





AP

6- Développement de l'application web mobile

1- Rappel de l'architecture applicative globale et objectif de cette étape

La page suivante présente l'architecture générale de l'application **TraceGPS**.

Cette étape porte sur le **développement et le test de l'application web mobile** qui offrira différentes fonctionnalités aux utilisateurs :

Le suivi (ou pratiquant) pourra effectuer les actions suivantes :

- créer un compte utilisateur
- demander un courriel avec un nouveau mot de passe en cas d'oubli du mot de passe
- tester la géolocalisation
- changer de mot de passe
- retirer l'autorisation d'accéder à tous ses parcours
- démarrer l'enregistrement d'un parcours, transmettre régulièrement sa position, et arrêter l'enregistrement d'un parcours
- supprimer un parcours
- consulter les données de son parcours

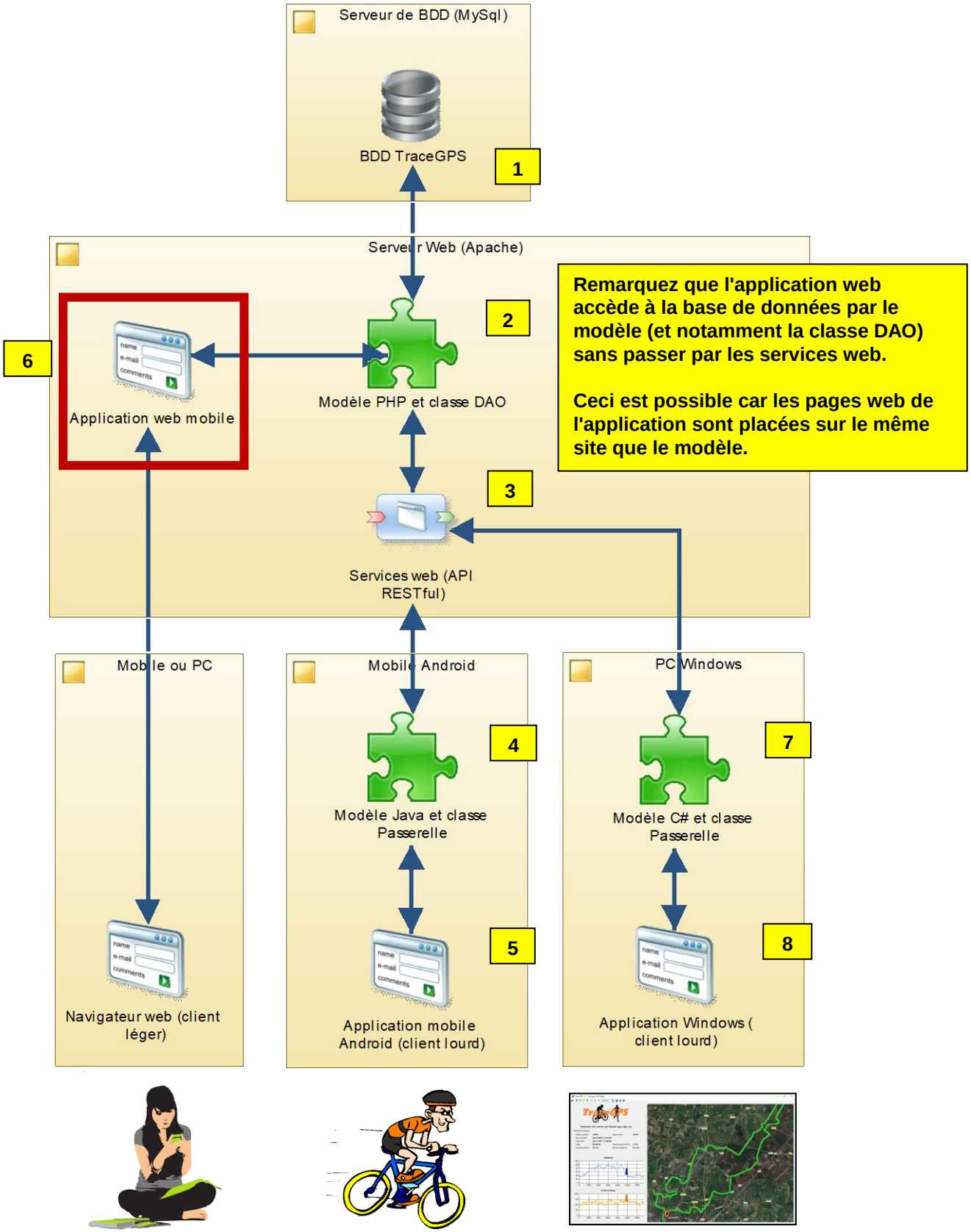
Le suiveur (un proche du pratiquant) pourra effectuer les actions suivantes :

- créer un compte utilisateur
- demander un courriel avec un nouveau mot de passe en cas d'oubli du mot de passe
- tester la géolocalisation
- changer de mot de passe
- demander à un pratiquant l'autorisation de le géolocaliser quand il effectue un parcours seul
- consulter les données d'un parcours (s'il a été auparavant autorisé par le pratiquant)

L'administrateur pourra effectuer les actions supplémentaires suivantes :

- consulter la liste des utilisateurs
- supprimer un utilisateur

L'application web mobile proposera également de télécharger l'application mobile Android à tout visiteur connecté à partir d'un terminal Android.

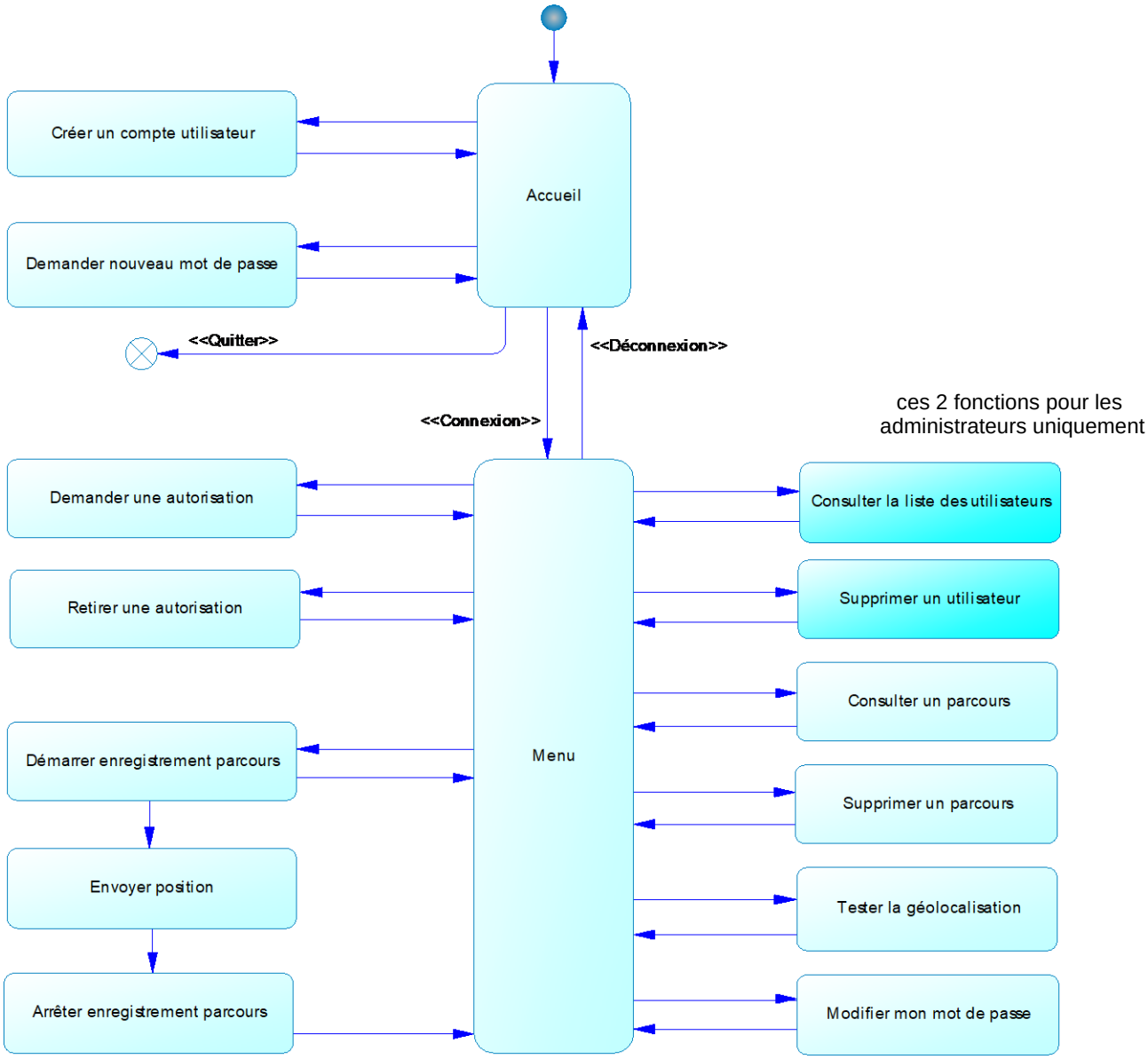

Serveur de BDD (MySql)
BDD TraceGPS
1
Serveur Web (Apache)
2
Modèle PHP et classe DAO
6
Application web mobile
3
Services web (API RESTful)
Remarquez que l'application web accède à la base de données par le modèle (et notamment la classe DAO) sans passer par les services web.
Ceci est possible car les pages web de l'application sont placées sur le même site que le modèle.
Mobile ou PC
Navigateur web (client léger)
Mobile Android
4
Modèle Java et classe Passerelle
5
Application mobile Android (client lourd)
PC Windows
7
Modèle C# et classe Passerelle
8
Application Windows (client lourd)

