

Projet Fil Rouge  
ASPTT Nantes Handball Club

# Spécifications Techniques

Tétiana Savitska  
Vincent Canarelli  
Clémentine Tournery  
Romuald Tuffreau  
Morgan Tréguier

## Objectifs du projet

L'objectif du site promouvoir le site de l'ASPTT et d'offrir aux membres du club un espace de consultation et de diffusion des activités proposées par le club.

L'interface doit être attrayante et responsive et doit comporter une page de contact et un outil de gestion de planning afin de permettre aux sponsors de contacter le club et aux joueurs et personnel de l'ASPTT de consulter les dates des matchs.

Il doit permettre aux différents acteurs (membres, internautes, gestionnaires, sponsors) d'interagir avec le site avec différents niveaux de droits afin de restreindre la consultation, la diffusion et l'édition de certains contenus (photos, plannings, événements ...)

## Périmètre du projet

Conformément au cahier des charges fourni par le client la liste des prestations à la charge des élèves de l'IMIE et à la charge du client est la suivante :

## Prestation de l'équipe de développement

- La programmation entière du site, coté back comme coté front
- La rédaction d'un guide utilisateur
- Un livret d'aide au déploiement pouvant contenir des conseils sur le choix de l'hébergeur ou du mode d'hébergement
- Le choix de l'outil de travail, en dehors des contraintes spécifiées par le cahier des charges

## Restant à la charge de l'ASPTT

- Le dépôt du nom de domaine
- Le choix du mode d'hébergement du site et de l'hébergeur
- Le référencement naturel du site et l'amélioration du positionnement par les moteurs de recherche
- La maintenance et les mises à jour après livraison
- La mise en place d'outils statistiques
- L'internationalisation du site n'est pas proposée par l'équipe de développement, le site est en français

## Sont déterminés conjointement

- Le design du site après réunion avec le client

## Choix techniques

### Problématique

Les critères de performances étant un élément important mais pas primordial pour un site à portée locale Nantes (club handball).

L'évolution du site se fera en termes de fonctionnalités et par le développement de nouveaux outils plutôt qu'en termes d'accroissement du trafic.

Par conséquent et afin de faciliter le déploiement sur un serveur Debian (cf : guide de déploiement dernier chapitre) le site est en PHP5.

### Contraintes techniques

Le site doit contenir une base de données relationnelle.

Le site peut soit être développé avec le framework Symfony soit être une solution CMS Wordpress

### Frameworks utilisés

Programmation : Symfony3

Tests : PHPUnit version 5.7

### Technologies utilisées

Coté front

- Twig
- JQuery
- JavaScript
- Bootstrap

Coté back

- PHP5 (php 5.6 minimum)

## Services associés

MySQL  
Apache2

### Liste des pages – site public

Page d'accueil

Liste des plannings

- Affichage par week-end
- Affichage par année

Détail d'un planning

Liste des équipes

- Page des cadets
- Page des juniors
- Page des seniors
- Toutes les équipes

Page à propos

Page de contact

Page d'inscription

Page de connexion

Page plus d'infos

# Règles de nommage

*Par respect des conventions et des bonnes pratiques et pour nous faciliter le travail l'équipe de développement utilisera les règles de nommage suivantes*

