

Récapitulatifs des demandes

1^{ère} demande : Création d'une de base de données :

- Indexer les informations via aux fichiers CSV (licence 2014, 2015 ...) dans la base **as-football**.
- Permettre la gestion des inscriptions des licenciés.
- Informier l'AFNIL sur le traitement de données personnelles concernant sur les informations pour les licenciés.
- Mettre en place un système d'accès pour les consultations d'informations sur les licenciés de la futur base de données.
- Ouverture de fiche individuelles.
- La base de données **devra mise en ligne sur un serveur** pour être accessible pour tous les utilisateurs sur poste Windows ou Debian avec accès Internet (adminer sio-hautil).

1.2) L'attribution des accès :

La gestion des droits d'accès sera géré par l'administrateur en autre les salariés de as-football (Président ? ...).

Concernant les utilisateurs sera géré à travers un login et mot de passe déposé par l'administrateur par mail de l'utilisateur en question. (Login, MDP).

Certains utilisateurs qui **pourront consulter et suivre leurs coordonnées** et d'autres **pourront aussi consulter et suivre des coordonnées de d'autres utilisateurs et de plus faire des modifications**.

Les catégories des accès :

Type de profil	Création des utilisateurs / gestion des droits	Saisie des licences, coordonnées, photos	Saisie des données financières, Attestation de paiement	Consultation/ Coordonnées	Suivi sportif	Financier	Suppression d'un licencié	Initialisation des saisons	Exploitation des données	Mailing
Administrateur	X	X	X	X	X	X	X	X	X	X
Dirigeant		X		X	X				X	X
Trésorier /secrétaire		X	X	X	X	X	X		X	X
Licencié visiteur				X	X					

1.3) Gestion des saisons:

Remise à niveau de la base de données as-football sera nécessaire pour chaque saison. Une saison dure du de septembre à juillet, cette remise à niveau permettre de rajouter les nouveaux licenciés et de garder en même temps les anciens licenciés toujours effective.

Les informations d'un licencié antérieur, les saisons passés seront obligatoirement affichées.

Le choix d'une devra être visible pour l'utilisateur.

1.4) L'exploitation des données :

L'affichage des données devra aussi à la fois être sur l'interface web ainsi que sur l'application mobile.

L'exploitation des données devra **s'appuyer sur l'ancienne base de données de 2014** en récupérant ses données. De plus les données sont regroupées en catégories :

- La liste des utilisateurs et des droits d'accès

- Par licenciés :

 - Les coordonnées du **joueur et des parents si celui-ci est mineur**

 - Le suivi sportif du joueur (catégories, licence, formations...)

 - Le suivi financier du joueur

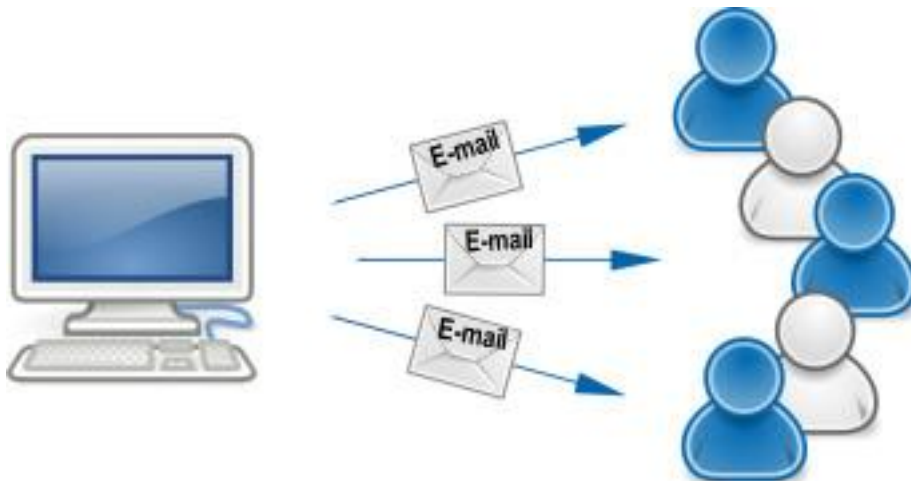
 - Le suivi des commandes, habillements. (Etats des paiements)

Les licenciées sont disponibles à l'adresse suivante :

[Les licenciés \(anciennes SGBD\)](#)

Schéma MCD de la futur SGBD 1^{ère} version :

licencié permettra de proposer une liste de noms ou prénoms selon le texte entré dans le formulaire de recherche et sortira les données utiles sous forme de liste de licenciés afin de permettre l'ouverture de fiches individuelles.