1-2 Choix des outils et des langages

- **MySql** : c'est le SGBD utilisé pour la base de données **tracegps**
- **PHP 5.5** : version de PHP compatible avec les serveurs **WAMP** utilisés en développement
- classe **PDO** : pour permettre aux pages PHP d'accéder aux bases de données de façon plus actuelle et plus sécurisée (les requêtes préparées permettent d'éviter l'injection SQL)
- **jQuery mobile** : framework basé sur **HTML5**, **CSS3** et **jQuery**, permettant de développer rapidement des applications web mobiles
- Pattern **MVC** (**Modèle-Vue-Contrôleur**) : architecture applicative utilisée par la plupart des frameworks de développement PHP

1-3 Navigation dans le site

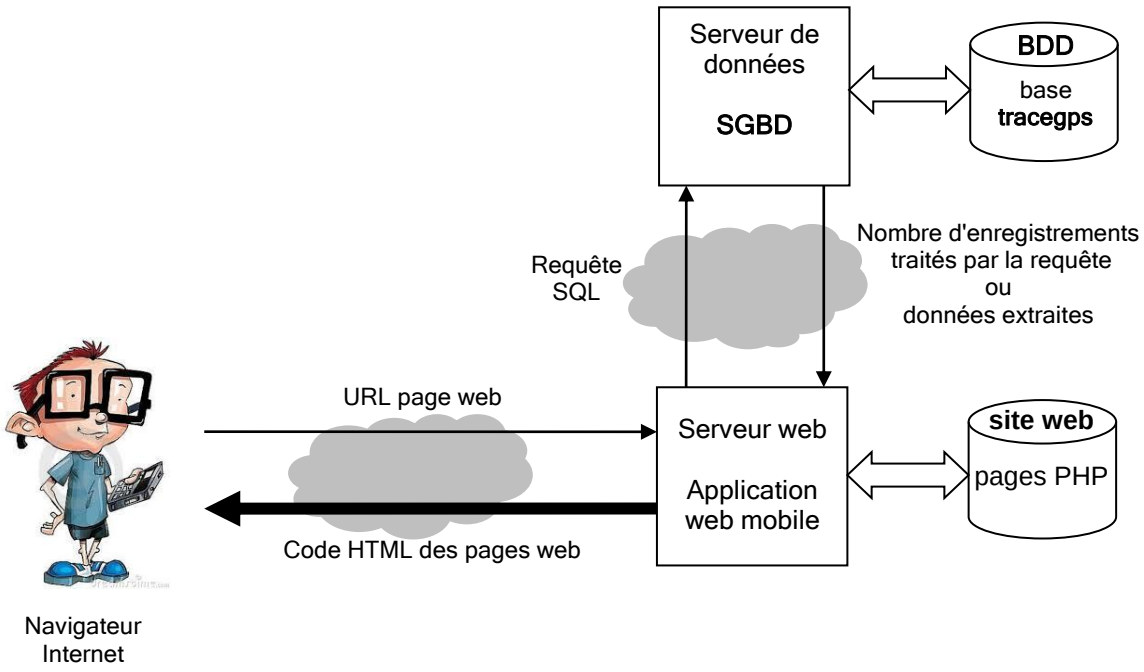
La navigation dans le site est présentée ici à l'aide d'un **diagramme d'activités** UML :



2- Architecture générale de l'application

2-1 Architecture 3 tiers classique

L'application web mobile **TraceGPS** sera basée sur une **architecture 3 tiers** classique :



2-2 Hébergement de l'application et de la base de données

Le développement (en **PHP**) et la mise au point de l'application web pourra se faire sur un hébergement local (**WAMP** sur **localhost** ou **serv-wamp1** ou **serv-wamp2**).

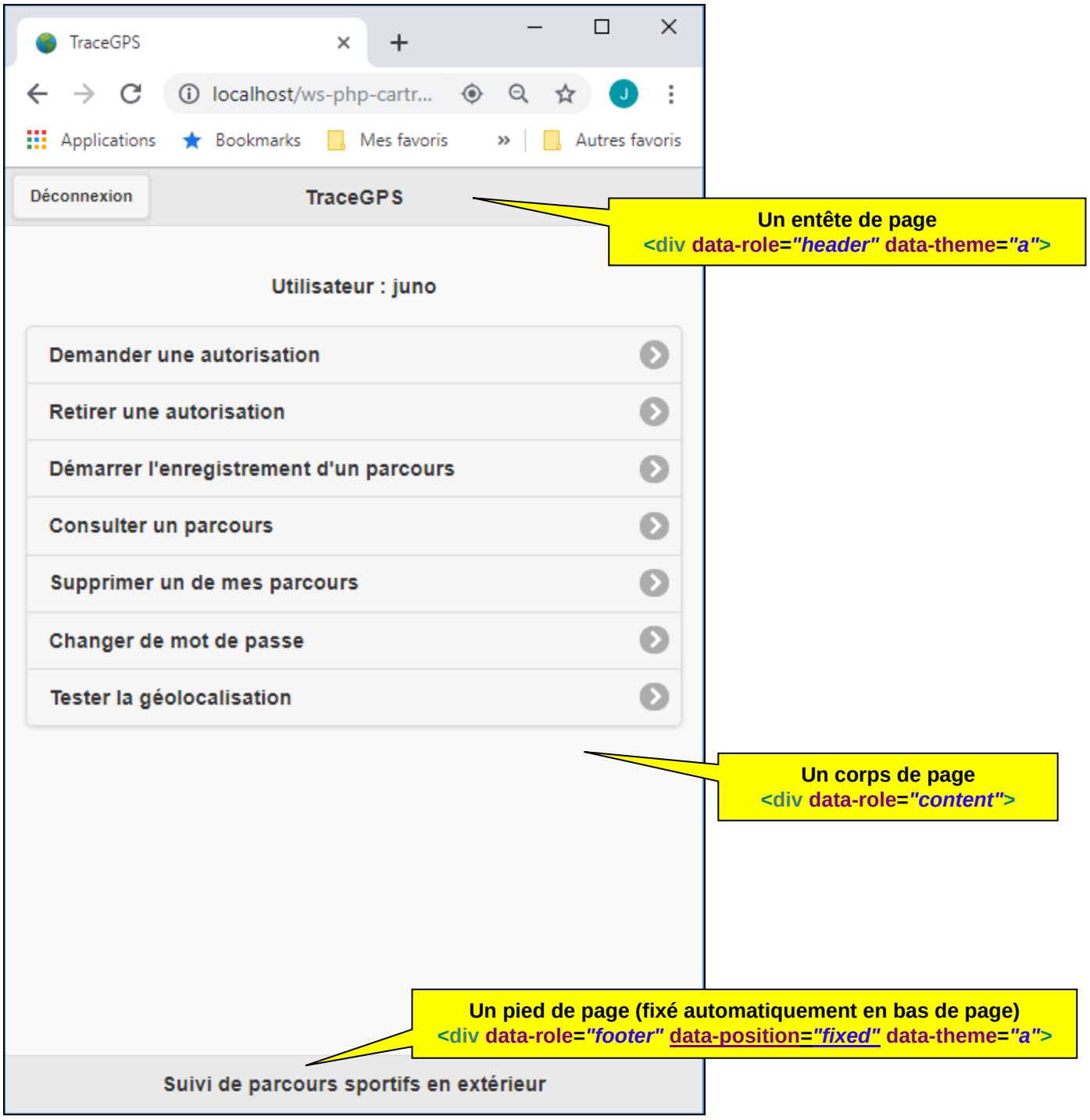
En revanche, l'appel de cette application web à partir d'un vrai mobile nécessite un **hébergement internet**, car les mobiles n'ont pas l'accès au réseau local du lycée.

