5	0		Problème !
⊙ Mot de passe sécurisé	10 mai – Aujourd'hui	2,5	0	0 %	
✓ Accès interdit pour personnes non-autor	Lundi	1,5	1	100 %	
⊙ Mails automatiques	16 – 18 mai	3,0	1	0 %	
⊙ mise en ligne	Aujourd'hui – 13 mai	2,5	0		
Ajouter une tâche...		TOT	18,5	TOT	8
				TOT	

Figure 2 : exemple de vue du projet sur Asana, consulté le 12 mai 2022

### 2.5.2 Conception de la base de données

La première tâche est d'analyser les données à traiter, et de concevoir un modèle de fonctionnement de la base, ce qu'on appelle un MCD, pour Modèle conceptuel de données. Pour commencer, une bonne pratique est de commencer à la main, pour ensuite faire valider par le chef de projet.

Juste avant de commencer la programmation des fonctions qui permettent la réservation des stades, le MCD plus ou moins définitif va ressembler plus ou moins à ceci :



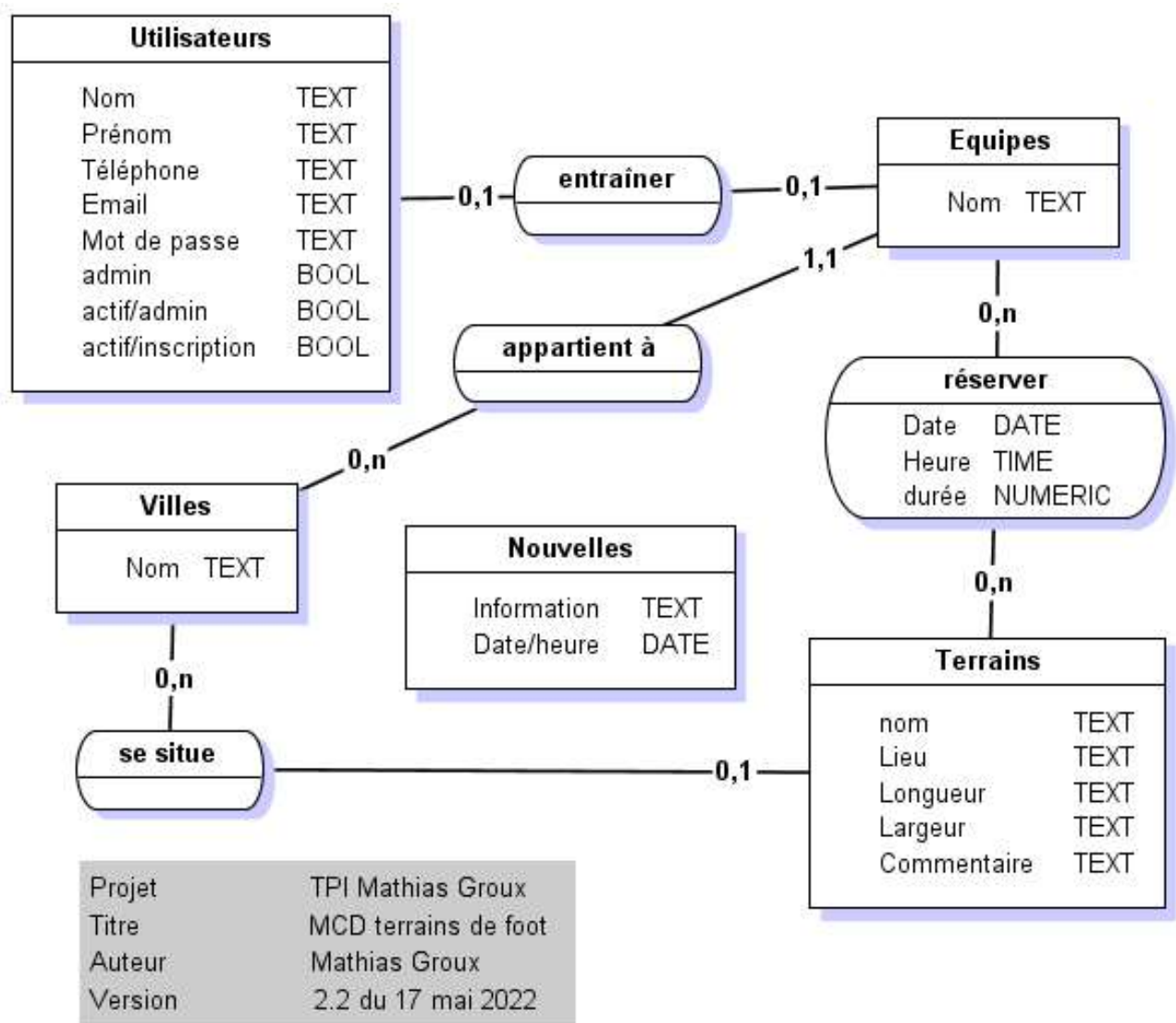


Figure 7 : MCD du site internet

**Entités du schéma :**

Il y a des utilisateurs, qui sont définis par leurs données personnelles, et un champ supplémentaire pour savoir s'ils sont administrateurs ou entraîneurs du site.

Il y a une série d'équipe également définies par leur nom. Chaque équipe se situe dans une ville, et chaque ville dispose d'une liste des terrains disponibles.

Il y a une liste des nouvelles qui sera disponibles pour tout le monde.