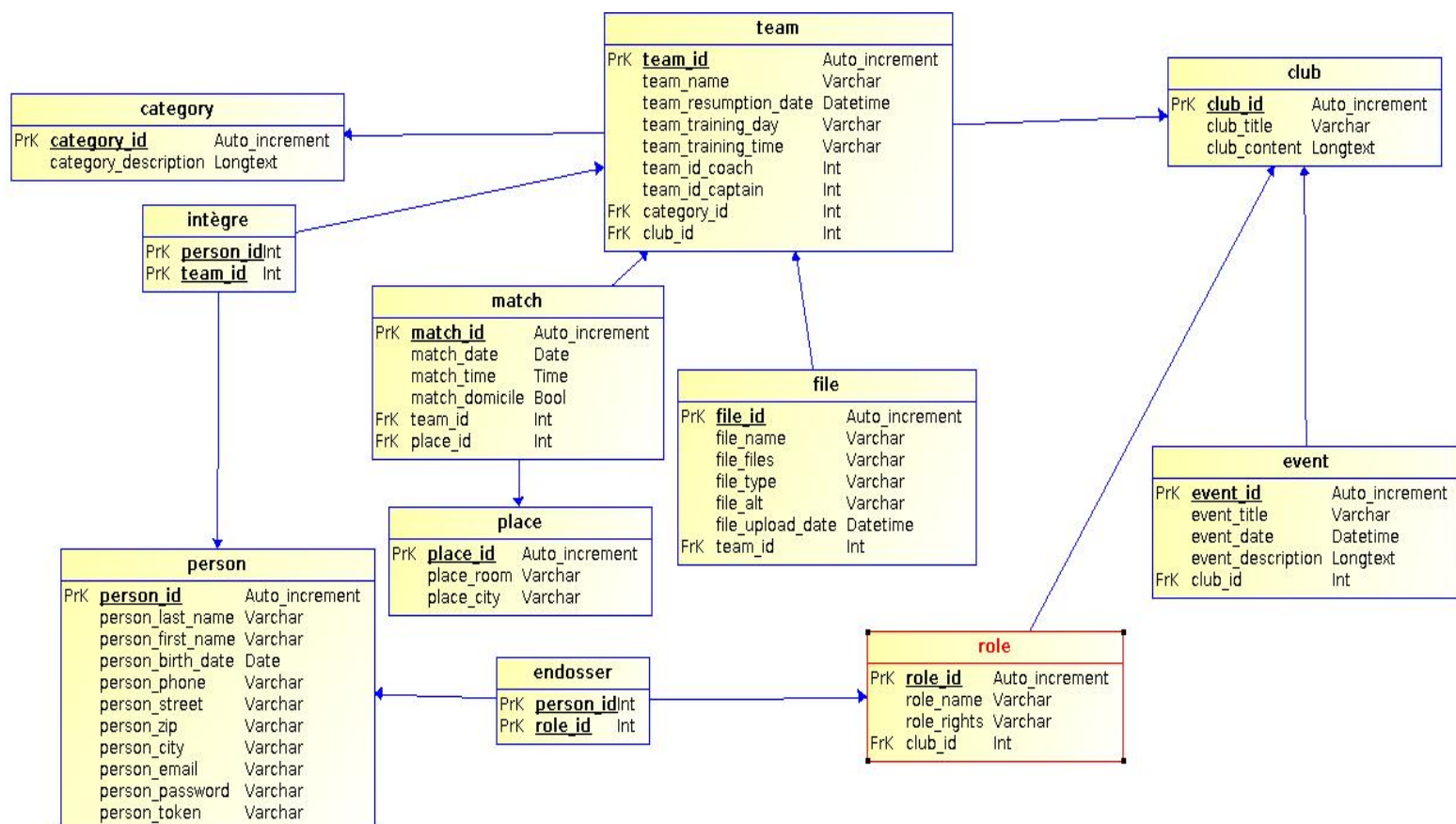
## Règles de nommage de la base de données

### Noms des tables

- Entièrement en minuscule séparé par des underscore si besoin (snake\_case)
- En anglais

### Noms des colonnes

- Comme pour les tables, entièrement en minuscule séparé par des underscore si besoin (snake\_case)
- Préfixés par le nom de la table
- Les clés primaires et étrangères s'appellent « nom\_de\_la\_table\_id »
- En anglais



## Règles de nommage du code source

### Nom des variables

- En minuscule en distinguant la séparation entre mots par une majuscule (lowerCamelCase)
- En anglais