Chaque groupe de projet devra donc se créer un hébergement sur Internet (vous disposez très souvent d'un espace de stockage avec votre opérateur Internet), et y installera :

- la base de données **tracegps** (le script de création est fourni)
- l'application web mobile que vous aurez développée

3- Structure d'une page jQuery mobile

La page suivante représente le menu de l'application web mobile :



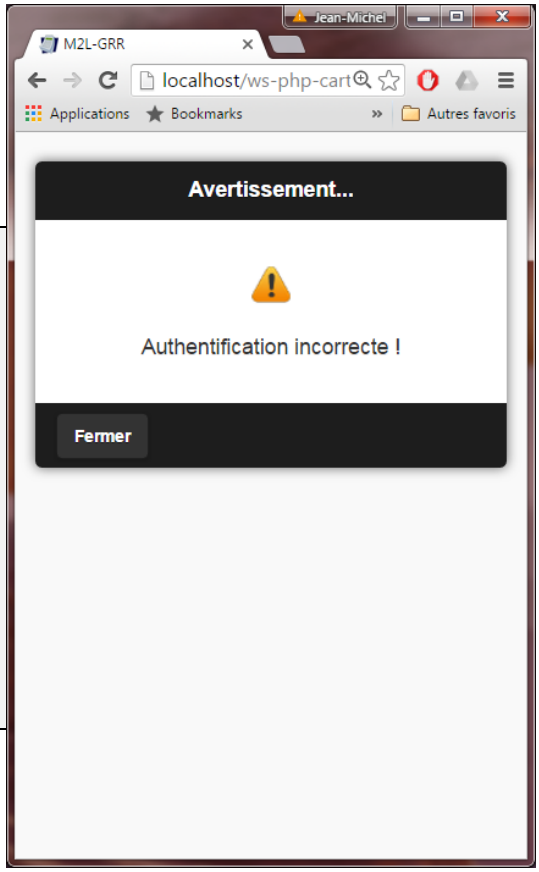
Et voici le code HTML envoyé au navigateur du client :

```
<!doctype html>
<html>
    <head>

        <title>TraceGPS</title>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1">

        <!-- favicon -->
        <link rel="shortcut icon" type="image/x-icon" href="./images/favicon.ico" />
        <link rel="shortcut icon" type="image/png" href="./images/favicon.png">
        <link rel="icon" type="image/x-icon" href="./images/favicon.ico" />
        <link rel="icon" type="image/png" href="./images/favicon.png" />
        <link rel="apple-touch-icon" href="./images/apple-touch-icon.png" />

        <link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
        <script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
        <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
    </head>
    <body>
        <div data-role="page">

            <div data-role="header" data-theme="a">
                <h4>TraceGPS</h4>
                <a href="index.php?action=Connecter" data-ajax="false" data-transition="flip">Déconnexion</a>
            </div>

            <div data-role="content">
                <h4 style="text-align: center; margin-top: 20px; margin-bottom: 20px;">Utilisateur : juno</h4>
                <ul data-role="listview" data-inset="true">
                    <li><a href="index.php?action=ChoisirUtilisateurPourDemanderAutorisation"
                            data-transition="flip">Demander une autorisation</a></li>
                    <li><a href="index.php?action=ChoisirUtilisateurPourRetirerAutorisation"
                            data-transition="flip">Retirer une autorisation</a></li>
                    <li><a href="index.php?action=DemarrerEnregistrementParcours" data-ajax="false"
                            data-transition="flip">Démarrer l'enregistrement d'un parcours</a></li>
                    <li><a href="index.php?action=ChoisirUtilisateurPourConsulterParcours"
                            data-transition="flip">Consulter un parcours</a></li>
                    <li><a href="index.php?action=ChoisirParcoursASupprimer"
                            data-transition="flip">Supprimer un de mes parcours</a></li>
                    <li><a href="index.php?action=ChangerDeMdp" data-ajax="false"
                            data-transition="flip">Changer de mot de passe</a></li>
                    <li><a href="index.php?action=TesterGeolocalisation" data-ajax="false"
                            data-transition="flip">Tester la géolocalisation</a></li>
                </ul>
            </div>

            <div data-role="footer" data-position="fixed" data-theme="a">
                <h4>Suivi de parcours sportifs en extérieur</h4>
            </div>

        </div>
    </body>
</html>
```

Ces 3 lignes incluent les frameworks jQuery et jQuery mobile

3-1 Gestion des messages d'erreurs

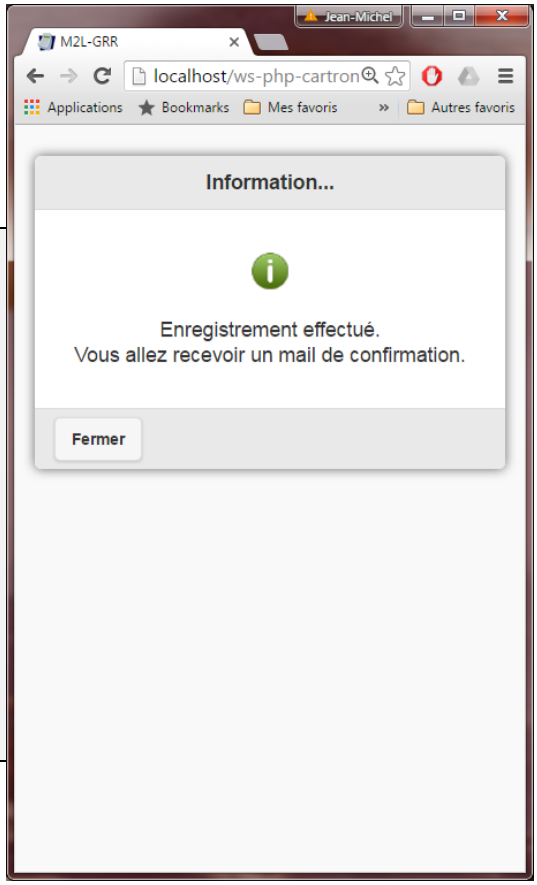
Les messages d'erreur sont affichés dans une boîte de dialogue avec un thème à fond noir (thème **b** avec jQuery 1.4.5) :




```
<div data-role="dialog" id="affichage_message" data-close-btn="none">
    <div data-role="header" data-theme="b">
        <h3>Avertissement...</h3>
    </div>

    <div data-role="content">
        <p style="text-align: center;">
            <img src="images/avertissement.png" class="image" />
        </p>
        <p style="text-align: center;">Authentification incorrecte !</p>
    </div>

    <div data-role="footer" class="ui-bar" data-theme="b">
        <a href="#page_principale" data-transition="flip">Fermer</a>
    </div>
</div>
```

3-2 Gestion des messages d'information

Les messages d'information sont affichés dans une boîte de dialogue avec un thème à fond gris (thème **a** avec jQuery 1.4.5) :




```
<div data-role="dialog" id="affichage_message" data-close-btn="none">
    <div data-role="header" data-theme="a">
        <h3>Information...</h3>
    </div>

    <div data-role="content">
        <p style="text-align: center;">
            <img src="images/information.png" class="image" />
        </p>
        <p style="text-align: center;">Enregistrement effectué.<br>
            Vous allez recevoir un mail de confirmation.</p>
    </div>

    <div data-role="footer" class="ui-bar" data-theme="a">
        <a href="#page_principale" data-transition="flip">Fermer</a>
    </div>
</div>
```

3-3 Paramétrage des styles

Les styles sont initialisés dans le fichier **index.php** :

- La version de jQuery mobile est déterminée par la variable **$version**
- Le thème de base est déterminé par la variable **$themeNormal**
- Le thème des messages d'erreurs est déterminé par la variable **$themeProbleme**
- La transition entre les pages est déterminée par la variable **$transition**

```
// choix des styles graphiques
$version = "1.4.5";         // choix de la version de JQuery Mobile (voir fichier head.php) : 1.2.0,  1.2.1,  1.3.2,  1.4.5
$themeNormal = "a";         // thème de base
$themeProbleme = "b";       // thème utilisé pour afficher un message en cas de problème
$transition ="flip";        // transition lors des changements de page :
                            // (pop, flip, fade, turn, flow, slidefade, slide, slideup, slidedown)
```

Le site **http://api.jquerymobile.com/** permet de se tenir informé des nouvelles versions de jQuery mobile.

La version de **jQuery mobile** est prise en compte à l'aide du fichier **vues/head.php** qui est inclus dans le bloc **<head>** de chaque vue :

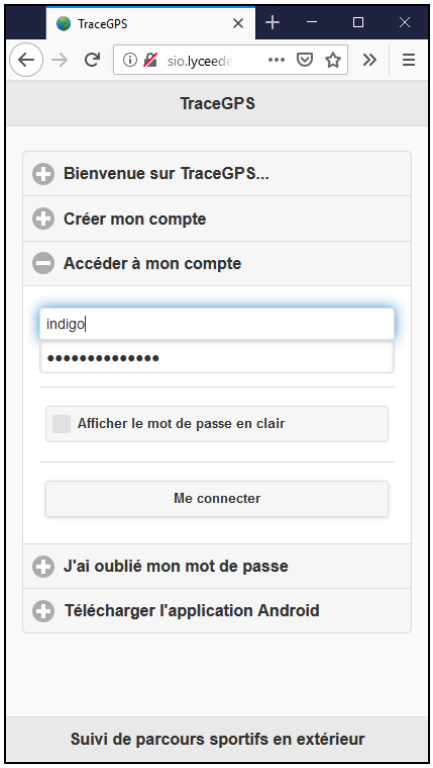
```
<?php
// Projet TraceGPS - version web mobile
// fichier : vues/head.php
// Rôle : fournit les données du bloc <head> de toutes les vues
// Dernière mise à jour : 01/11/2021 par dP

// Ce fichier inclut les données du bloc <head> :
//  - jeu de caractères
//  - balises <meta>
//  - l'icone de l'application
//  - la feuille de styles et le framework JQuery Mobile (en fonction de la variable $version réglée dans index.php

?>

        <title>TraceGPS</title>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1, minimum-scale=1, maximum-scale=1">

        <!-- favicon -->
        <link rel="shortcut icon" type="image/x-icon" href="./images/favicon.ico" />
        <link rel="shortcut icon" type="image/png" href="./images/favicon.png">
        <link rel="icon" type="image/x-icon" href="./images/favicon.ico" />
        <link rel="icon" type="image/png" href="./images/favicon.png" />
        <link rel="apple-touch-icon" href="./images/apple-touch-icon.png" />

<?php
if ($version == "1.2.0" ) { ?>
        <link rel="stylesheet" href="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.css">
        <script src="http://code.jquery.com/jquery-1.8.2.min.js"></script>
        <script src="http://code.jquery.com/mobile/1.2.0/jquery.mobile-1.2.0.min.js"></script>
<?php };

if ($version == "1.2.1" ) { ?>
        <link rel="stylesheet" href="http://code.jquery.com/mobile/1.2.1/jquery.mobile-1.2.1.min.css" />
        <script src="http://code.jquery.com/jquery-1.8.3.min.js"></script>
        <script src="http://code.jquery.com/mobile/1.2.1/jquery.mobile-1.2.1.min.js"></script>
<?php };

if ($version == "1.3.2" ) { ?>
        <link rel="stylesheet" href="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.css" />
        <script src="http://code.jquery.com/jquery-1.9.1.min.js"></script>
        <script src="http://code.jquery.com/mobile/1.3.2/jquery.mobile-1.3.2.min.js"></script>
<?php };

if ($version == "1.4.5" ) { ?>
        <link rel="stylesheet" href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.css">
        <script src="http://code.jquery.com/jquery-1.11.1.min.js"></script>
        <script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-1.4.5.min.js"></script>
<?php }; ?>
```