La plupart des champs sont définis par le cahier des charges. Pour autant, il y a quelques modifications qui y ont été apportées. Un utilisateur est défini par un pseudo. Ce champ permet une identification plus rapide pour l'utilisateur, plutôt que de passer par son email.

L'utilisateur contient trois valeur booléennes. Une qui précise si l'utilisateur est un administrateur. Deux valeur qui précisent si l'utilisateur et l'administrateur ont donné leur accord pour que l'utilisateur soit actif. Les deux chiffres doivent être égales à 1 pour pouvoir s'identifier avec le compte.

La liste des nouvelles n'a aucune interaction avec les autres tables. Si un administrateur ajoute une nouvelle, elle est stockée dans un espace qui lui est propre, c'est tout.

Une r

**Fonctionnement du schéma**

Si un utilisateur est un entraîneur, il peut entraîner entre zéro et une équipe. Si une équipe est dans le championnat, elle est forcément entraînée par un et un seul entraîneur.

Pour planifier un match, une équipe va faire une réservation, via son entraîneur. Elle intègre une date, une heure et une durée, ainsi les deux équipes en jeu.

Il est libre d'en créer **zéro** et une plusieurs. Et un terrain peut être réservé **entre zéro et plusieurs fois**.

Depuis ce Modèle conceptuel, on en créé le prochain qui est le modèle logique de données. Celui-ci montre comment toutes les informations sont insérées et stockées.

