

# **Projet Gestion des terrains de foot**



---

Table des matières

1	Analyse préliminaire .....	4
1.1	Introduction .....	4
1.2	Objectifs.....	4
1.3	Planification initiale .....	5
2	Analyse / Conception.....	5
2.1	Concept .....	5
2.2	Stratégie de test.....	5
2.3	Risques techniques .....	5
2.4	Planification .....	5
2.5	Dossier de conception .....	6
3	Réalisation.....	11
3.1	Dossier de réalisation .....	11
3.2	Description des tests effectués .....	14
3.3	Erreurs restantes .....	14
3.4	Liste des documents fournis .....	14
4	Conclusions .....	14
5	Annexes.....	16
5.1	Résumé du rapport du TPI / version succincte de la documentation .....	16
5.2	Sources – Bibliographie.....	16
5.3	Journal de travail .....	16
5.4	Manuel d'Installation .....	16
5.5	Manuel d'Utilisation.....	16
5.6	Archives du projet.....	16

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/association de football a besoin d'un site internet pour organiser ses matchs et gérer la réservation des terrains.

## **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

### 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 annexe:*

*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 des 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

#### **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.

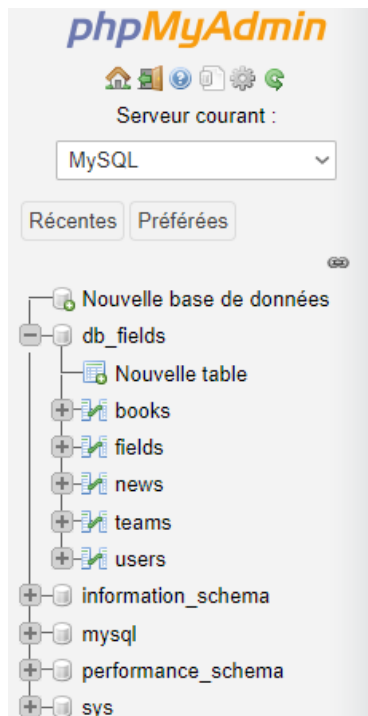
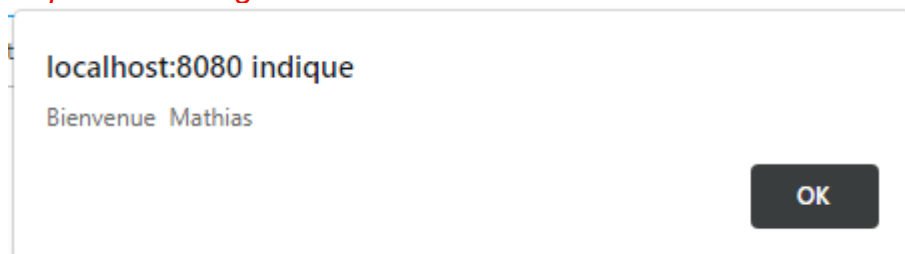


Figure 1 : Interface de PhpMyAdmin

*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.*

### **JMerise par jfreessoft**

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.

### **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.

### PHPMyAdmin

### 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 écrits pour la maquette sur site.

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 :