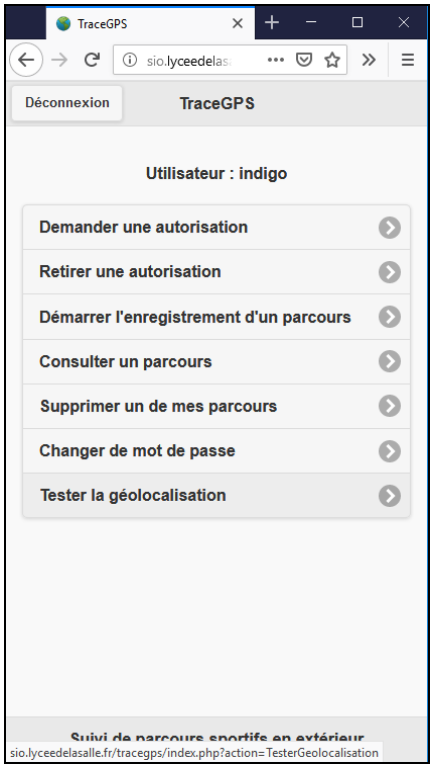
4- L'IHM (Interface Homme-Machine)

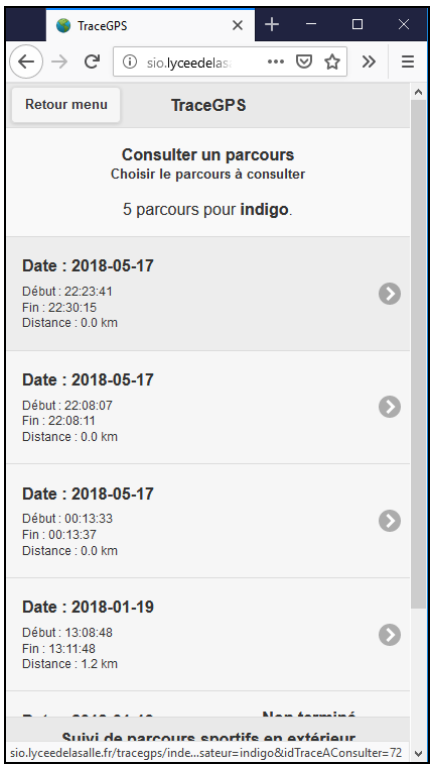
Vous pouvez avoir un aperçu complet de l'IHM en testant l'application installée chez OVH par votre cher professeur, à l'adresse :

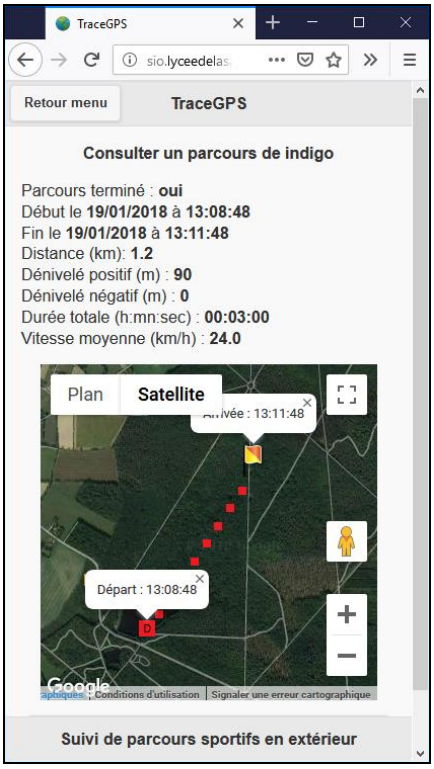
http://sio.lyceedelasalle.fr/tracegps

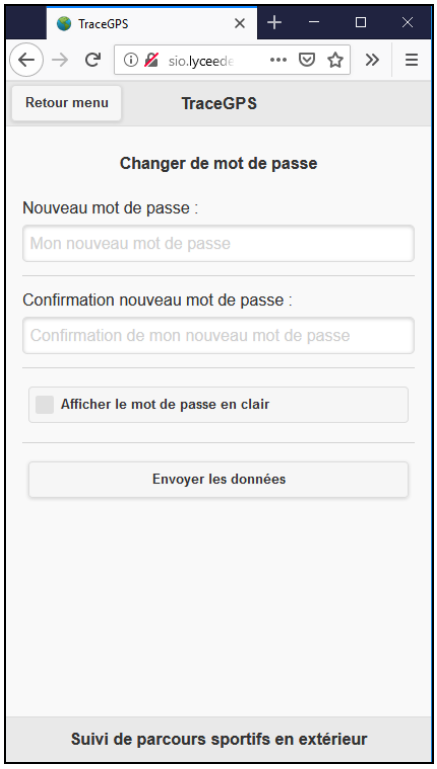
En voici quelques extraits :











5- Le modèle de conception MVC (Modèle-Vues-Contrôleurs)

5-1 Les styles et patrons d'architecture

Les **patrons d'architecture** (en anglais : **architecture patterns**) sont des modèles de référence en développement de logiciels.

Le patron **modèle-vue-contrôleur (MVC)** est souvent utilisé dans le développement d'applications comportant des interfaces graphiques. Il repose sur 3 composants :

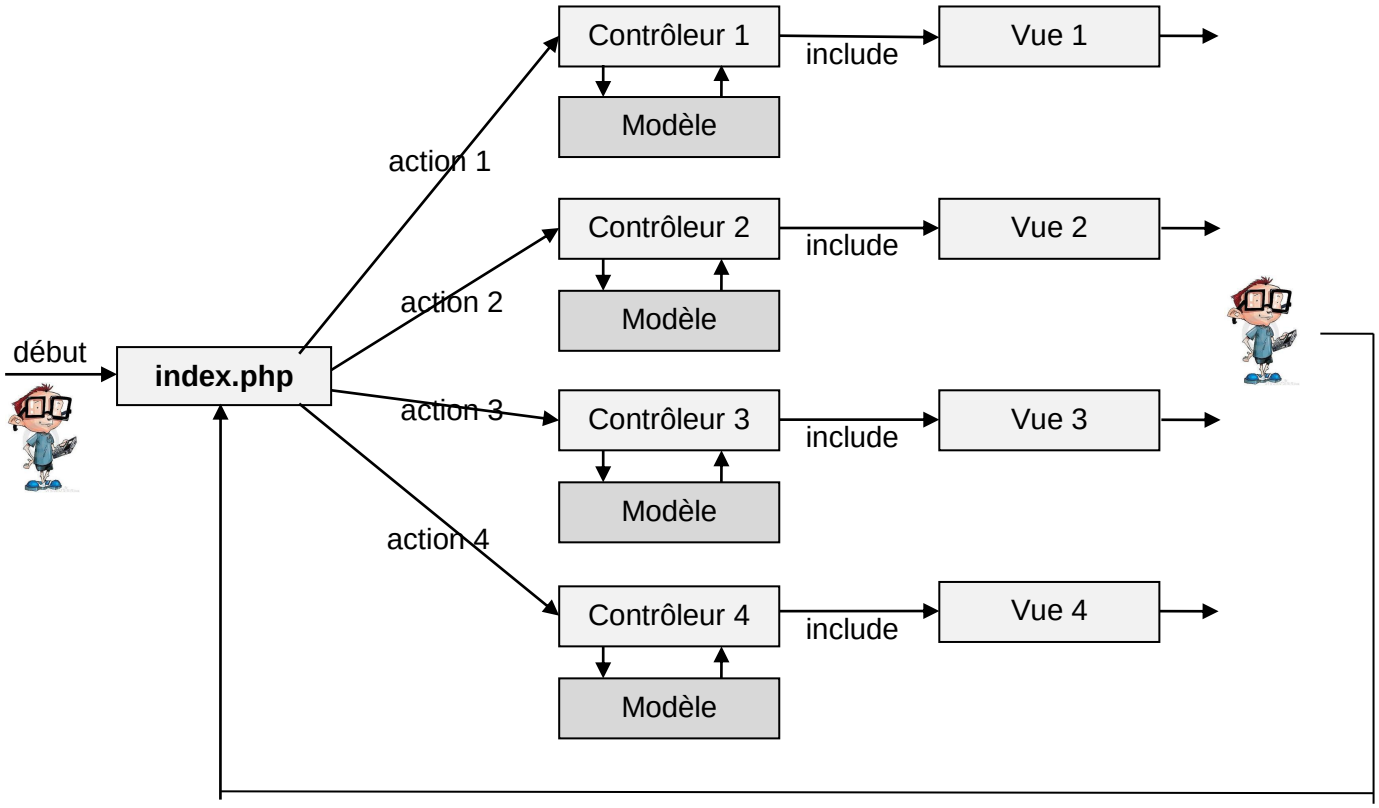
- le **modèle** qui correspond aux classes métiers et aux classes d'accès à la base de données
- les **vues** qui décrivent l'IHM
- les **contrôleurs** qui prennent en charge les opérations effectuées par l'utilisateur à travers l'IHM et les transforment en messages destinés aux vues