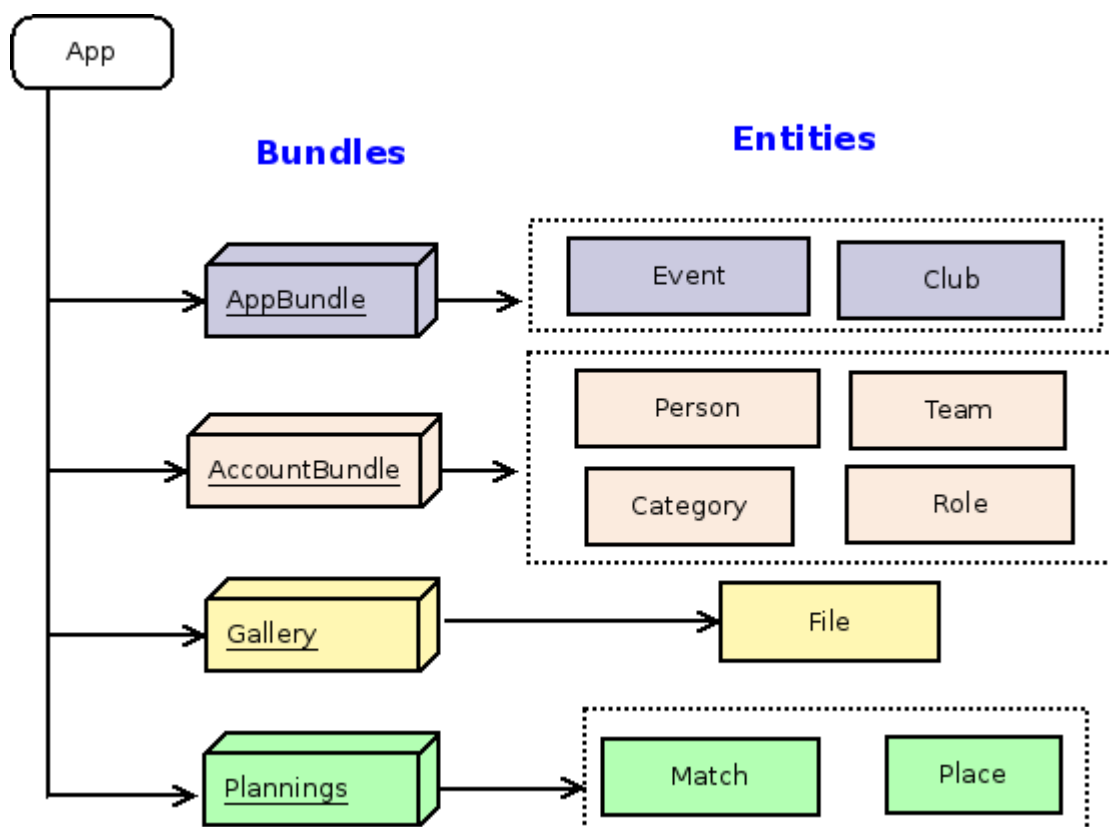
### Nom des classes

- Avec la première lettre en majuscule en distinguant la séparation entre mots par une majuscule (UpperCamelCase)
- En anglais

### Nom des constantes

- En majuscule, avec des mots séparés par des underscore si besoin est
- En anglais

## Architecture du site



Afin de faciliter la maintenance du code source et l'évolution future du projet nous avons organisé notre code en parties séparées

- **PlanningBundle** : partie dédiée à l'outil de gestion de plannings
- **AccountBundle** : partie consacrée aux sessions et aux comptes utilisateurs
- **AppBundle** : partie générale du site avec les pages de contact, d'historique ...
- **GalleryBundle** : partie dédiée aux galeries d'images des équipes

## Spécifications des formulaires

### Inscription

#### Champs obligatoires

- Nom
- Prénom
- Mot de passe (x2)
- E-mail

#### Contraintes sur les champs

- Mot de passe
  - longueur minimale → 5 caractères
  - caractères obligatoires → 1 majuscule, 1 minuscule, 1 caractère spécial, 1 chiffre
  - hashé en base de données → algorithme bcrypt, coût 12
- Nom et Prénom
  - Ne doivent pas contenir de chiffres



- E-mail  
doit être formaté correctement

## Contact

### Champs obligatoires

- Nom
- Objet
- E-mail → doit être correctement formaté
- Message

## Planning

### Champs obligatoires

- Date
- Équipe (x2)

### Champ équipe

Bandeau déroulant qui liste les équipes enregistrées en base de données

### Champ Lieu

Bandeau déroulant qui liste les lieux enregistrés en base de données

## Droits et sessions

- Identification des utilisateurs non authentifiés pour restreindre l'accès à la galerie de photos
- Création de plusieurs types de droits attribués à un rôle au sein du club

Exemples : Rôle joueur → niveau de droit Utilisateur

Rôle trésorier → niveau de droit Manager

Rôle entraîneur → niveau de droit Manager

Rôle administrateur → niveau de droit Admin

## Réglage du pare feu de Symfony selon les règles suivantes

### Hiérarchie des rôles :

- **Admin** : niveau de droit maximum (droit d'administration + droit de gestion + droit d'utilisation du système)
- **Manager** : niveau de droit moyen (droit de gestion + droit d'utilisation)
- **Utilisateur** : niveau de droit faible (droit d'utilisation, de consultation de contenu nécessitant d'être authentifié)