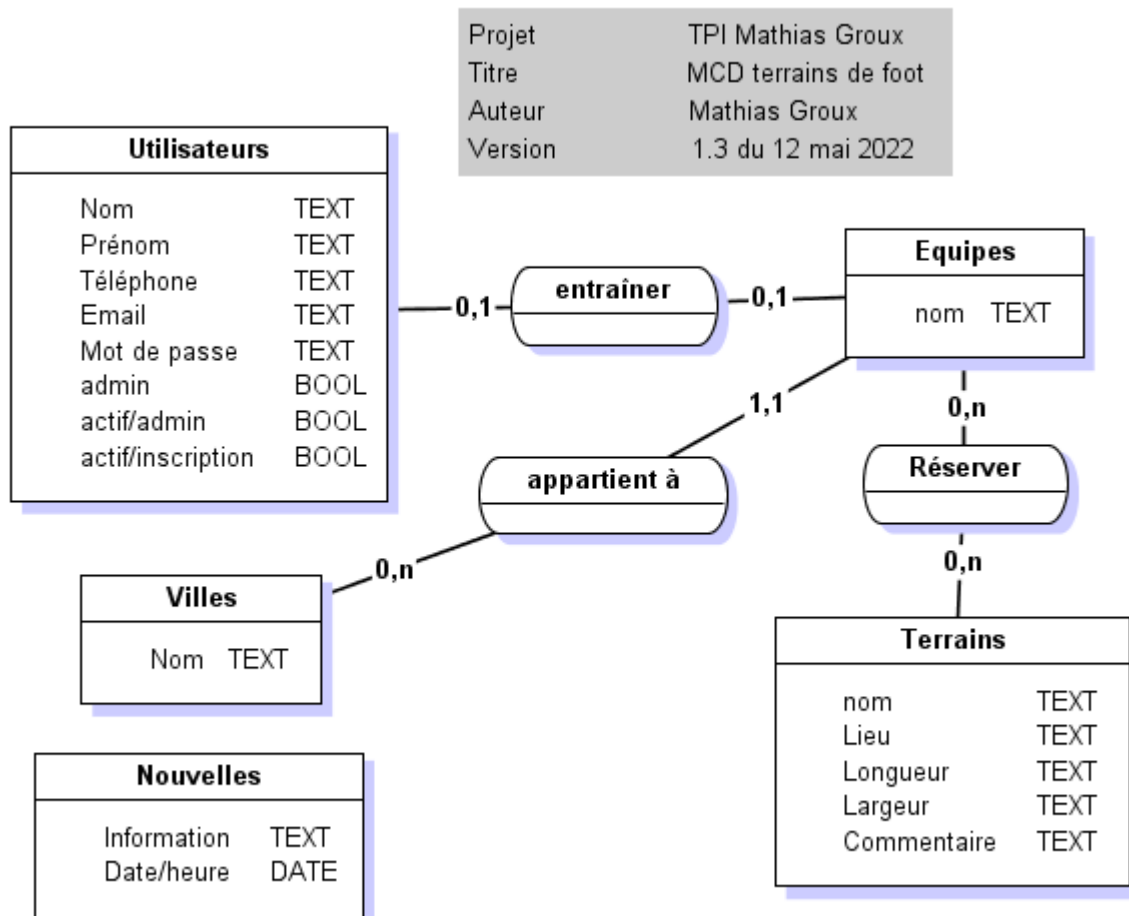


Figure 2 :MCD initiale du site internet

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

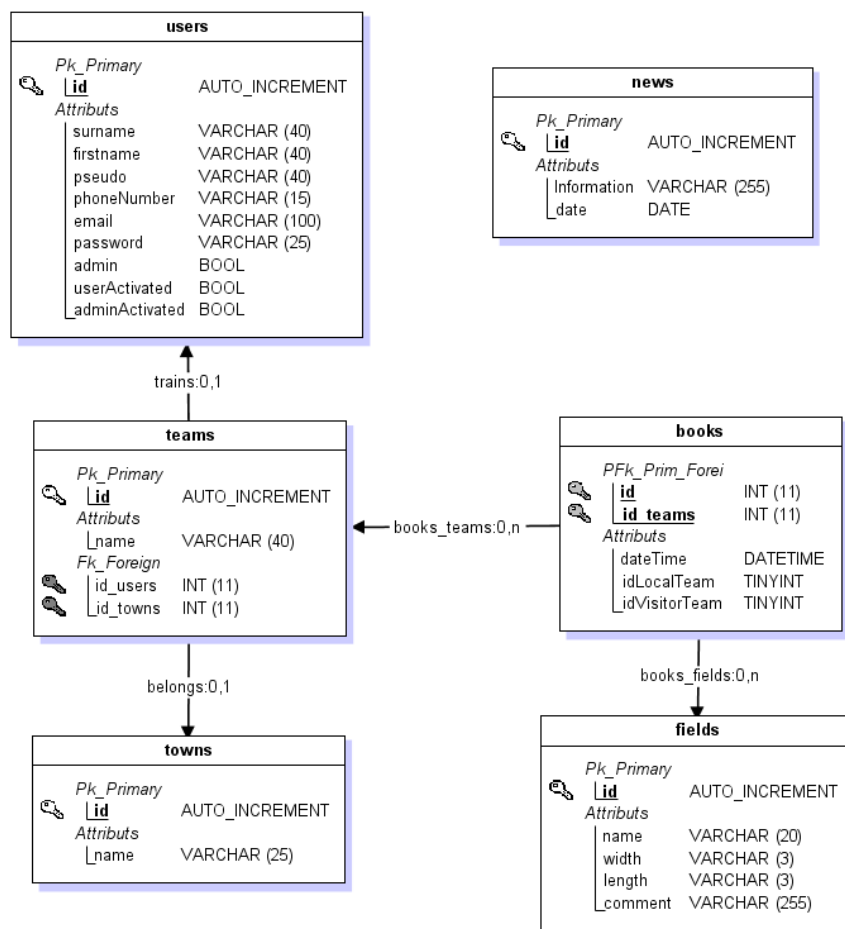


Figure 3 : Modèle logique de données enlever LOCALTEAM

### 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 leurs noms et leurs villes, et une liste des terrains disponibles, définis par le cahier des charges.

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

### 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 et 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**.

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

### Formulaire PHP, formulaire de login

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

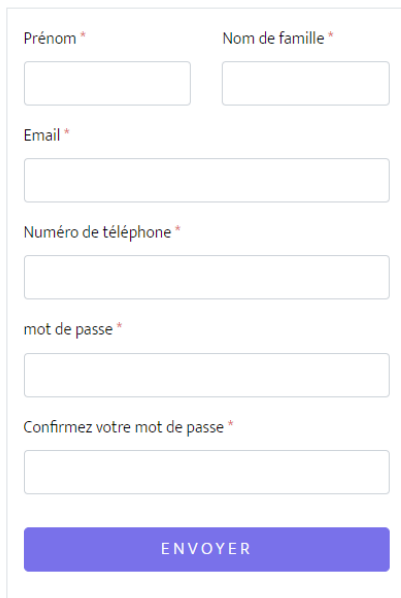
Le formulaire d'inscription est présenté dans une boîte blanche avec une bordure grise. Il contient six champs de saisie et un bouton. Les champs sont : 'Prénom \*' (petit rectangle blanc), 'Nom de famille \*' (petit rectangle blanc), 'Email \*' (rectangle blanc plus long), 'Numéro de téléphone \*' (rectangle blanc plus long), 'mot de passe \*' (rectangle blanc plus long), et 'Confirmez votre mot de passe \*' (rectangle blanc plus long). Le bouton 'ENVOYER' est un rectangle bleu foncé avec du texte blanc, situé en bas du formulaire.

Figure 4 : *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 : Evitez 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

Page DBAccess

Cette page contient toutes les transitions entre les site internet, et les données. 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.

Dans ce

Connexion

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.

Hashage du mot de passe

Inscription

DBAccess

Affichage des données

L'affichage des données sur 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 5 :Affichage des données

Liste des équipes

Listes des terrains

Tableau des données

Menu de navigation

*Formulaires*

### **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*