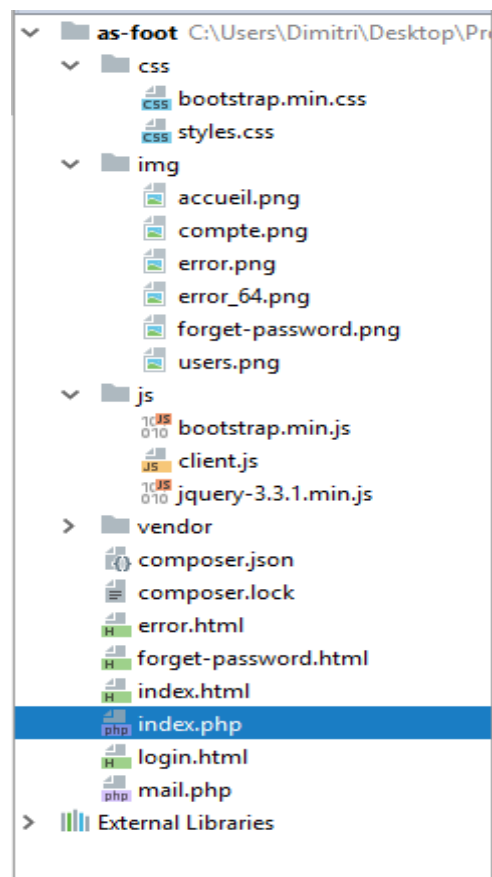


1.6) Logo & Site :

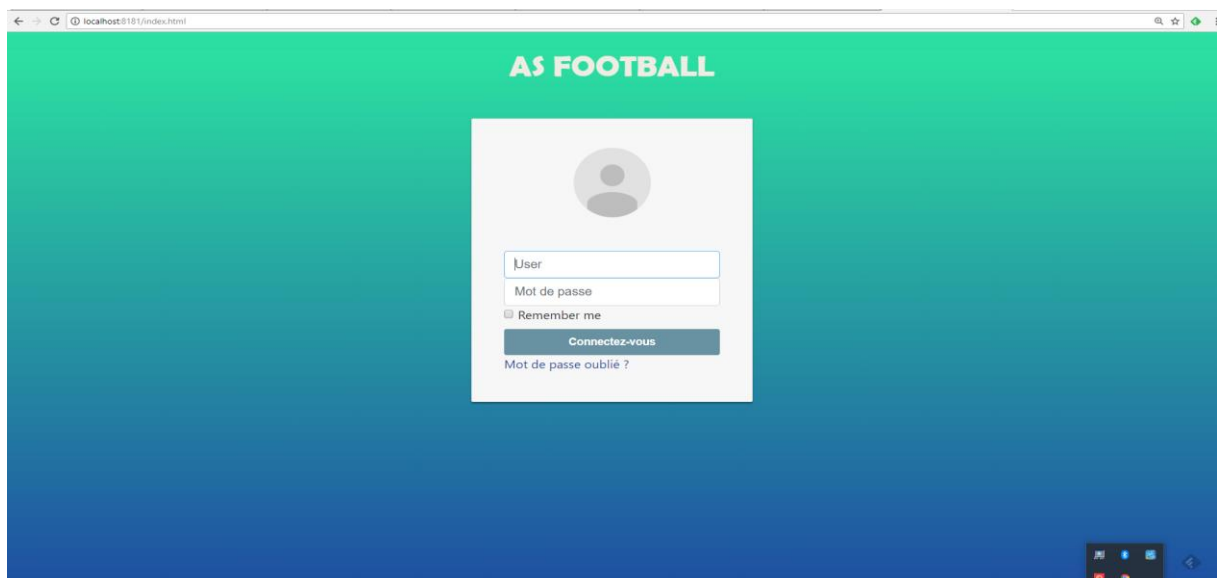
Les codes couleurs utilisés sera sous plusieurs versions : fond blanc, bleu et d'éviter les couleurs pour le logo.

1.7) Création du site :

Arborescence du site as-foot :