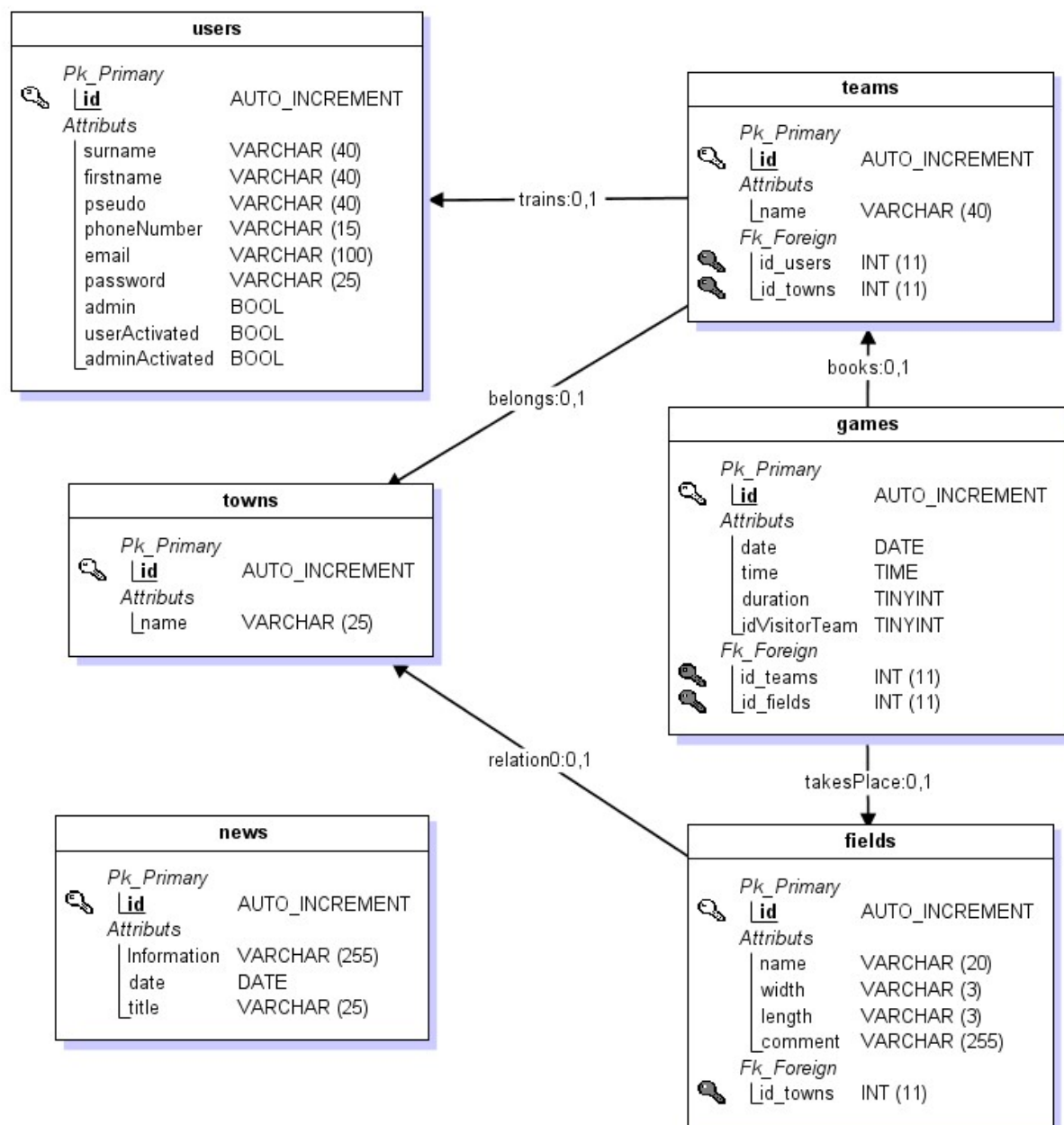


Figure 8 : Modèle logique de données

## Formulaire PHP

Il y a plusieurs formulaires PHP de prévus pour ce projet. Le principe est très simple, et est repris partout sur le site.

```

<form action="formLogin.php" method="post">
  <label>Pseudo </label>
  <input type="text" name="pseudo">
  <input type="submit" value="Se connecter">
  
```

</form>

Et l'autre page PHP est là pour récupérer et traiter les données :

```
$pseudo = $_POST['pseudo'];
```

### Fonctionnement de l'inscription

L'utilisateur va être invité à rentrer ses informations.

Veuillez entrer les informations nécessaires

Le formulaire d'inscription est présenté dans une boîte grise. Il contient six champs de saisie et un bouton. Les champs sont : 'Prénom \*' (avec un astérisque en rouge), 'Nom de famille \*' (avec un astérisque en rouge), 'Email \*' (avec un astérisque en rouge), 'Numéro de téléphone \*' (avec un astérisque en rouge), 'mot de passe \*' (avec un astérisque en rouge), et 'Confirmez votre mot de passe \*' (avec un astérisque en rouge). Le bouton 'ENVOYER' est un rectangle bleu avec le texte en blanc.

Figure 9 : *Formulaire inscription -> Analyse / Réalisation*

***Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !***

## 3 Réalisation

### 3.1 Dossier de réalisation

*Décrire la réalisation "physique" de votre projet*  
*Inscription*

- *les répertoires où le logiciel est installé*
- *la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)*
- *les versions des systèmes d'exploitation et des outils logiciels*
- *la description exacte du matériel*
- *le numéro de version de votre produit !*

- *programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.*

*NOTE : Évitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...*

Structure du site

Page Model DBAccess

Cette page est centrale. Toutes les requêtes à la base de données y sont répertoriées.

Il y a trois fonctions qui sont reprises à chaque requêtes, dbConnect(), executeSQLRequest, et dbUnconnect. Ces trois fonctions permettent de faire le lien, avec la base de données, interroger ou interagir, et refermer la requête pour permettre la suivante de s'exécuter.

Connexion

## Se connecter



Figure 10 : Formulaire de connexion au site

La connexion est un formulaire simple avec deux champs. Un login, et un mot de passe. Le formulaire renvoie les informations sur une deuxième page, qui reçoit les champs, les traite, et fait une vérification avec la base de données.