5-2 L'enchainement des différents composants

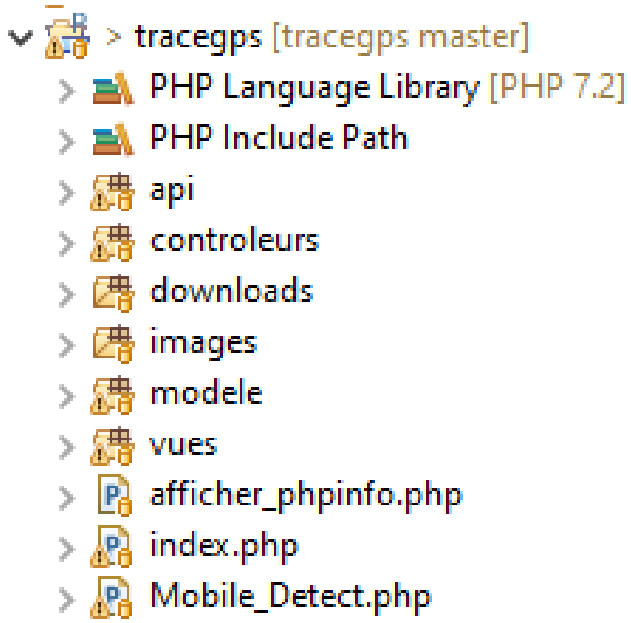
Tous les liens et les validations de formulaires appellent la page **index.php** avec un parametre **action** :

http://...../tracegps/index.php**?action=xxxxxx**

La page **index.php** redirige vers le contrôleur correspondant à l'action.
Chaque contrôleur traite la demande (en utilisant généralement le modèle) et préparer les données à insérer dans la vue envoyée au client.



5-3 L'arborescence du site web mobile



api : contient les services web de l'API RESTful appelée par l'application Android.

controleurs : contient tous les contrôleurs (codés en PHP uniquement - pas de HTML ici) ;
les contrôleurs utilisent le modèle et chargent les vues à envoyer à l'utilisateur.

downloads : contient les fichiers à télécharger (l'application Android).

images : contient les icônes de l'application (fichier favicon, ...).

modele : contient les classes métiers (**Point, PointDeTrace, Trace, Utilisateur**), les classes techniques (**Outils** et **DAO**), leurs tests ainsi que les paramètres de connexion ;
seule la classe **DAO** a accès à la base de données, c'est donc le seul fichier ou l'on peut trouver des requêtes **SQL**.

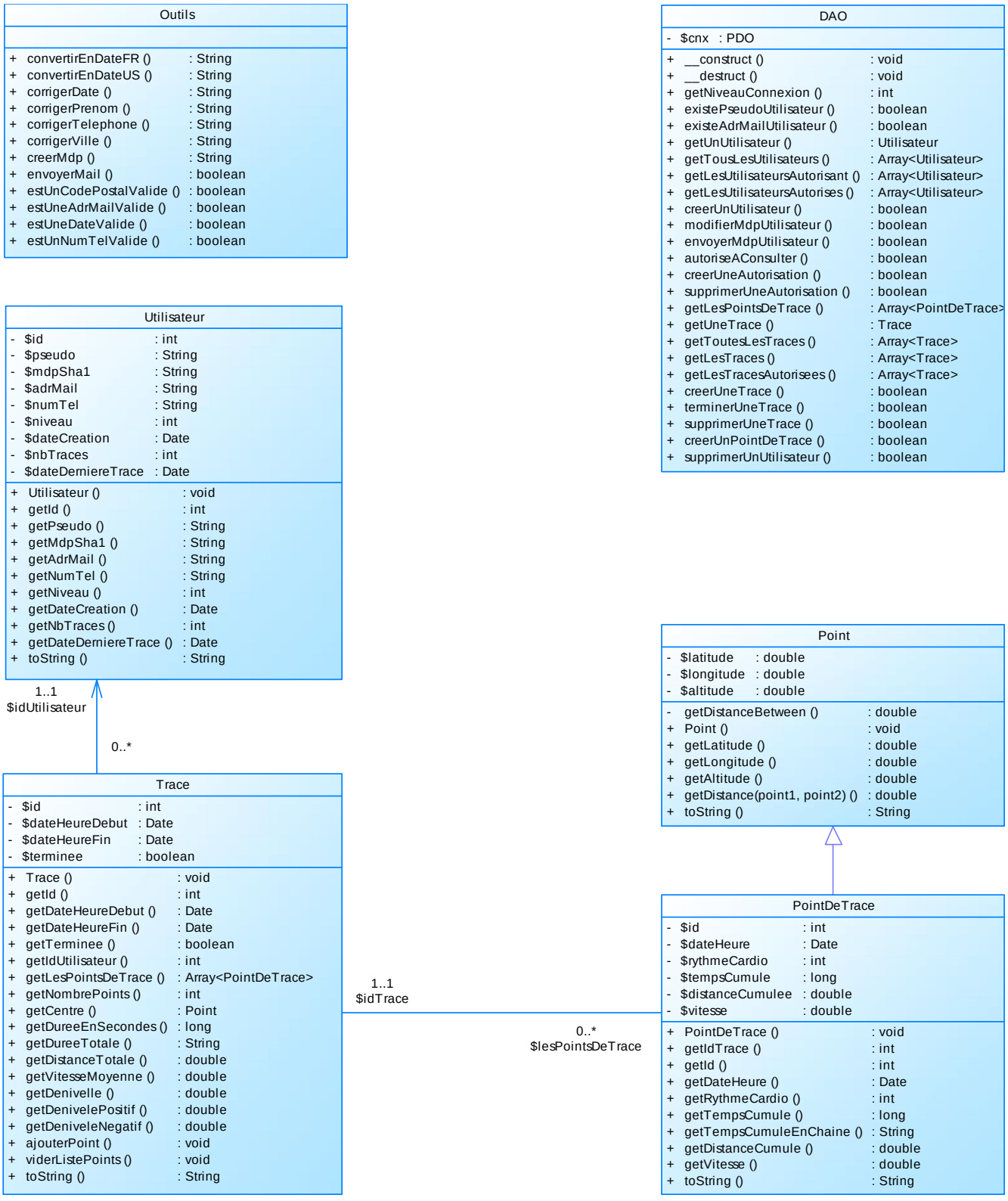
vues : contient toutes les vues et le fichier **head.php** ; elles sont codées essentiellement en **HTML5**, avec quelques séquences courtes en **PHP** pour incorporer des valeurs préparées par les contrôleurs.

index.php : c'est la page centrale, par laquelle passe toute action ; en fonction de l'action choisie, elle appelle le contrôleur chargé du traitement de l'action.

Mobile_Detect.php : module PHP permettant de détecter le type du terminal appelant.

5-4 Le modèle PHP

- La classe **DAO** (Data Access Object) fournit toutes les méthodes utiles pour accéder à la base de données **tracegps** ; elle utilise les classes **PDO** et **PDOStatement**.
- La classe **Outils** fournit des méthodes statiques offrant divers services utilitaires.



La classe **DAO** est la seule à accéder à la base de données, c'est donc la seule classe pouvant contenir des requêtes SQL.

Le dossier **modele** contient également les **tests** des classes.
Dans le cas de la classe **DAO**, le test est à adapter selon le contenu de la base de données.

5-5 La page index.php

C'est la page centrale, par laquelle passe toute demande d'action.
En fonction de l'action choisie, elle appelle le contrôleur chargé du traitement de l'action.