**Anonyme** : niveau de droit minimum (pas de consultation du contenu nécessitant d'être authentifié)

### Constantes utilisées (table rôle):

ROLE\_USER, ROLE\_MANAGER, et ROLE\_ADMIN

Les champs role\_name et role\_rights de la table correspondent respectivement au nom du rôle dans le club et a la constante du niveau de droit autorisé

Exemple :

role		
Id	role_name	role_rights
1	joueur	ROLE_USER
2	famille	ROLE_USER
3	entraîneur	ROLE_MANAGER
4	administrateur	ROLE_ADMIN

### Connexion de l'utilisateur

Le champ qui servira d'identifiant à l'utilisateur est son e-mail, il doit être unique en base de données.

Le champ qui servira à authentifier d'utilisateur est le mot de passe. Il faudra l'entrer deux fois à l'inscription pour éviter les erreurs de saisie.

## Contrôle des accès

Le pare feu de symfony disposant d'une liste pour contrôler les accès, certaines urls seront réservées aux utilisateurs suivants :

**/login** : utilisateurs non authentifiés

**/register** : utilisateurs non authentifiés

**/admin** : administrateur uniquement

**/m** : manager et administrateur

Ainsi seuls les utilisateurs non authentifiés auront la possibilité de créer un compte ou de se connecter.

Seuls les administrateurs et managers auront les droits d'édition, suppression ou création de contenu.

Seul l'administrateur aura accès au panneau d'administration.

## Spécifications techniques détaillées : fonctionnalité planning

### Fonctions de services

La fonctionnalité planning va permettre à l'administrateur de saisir les matchs à venir de la saison, de

les modifier ou de les supprimer si besoin.

Cette fonctionnalité permettra aussi à un utilisateur lambda de consulter les matchs à venir pour

toutes les équipes du club de handball.

### Fonctionnement

Pour utiliser cette fonctionnalité, l'utilisateur doit se placer sur l'onglet « plannings » du site puis

cliquer sur le bouton « nouveau match » en bas à droite.

L'écran lui permet alors de saisir un match, pour cela il doit renseigner plusieurs champs :

## Description des données

Date du match	
Vue	Menu déroulant
Données	datetime
Niveau min	2017
Niveau max	2022

Domicile	
Vue	Case à cocher
Données	Booléen

Lieu	
Vue	Menu déroulant
Données	Varchar (128)

Equipe	
Vue	Menu déroulant
Données	Varchar(64)

Equipe adverse	
Vue	Menu déroulant
Données	Varchar(64)

Lieu, équipe et équipe adverse représentent un enregistrement d'une équipe ou d'un lieu en base de données (clés étrangères), on récupérera les valeurs du menu déroulant par une requête.

## Contraintes

- Le champ « Date du match » devra attendre une date. L'utilisateur ne pourra pas saisir de valeur. Il devra obligatoirement passer par le menu déroulant. La date devra être cadrée (c'est-à-dire que l'utilisateur ne devra pas pouvoir saisir de date « fantaisiste »)
- Le champ « domicile » devra être une case à cocher. Ça devra être un booléen, l'utilisateur ne devra pas y avoir accès d'une autre manière que celle citée au-dessus.
- Le champ « lieu » devra attendre une chaîne de caractères. Mais cette valeur devra être disponible uniquement depuis un menu déroulant et ne devra pas être saisi d'une autre manière. Pour cela, il faudra au préalable que la base de données ait été chargée et que les informations attendues aient été saisies au préalable.
- Le champ « équipe & équipe adverse » devront attendre une chaîne de caractères. Les valeurs attendues devront être uniquement disponible depuis un menu déroulant, l'utilisateur ne devra pas pouvoir la saisir autrement que par ce moyen. Pour cela, il faudra au préalable avoir accès à la base de données, que celle-ci renvoie correctement l'information et que l'utilisateur ait renseigné au préalable l'information.