Hachage du mot de passe

Login

Le login est un formulaire comme décrit dans au point « analyse ». La page formLogin.php va d'abord vérifier que aucun champ n'est vide. Les informations sont ensuite enregistrés dans des variables.

```
// Check if values is not empty
if ( isset($_POST) && (!empty($_POST['pseudo'])) && (!empty($_POST['password'])) ) {
    // Values
    $pseudo = $_POST['pseudo'];
    $password = $_POST ['password'];
}
```

Figure 11 : test si les champs ne sont pas vides et enregistrement des informations

Une fois le premier test effectué, la page va vérifier si les informations sont correctes. Pour cela, il faut appeler la fonction getUser de la page DBAccess, avec comme paramètre le pseudo

```
// gets the user information from the pseudo
$data = $db->getUser($pseudo);
```

Figure 12 : appel de la fonction getUser

La fonction getUser va finalement interroger la base de données, pour savoir si le pseudo correspond à un utilisateur, et demande le mot de passe par la même occasion.

```
public function getUser($pseudo)
{
    // sql request
    $sql = "SELECT id,pseudo,password FROM db_fields.users WHERE pseudo = '$pseudo'";

    // Execute request
    return $this->executeSqlRequest($sql)[0];
} // End getUser()
```

Figure 13 : Fonction qui fait la requête SQL à la base de données

Les informations sont récupérées. Il est maintenant possible de tester si le mot de passe correspond au pseudo. Et ainsi, on peut avertir l'utilisateur que la session est ouverte, et le renvoyer à la page d'accueil.

```
// if password is correct
if ($password == $data['password'])
{
    // Session information
    $_SESSION['pseudo'] = $pseudo;

    // Displays a message to inform the user is logged
    echo "<body onLoad=\"alert('Bienvenue $pseudo')\">";
    //goes back to the index page
    echo '<meta http-equiv="Refresh" content="0;URL=index.php">';
}
```

Figure 14 : test du mot de passe, et redirection à la page d'accueil

### Inscription

Il y a deux cas de figure pour une inscription, il y a donc deux variables booléennes qui représentent « l'accord » pour que cet utilisateur soit actif. Il faut que ces deux valeurs soient égales à 1.

#### Préinscription de l'administrateur

L'administrateur ajoute un utilisateur avec juste l'adresse email. Et il est ajouté à la base de données. Les autres informations personnelles sont entrées plus tard par l'utilisateur.

La variable « adminActivated » à instanciée à 1, car l'administrateur donne son accord de base par ce procédé.

Dès que l'utilisateur souhaite s'inscrire, il y a une vérification si la préinscription a déjà été entrée. Il y a une vérification de l'email. Si tel est le cas, il s'agit juste d'activer le compte en changeant la valeur « userActivated ». Ce procédé est montré en rouge sur la **figure 13**

### Inscription de l'utilisateur

Si l'utilisateur n'est pas encore inscrit, les informations sont enregistrées sous une nouvelle entrée. L'accord de l'utilisateur est instancié à 1 d'office, et l'administrateur pourra donner son aval par la suite. Ce procédé est montré en bleu sur la **figure 13**

### Intérêt pour deux variables booléennes à la place d'une seule

Ce moyen est plus optimisé pour le fonctionnement du site Web. En effet, ces deux variables ne consomment que très peu d'espace de stockage (deux fois 1 bit), plutôt qu'une seule variable qui prendrait d'office au moins huit bits. Par ailleurs, il est probable que le fonctionnement du site soit plus rapide.