Voici son code complet :

```php
<?php
// Projet TraceGPS - version web mobile
// fichier : index.php
// Rôle : analyser toutes les demandes (appels de page ou traitements de formulaires) et activer le contrôleur chargé de traiter l'action demandée
// Dernière mise à jour : 01/11/2021 par dP

// Ce fichier est appelé par tous les liens internes, et par la validation de tous les formulaires
// il est appelé avec un paramètre action qui peut prendre les valeurs suivantes :

//    index.php?action=Connecter :                    pour afficher la page de connexion
//    index.php?action=CreerUtilisateur :             pour afficher la page de création de compte
//    index.php?action=DemanderMdp :                  pour afficher la page "mot de passe oublié"
//    index.php?action=Menu :                         pour afficher le menu
//    index.php?action=ConsulterListeUtilisateurs :pour afficher la page de consultation des utilisateurs
//    index.php?action=ChangerDeMdp :                 pour afficher la page de changement de mot de passe
//    index.php?action=DemanderAutorisation :         pour afficher la page de demande d'autorisation
//    index.php?action=RetirerAutorisation :          pour afficher la page de suppression d'autorisation
//    index.php?action=ConsulterParcours :            pour afficher la page de consultation de parcours
//    index.php?action=SupprimerParcours :            pour afficher la page de suppression de parcours
//    index.php?action=TesterGeolocalisation :        pour afficher la page de test de la géolocalisation

// il faut être administrateur pour les 2 actions suivantes :
//    index.php?action=CreerAdministrateur :          pour afficher la page de création d'un administrateur
//    index.php?action=SupprimerUtilisateur :         pour afficher la page de suppression d'un utilisateur

session_start();      // permet d'utiliser des variables de session

// inclusion des paramètres de l'application
include_once ('modele/parametres.localhost.php');

// choix des styles graphiques
$version = "1.4.5";         // choix de la version de JQuery Mobile (voir fichier head.php) : 1.2.0,  1.2.1,  1.3.2,  1.4.5
$themeNormal = "a";      // thème de base
$themeProbleme = "b";  // thème utilisé pour afficher un message en cas de problème
$transition ="flip";          // transition lors des changements de page (pop, flip, fade, turn, flow, slidefade, slide, slideup, slidedown)

// on vérifie le paramètre action de l'URL
if ( ! isset ($_GET['action']) == true)  $action = '';  else   $action = $_GET['action'];

// lors d'une première connexion, ou après une déconnexion, on initialise à vide les variables de session
if ($action == '' || $action == 'Deconnecter')
{    unset ($_SESSION['pseudo']);
     unset ($_SESSION['mdp']);
     unset ($_SESSION['niveauConnexion']);
     unset ($_SESSION['afficherMdp']);
}

// tests des variables de session
if ( ! isset ($_SESSION['pseudo']) == true)  $pseudo = '';
     else  $pseudo = $_SESSION['pseudo'];
if ( ! isset ($_SESSION['mdp']) == true)  $mdp = '';
     else  $mdp = $_SESSION['mdp'];
if ( ! isset ($_SESSION['niveauConnexion']) == true)  $niveauConnexion = 0;
     else  $niveauConnexion = $_SESSION['niveauConnexion'];

// pour mémoriser le choix d'afficher en clair (ou pas) le mot de passe :
```

```
if ( isset ($_SESSION['afficherMdp']) == false)  $afficherMdp = 'off';
    else  $afficherMdp = $_SESSION['afficherMdp'];

// si l'utilisateur n'est pas encore identifié, il sera automatiquement redirigé vers le contrôleur d'authentification
// (sauf s'il veut créer un compte ou bien se faire envoyer son mot de passe qu'il a oublié)
if ($pseudo == '' && $action != 'CreerUtilisateur' && $action != 'DemanderMdp')
    $action = 'Connecter';

switch($action){
    case 'Connecter': {
        include_once ('controleurs/CtrlConnecter.php'); break;
    }
    case 'CreerUtilisateur': {
       include_once ('controleurs/CtrlCreerUtilisateur.php'); break;
    }
    case 'DemanderMdp': {
       include_once ('controleurs/CtrlDemanderMdp.php'); break;
    }
    case 'Menu': {
        include_once ('controleurs/CtrlMenu.php'); break;
    }
    case 'ConsulterListeUtilisateurs': {
       include_once ('controleurs/CtrlConsulterListeUtilisateurs.php'); break;
    }
    case 'ChangerDeMdp': {
       include_once ('controleurs/CtrlChangerDeMdp.php'); break;
    }
    case 'TesterGeolocalisation': {
       include_once ('controleurs/CtrlTesterGeolocalisation.php'); break;
    }

    case 'ChoisirUtilisateurPourDemanderAutorisation': {
     include_once ('controleurs/CtrlChoisirUtilisateurPourDemanderAutorisation.php'); break;
    }
    case 'DemanderAutorisation': {
      include_once ('controleurs/CtrlDemanderAutorisation.php'); break;
    }


    case 'ChoisirUtilisateurPourRetirerAutorisation': {
        include_once ('controleurs/CtrlChoisirUtilisateurPourRetirerAutorisation.php'); break;
    }
    case 'RetirerAutorisation': {
      include_once ('controleurs/CtrlRetirerAutorisation.php'); break;
    }


    case 'DemarrerEnregistrementParcours': {
       include_once ('controleurs/CtrlDemarrerEnregistrementParcours.php'); break;
    }
    case 'EnvoyerPosition': {
       include_once ('controleurs/CtrlEnvoyerPosition.php'); break;
    }
    case 'ArreterEnregistrementParcours': {
       include_once ('controleurs/CtrlArreterEnregistrementParcours.php'); break;
    }


    case 'ChoisirUtilisateurPourConsulterParcours': {
```

```
        include_once ('controleurs/CtrlChoisirUtilisateurPourConsulterParcours.php'); break;
    }
    case 'ChoisirParcoursAConsulter': {
       include_once ('controleurs/CtrlChoisirParcoursAConsulter.php'); break;
    }
    case 'ConsulterParcours': {
       include_once ('controleurs/CtrlConsulterParcours.php'); break;
    }


    case 'ChoisirParcoursASupprimer': {
        include_once ('controleurs/CtrlChoisirParcoursASupprimer.php'); break;
    }
    case 'SupprimerParcours': {
      include_once ('controleurs/CtrlSupprimerParcours.php'); break;
    }


    case 'CreerAdministrateur': {
        include_once ('controleurs/CtrlCreerAdministrateur.php'); break;
    }


    case 'ChoisirUtilisateurASupprimer': {
       include_once ('controleurs/CtrlChoisirUtilisateurASupprimer.php'); break;
    }
    case 'SupprimerUtilisateur': {
        include_once ('controleurs/CtrlSupprimerUtilisateur.php'); break;
    }


    default : {
        // toute autre tentative est automatiquement redirigée vers le contrôleur d'authentification
        include_once ('controleurs/CtrlConnecter.php'); break;
    }
}
```

5-6 Les contrôleurs

Ce sont des fichiers ne comportant que du code **PHP** (**SQL** est réservé à la classes **DAO**, et **HTML** est réservé aux vues).

Chaque contrôleur effectue les traitements suivants :

- contrôle de l'autorisation d'accès avec les variables de session (l'absence de variable de session renvoie directement à la page index.php avec l'action "Connecter")
- contrôle de la demande d'action (les contrôles sont les mêmes que ceux qui ont été mis en place lors du développement des services web),
- exécution de l'action demandée si les conditions sont réunies,
- mise à jour de la vue actuelle en y affichant un message d'erreur ou un compte-rendu d'exécution dans une boîte de dialogue.

5-7 Les vues

Ce sont des fichiers codés essentiellement en **HTML5**, avec quelques séquences courtes en **PHP** pour incorporer des valeurs préparées par les contrôleurs.

L'entête des pages est réalisé par l'inclusion du fichier **head.php** présenté au point 2-2 de ce document. Cette inclusion de fichier permet de changer rapidement la version du framework **jQuery mobile**.

6- Description de quelques problèmes particuliers

6-1 Détection du type de terminal client

Le script **Mobile_detect.php** permet de détecter le type d'appareil du client.
L'adresse mobiledetect.net permet de télécharger ce script et fournit des exemples d'utilisation.

6-2 Détermination du type d'OS du terminal client

Le script **Mobile_detect.php** permet aussi de détecter le système d'explotation de l'appareil du client.