## Description des traitements

### Affichage de la liste des plannings

Une requête vers `planning/` permet l'affichage de la liste

- **Filtrage par week-end**  
Pour naviguer d'un planning à l'autre il y a un système de pagination  
Le filtre par week-end permettra d'affiner la liste des matchs  
Une requête en base de données permettra de choisir la semaine  
Granularité : semaine par semaine  
Route : `planning/week/{id}`
- **Filtrage par année**  
Le filtre par année permet de voir les matchs d'une manière plus globale sur une année entière.  
Une requête en base de données permettra de choisir l'année  
Route : `planning/year/{year}`

## Affichage du détail d'un planning

Une requête vers une page de détails *match/{id}/show* permettra d'afficher les champs suivants :

Date et heure : type datetime

Lieu : booléen, indique si le match est à domicile ou non

Salle : type varchar, représente le lieu du match

Equipe : type varchar, nom de l'équipe disputant le match

Equipe adverse : type varchar, nom de l'équipe adverse

## Création d'un planning

Réservé à l'administrateur et aux managers

Une requête vers la route *matches/new* affichera un formulaire de création de match

L'utilisateur autorisé devra renseigner les champs

## Modification d'un planning

Réservé à l'administrateur et aux managers

Une requête vers la route *matches/{id}/edit* affichera un formulaire de modification de match, une requête SQL sur l'identifiant du match permettra d'afficher les informations préexistantes dans le formulaire

L'utilisateur autorisé devra renseigner les champs

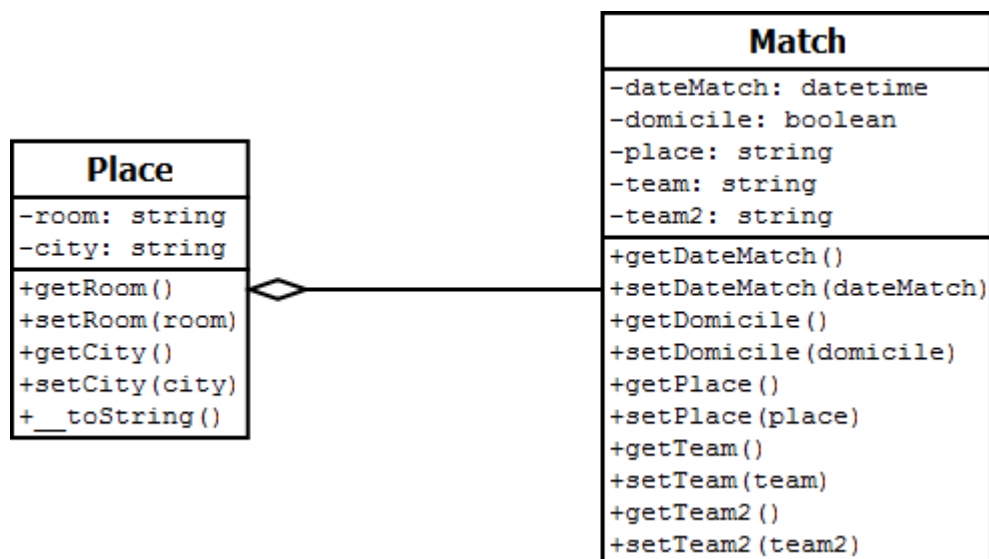
## Suppression d'un planning

Réservé à l'administrateur et aux managers

Une requête vers *matches/{id}/delete*

Une requête SQL sur l'identifiant du match permettra de supprimer l'entrée correspondante en base de données.

## Modélisation de la classe



# Choix de l'environnement de production - comparatif

## Red Hat

**Avantages** : support très rapide, applications relativement récentes, mises à jour de sécurité suivies.

**Inconvénients** : payant (à partir de 299\$ US à l'année).

## Debian

**Avantages** : très stable, communauté importante, documentation bien faite et en français, utilisation plutôt simple, Long Term Support : mises à jour de sécurité assurées jusqu'en 2020.

**Inconvénients** : peu de nouveautés dans les dépôts de paquets, support uniquement communautaire (pas immédiat).

## Ubuntu

**Avantages** : paquets plus récents que sous Debian, communauté très importante, utilisation simple, documentation en français.

**Inconvénients** : un peu moins stable que Debian, certains dépôts sont propriétaires.

## Archlinux

**Avantages** : avant-gardiste, dépôts de paquets très à jour, il y a toutes les nouveautés logicielles, communauté active.

**Inconvénients** : peu stable car les paquets des dépôts sont « testing », donc peu adapté à un environnement de production.