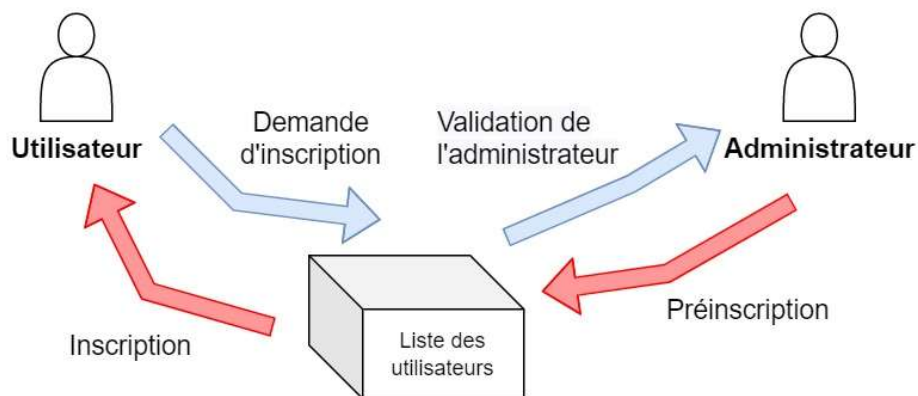


Figure 15 : Fonctionnement de l'inscription

### Affichage des données

L'affichage des données consiste à créer un tableau HTML simple. Une fonction PHP appelle la base de données, et lui demande les informations nécessaires. Le rendu est sous forme d'un tableau[] PHP. Une boucle « foreach » déroule toutes les données dans l'ordre du tableau. Voici un aperçu du code simplifié.



```

<table>
  <tr>
    <th>nom</th>
    <th>largeur</th>
    <th>longueur</th>
  </tr>
  <?php
    //Call class
    $db = new DBAccess();

    //Research all fields information
    $allFields = $db->getAllFields();

    //Display all fields values
    foreach ($allFields as $field)

      echo"
  <tr>
    <td>$field[name]</td>
    <td>$field[width]</td>
    <td>$field[length]</td>
    <td>$field[id]</td>
  </tr>
  ";
  ?>
</table>

```

Figure 16 :Affichage des données

Liste des équipes

Listes des terrains

Tableau des données

Menu de navigation

[AJOUTER ENTRAÎNEUR](#)
[TERRAINS](#)
[EQUIPES](#)
[NEWS](#)
[ENTRAINEURS](#)

DÉCONNEXION

Figure 17 : menu de navigation, image prise le 16 mai 2022

Le menu de navigation en haut du site est le moyen mis à disposition pour les visiteurs, les utilisateurs et les administrateurs pour naviguer sur le site. Le menu est soumis à conditions, les propositions ne sont pas les mêmes pour chacun,

#### Ajout d'un entraîneur

Seul l'administrateur peut ajouter un nouvel entraîneur. Le droit est contrôlé par l'affichage de la page du menu, et lors du chargement de la page. Il faut que le champ « admin » soit égal à 1. Un champ de contrôle est là pour vérifier que l'administrateur entre bien l'email qu'il souhaite.

L'affichage du formulaire est minimaliste. Mais le traitement des données demande certains contrôles.

Un deuxième champ pour contrôler

Plusieurs étapes sont nécessaires

Ajouter nouvel entraîneur



Figure 18 : Formulaire minimaliste pour ajouter un entraîneur

Dans un premier temps, il faut vérifier que les champs ne sont pas vides, comme d'habitudes.

Ensuite, il faut que les deux emails soient les mêmes :

if (\$confirmPassword == password)

Puis, demander à la base de données, si l'email n'existe pas déjà. Cela se passe par la commande

```
SELECT email FROM users WHERE email = 'adresseEmailDeTest@gmail.com'
```

Si le retour est vide, il est maintenant possible d'insérer l'entrée dans la base de données.

```
INSERT INTO `users` (`id`, `surname`, `firstname`, `pseudo`, `phoneNumber`, `email`, `password`, `admin`, `userActivated`, `adminActivated`) VALUES (NULL, "", "", "", "", 'adresseEmailDeTest@gmail.com', "", '0', '0', '1');
```

### 3.2 Description des tests effectués

*Pour chaque partie testée de votre projet, il faut décrire:*

- les conditions exactes de chaque test
- les preuves de test (papier ou fichier)

- *tests sans preuve: fournir au moins une description*

### 3.3 **Erreurs restantes**

*S'il reste encore des erreurs:*

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

### 3.4 **Liste des documents fournis**

*Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions*

- *le rapport de projet*
- *le manuel d'Installation (en annexe)*
- *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
- *autres...*

## 4 **Conclusions**

*Développez en tous cas les points suivants:*

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

## **5 Annexes**

### **5.1 Résumé du rapport du TPI / version succincte de la documentation**

### **5.2 Sources – Bibliographie**

*Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)*

### **5.3 Journal de travail**

### **5.4 Manuel d'Installation**

### **5.5 Manuel d'Utilisation**

### **5.6 Archives du projet**

*Media, ... dans une fourre en plastique*