Les lignes de code suivantes permettent de détecter si le terminal mobile du client est sous Android (afin de lui afficher le bouton de téléchargement de l'application Android) :

```
// on teste si le terminal client est sous Android, pour lui proposer de télécharger l'application Android
require_once 'Mobile_Detect.php';
$detect = new Mobile_Detect;
if ( $detect->isAndroidOS() ) $OS = "Android"; else $OS = "autre";
```

6-3 Téléchargement du fichier apk (application Android)

Ce script permet au visiteur (muni d'un terminal Android) de télécharger l'application Android à partir du fichier **downloads/tracegps.apk** :

```
<?php
// Projet TraceGPS - version web mobile
// fichier : controleurs/CtrlTelechargerApk.php
// Rôle : traiter le téléchargement de l'application Android
// Dernière mise à jour : 01/11/2021 par dP

// Nom du fichier à télécharger : "tracegps.apk"
// Dossier contenant ce fichier : "downloads"

// désactive le temps max d'exécution
//set_time_limit(0);

$name = "tracegps.apk";
$filename = "../downloads/tracegps.apk";
$size = filesize($filename);

// source de cette version 2 : http://www.apprendre-php.com/tutoriels/tutoriel-25-forcer-le-telechargement-d-un-fichier.html
// désactive la mise en cache
header('Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0');
header('Pragma: no-cache');
header('Content-Tranfer-Encoding: none');
header('Expires: 0');
// prépare le téléchargement
header('Content-Type: application/octet-stream');
header('Content-disposition: attachment; filename="' . $name . '"');
header('Content-Length: ' . $size);                    // indique la taille du fichier à télécharger

// envoi le contenu du fichier
readfile($filename);
```

Sources Internet :

http://www.apprendre-php.com/tutoriels/tutoriel-25-forcer-le-telechargement-d-un-fichier.html

http://programmation-web.net/2012/04/comment-forcer-le-telechargement-dun-fichier-en-php/

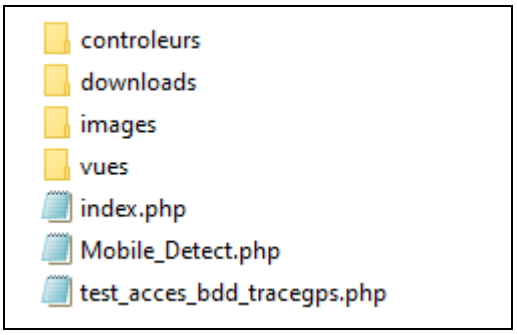
7- Installation et test de l'application fournie

7-1 Importation des fichiers fournis

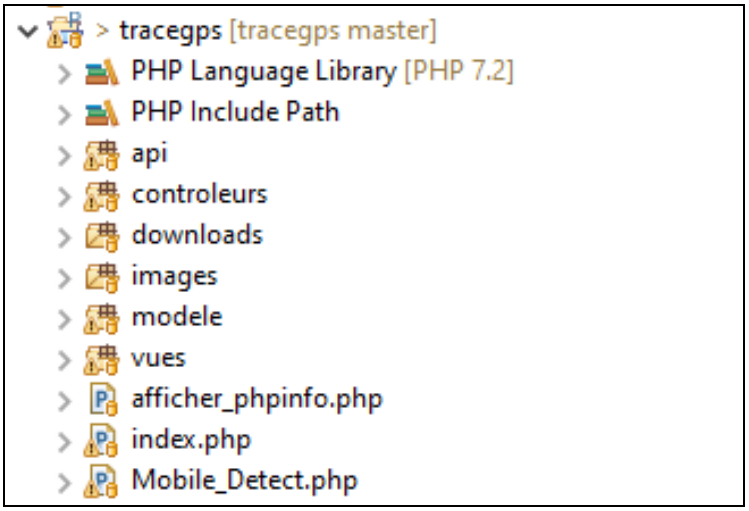
Ouvrez **Eclipse** avec votre **workspace PHP** et sélectionnez le projet **tracegps** qui contient pour l'instant 2 dossiers :

- **api** : contient les services web de l'API RESTful appelée par l'application Android.
- **modele** : contient les classes métiers (**Point, PointDeTrace, Trace, Utilisateur**), les classes techniques (**Outils** et **DAO**), leurs tests ainsi que les paramètres de connexion ;

Sélectionnez la racine du projet, et avec la commande **File / Import...** , importez les dossiers et fichiers fournis :



L'arborescence du site dans Eclipse doit devenir :



7-2 Obtention d'une clé API pour Google Maps en JavaScript

Pour créer une clé API pour Google Maps en JavaScript, allez à cette adresse :

https://developers.google.com/maps/documentation/javascript/get-api-key

7-3 Paramétrage

Ouvrez votre fichier **parametres.php** situé dans le dossier **modele**.

Remplacez son contenu par ceui-ci qui contient des paramètres utilisés par les pages PHP et modifiez bien les 2 paramètres indiqués :

```
<?php
// Projet TraceGPS - version web mobile
// fichier : modele/parametres.php
// Rôle : inclure les paramètres de l'application (hébergement en localhost)
// Dernière mise à jour : 01/11/2021 par dP

// paramètres de connexion ----------------------------------------------------------------------
global $PARAM_HOTE, $PARAM_PORT, $PARAM_BDD, $PARAM_USER, $PARAM_PWD;
$PARAM_HOTE = "localhost";              // si le sgbd est sur la même machine que le serveur php
$PARAM_PORT = "3306";                   // le port utilisé par le serveur MySql
$PARAM_BDD = "tracegps";                // nom de la base de données
$PARAM_USER = "tracegps";               // nom de l'utilisateur
$PARAM_PWD = "spgecart";                // son mot de passe

// Autres paramètres ----------------------------------------------------------------------------
global $TITRE_APPLI, $NOM_APPLI, $CLE_API, $FREQUENCE_AFFICHAGE, $ADR_MAIL_EMETTEUR,
$ADR_SERVICE_WEB;

// titre de l'application (en entête des vues)
$TITRE_APPLI = "TraceGPS";

// nom de l'application (en pied de page des vues)
$NOM_APPLI = "Suivi de parcours sportifs en extérieur";

// clé API pour utiliser Google Maps en JavaScript
$CLE_API = "AIzaSyCdcnX.....ZSZgCvGzRw";

// valeur de la fréquence de réactualisation de l'affichage (en secondes) d'un parcours
$FREQUENCE_AFFICHAGE = 60;                          // 60 sec ou 1 mn

// adresse de l'émetteur lors d'un envoi de courriel
$ADR_MAIL_EMETTEUR = "delasalle.sio.crib@gmail.com";

// adresse de l'API web en localhost ------------------------------------------------------------
$ADR_SERVICE_WEB = "http://localhost/ws-php-VotreNom/tracegps/api/";
// adresse de l'API web chez OVH ----------------------------------------------------------------
//$ADR_SERVICE_WEB = "http://sio.lyceedelasalle.fr/tracegps/api/";

// ATTENTION : on ne met pas de balise de fin de script pour ne pas prendre le risque
// d'enregistrer d'espaces après la balise de fin de script !!!!!!!!!!!!
```

Collez ici votre clé API pour Google Maps en Javascript

Indiquez ici le nom de votre workspace

7-4 Test de l'application

Testez l'application dans un navigateur web, en appelant l'URL de la page index.php

En cas de problème, vérifiez bien le fichier des paramètres (et que votre serveur web est bien démarré).

8- Adaptations souhaitées

Avant d'envisager de modifier cette application, il est évident que vous devez prendre le temps d'analyser et de comprendre le code fourni.

8-1 Obligation de modifier le mot de passe s'il n'est pas assez fort

Cette modification ne concerne que les contrôleurs (aucune modification des vues).

Le contrôleur **CtrChangerDeMdp.php** gère le changement de mot de passe par l'utilisateur.
Actuellement, ce contrôleur vérifie seulement que le nouveau mot de passe contient au moins 8 caractères.
On veut ajouter une condition de validation : **les nouveaux mots de passe devront en plus comporter au moins 1 lettre minuscule, 1 lettre majuscule et 1 chiffre**.

8-1-1 Première modification

La première modification consiste à modifier le contrôleur pour intégrer cette condition de validation ; en cas de non-validation, le message retourné sera :

> ***Le mot de passe doit comporter au moins 8 caractères, dont au moins une lettre minuscule, une lettre majuscule et un chiffre !***

Vous avez déjà réalisé cette vérification dans les TP web réalisés en module, à l'aide d'une expression régulière. Il suffit donc de transposer la solution du TP.

Conseil : ajoutez une nouvelle fonction **estUnMdpValide($unMdp)** dans la classe **Outils.class.php** en vous inspirant des 4 fonctions similaires existantes.

8-1-2 Deuxième modification

La nouvelle règle que vous venez de mettre en place ne s'applique que lorsque l'utilisateur modifie son mot de passe, ce qui signifie que les anciens mots de passe (qui ne vérifiaient pas obligatoirement la nouvelle règle) continuent à être utilisables lors de la connexion.

On souhaite maintenant **obliger les utilisateurs à modifier leur mot de passe** si le mot de passe ne vérifie pas les nouvelles règles.

La solution choisie part du principe que dès que la connexion est validée, l'application affiche le menu (la page **index.php** active le contrôleur **CtrlMenu.php** qui ne fait rien d'autre que d'afficher la vue **VueMenu.php**).