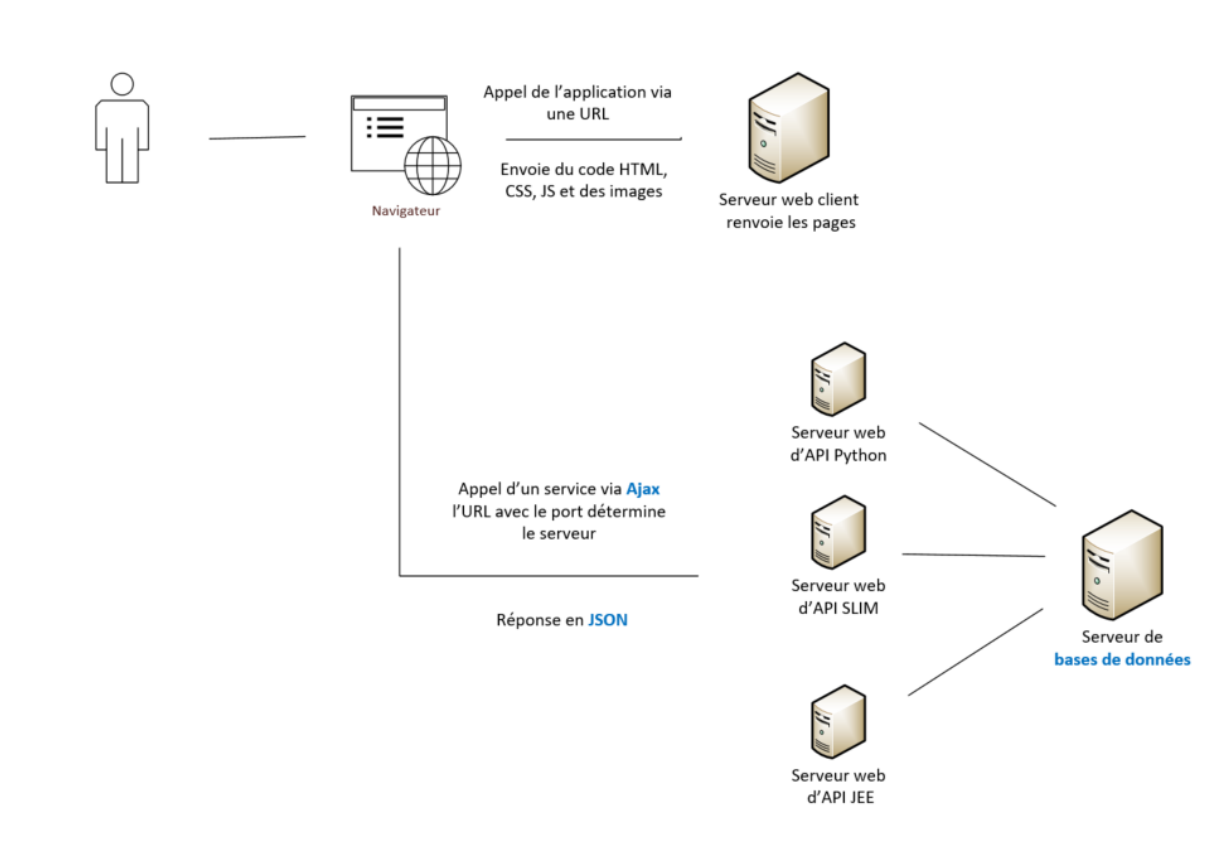


Exemple du futur site :



Architecture du site :

Le site doit obligatoirement respecter cette architecture.



Ce projet se fera en PHP donc utilisera **le serveur slim** pour se connecter à la base.



Exemple code : index.php-> serveur slim

```

$app= new \Slim\App;
error_reporting( level: E_ALL);

$app->post( pattern: '/login',function (\Slim\Http\Request $request, \Slim\Http\Response $response){
    $login = $request->getQueryParam( key: 'user');
    $password = $request->getQueryParam( key: 'password');
    $bb=PostConnector($login,$password);
    $aff = json_encode($bb, options: JSON_PRETTY_PRINT);
    return $aff;
});

/*$app->get('/users',function (Request $request, Response $response){
    return GetUsers();
});
$app->get('/dimitri',function (Request $request,Response $response){
    return $response->getBody()->write("C'est moi");
});
$app->get('/{login}/{mdp}', function (Request $request,Response $response){
    $login = $request->getAttribute('login');
    $password = $request->getAttribute('name');
    return Getcompte($login,$password);
});*/

function Connexion (){ // SQL DATABASE -> 'test-as'
    $pdo = new PDO ( dsn: 'mysql:host=localhost;dbname=test-as', username: 'root', passwd: '');
    return $pdo;
}

```

```

function Connexion (){ // SQL DATABASE -> 'test-as'
    $pdo = new PDO ( dsn: 'mysql:host=localhost;dbname=test-as', username: 'root', passwd: '');
    return $pdo;
}

function PostConnector($login,$password){
    session_start();
    try {
        $sql = "SELECT * FROM users where login = :login AND password= :password"; // vérifie le mot de passe , :password = a clé , affecter par 'password' => $password
        $dbh = Connexion();
        $req = $dbh->prepare($sql);
        $tb=array(
            'login' => $login,
            'password' => $password,
        );
        $req->execute($tb);
        $res = $req->fetch();
        return $res;
    }catch(PDOException $e){
        return '{"error":'.$e->getMessage().'}';
    }
}

```


Ensuite il aura le serveur client qui aura pour but de transmettre les pages correspondant à travers les routes appelés par le navigateur selon la méthode.

Exemple code Serveur client -> client.js :

```
$(document).ready(function() {
    $('#connexion').click(function () {
        var Success = false;
        var user=$('#inputUser').val();
        var password=$('#inputPassword').val();
        alert(user);
        alert(password);
        $.ajax({
            type: "POST",
            contentType: 'application/json; charset=utf-8',
            url: "/login", //?user="+user+"&password="+password,
            success: function (data) {
                console.log(data);
                Success = true;
                location.href = 'login.html?user='+user; // renvoie la page en question
            },
            error: function (data) {
                console.log(data);
                Success = false;
                Location.href='error.html';
            }
        });
    });
});
```

Planification des tâches :