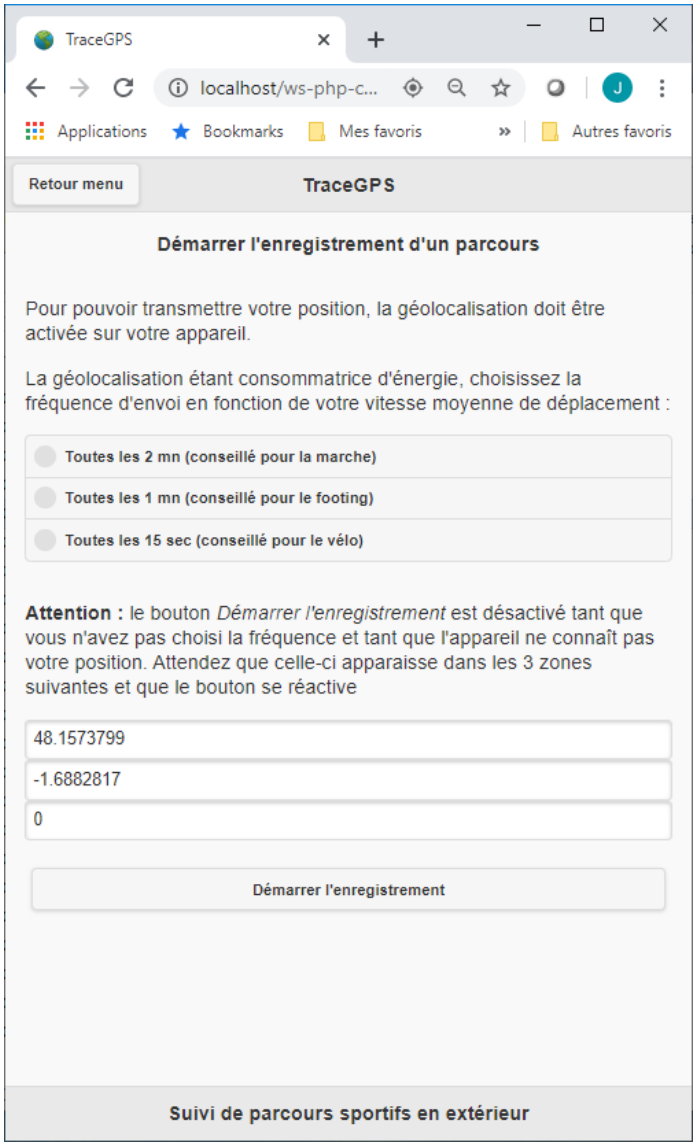
Vous allez compléter le contrôleur **CtrlMenu.php** pour tester la force du mot de passe utilisé pour la connexion (et mémorisé dans une variable de session) ; si la force est insuffisante, le contrôleur affichera la vue **VueChangerDeMdp.php** au lieu de la vue **VueMenu.php**.
Bien sûr, le contrôleur préparera les données dont la vue **VueMenu.php** a besoin, et notamment le message suivant :

> ***Pour des raisons de sécurité, nous vous invitons à changer votre mot de passe. Le nouveau mot de passe doit comporter au moins 8 caractères, dont au moins une lettre minuscule, une lettre majuscule et un chiffre !***

Si l'utilisateur refuse de faire le changement en choisissant l'option **Retour au menu**, le contrôleur **CtrlMenu.php** renouvellera le test du mot de passe et renverra automatiquement l'utilisateur sur la page de changement de mot de passe. Il sera donc obligé de le modifier !

8-2 Possibilité d'envoyer un courriel automatique lors du démarrage d'un parcours

Cette modification concerne le contrôleur **CtrlDemarrerEnregistrementParcours.php** et la vue **VueDemarrerEnregistrementParcours.php** :



L'objectif est ici de permettre à l'utilisateur qui démarre un parcours d'envoyer (ou non) un courriel automatique à tous les utilisateurs à qui il a donné l'autorisation de suivre ses parcours.

Pour cela, une case à cocher sera ajoutée à la vue (avec une étiquette) pour que l'utilisateur indique s'il souhaite envoyer un courriel automatique aux utilisateurs autorisés.

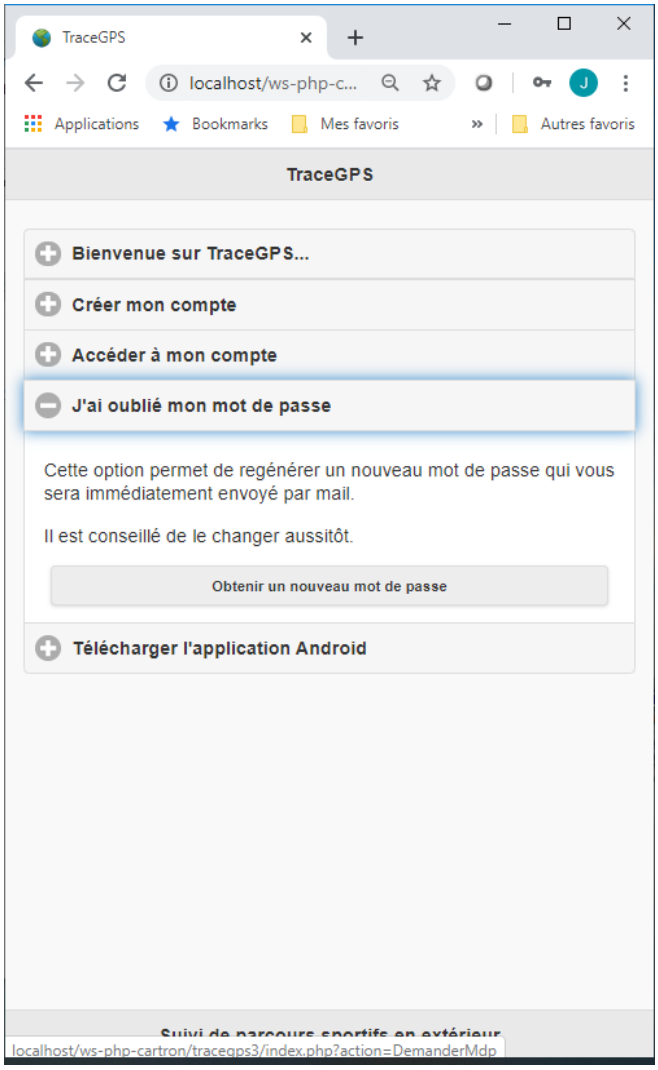
Le contrôleur se chargera de tester si la case est cochée, et assurera la sélection des utilisateurs autorisés et l'envoi d'un courriel de ce type :

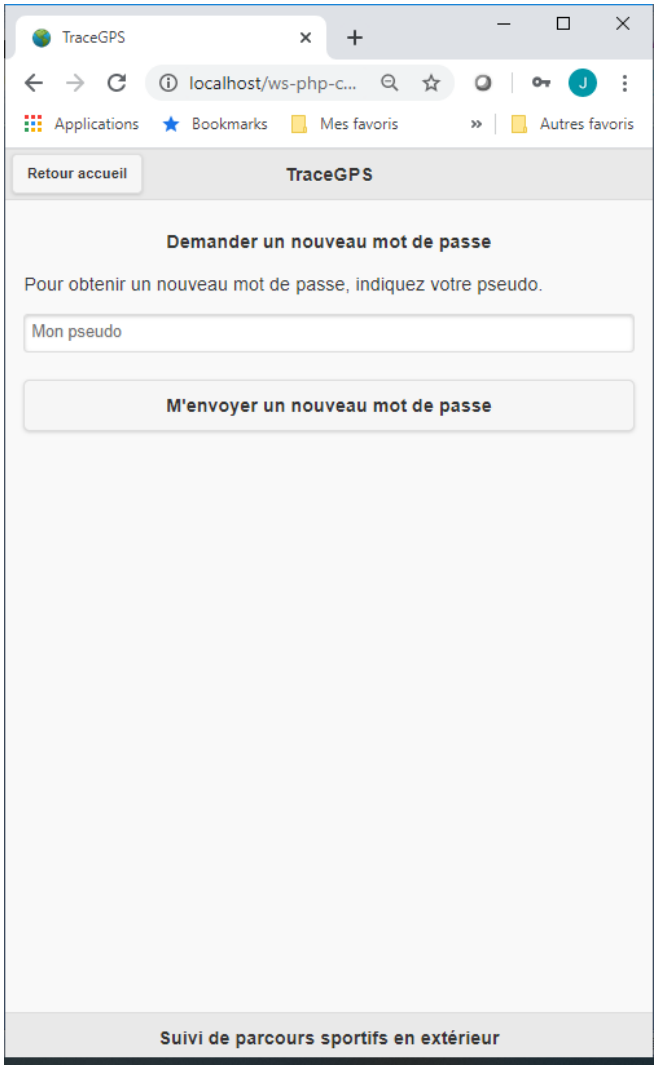
> ***Cher ou chère xxxxxxx,***
>
> ***Vous avez demandé à yyyyyyy l'autorisation de consulter ses parcours.***
> ***yyyyyyy vient de démarrer un nouveau parcours à hh:mm:ss.***
>
> ***Cordialement.***
> ***L'équipe TraceGPS.***

8-3 Amélioration de la fonction "J'ai oublié mon mot de passe"

La page d'accueil propose une fonctionnalité pour les utilisateurs qui ont oublié leur mot de passe.

Avant de demander au système de générer un nouveau mot de passe (qui sera envoyé par courriel), l'utilisateur doit indiquer son pseudo :





Les pseudos sont publics et consultables facilement, ce qui permet à des utilisateurs mal intentionnés de redemander des nouveaux mots de passe en utilisant des pseudos récupérés.

Pour limiter ce risque, vous allez compléter la vue **VueDemanderMdp.php** pour y ajouter la saisie de l'adresse mail en plus du pseudo. Vous prévoirez bien sûr d'utiliser un **pattern** pour vérifier que l'adresse mail semble correcte.

Puis vous compléterez le contrôleur **CtrlDemanderMdp.php** pour vérifier que le pseudo et l'adresse mail saisis correspondent bien au même utilisateur. Vous prévoirez bien sûr un message d'erreur explicatif (et sans faute d'orthographe) si les données ne correspondent pas.

Le nouveau mot de passe ne sera bien sûr regénéré que si les 2 données existent et correspondent bien au même utilisateur.

Vous ferez bien sûr les tests.