

Rapport personnel

Date de création : 02.02.2023  
Version 1 du 02.02.2023

Damian Piller

|  |  |  |
| --- | --- | --- |
|  |  | Module du 02.02.2023 au 09.03.2023 |

Table des matières

[1 Introduction 1](#_Toc127444680)

[2 Analyse 1](#_Toc127444681)

[2.1 Présentation du projet 1](#_Toc127444682)

[2.2 Maquette 2](#_Toc127444683)

[2.3 Use Case 3](#_Toc127444684)

[2.4 Schéma entite relations 3](#_Toc127444685)

[2.5 Diagramme d’activite 4](#_Toc127444686)

[3 Conception 5](#_Toc127444687)

[3.1 Partie Client 5](#_Toc127444688)

[3.2 Partie Serveur 5](#_Toc127444689)

[3.3 Sequence system 5](#_Toc127444690)

[4 Réalisation 5](#_Toc127444691)

[5 Conclusion 5](#_Toc127444692)

[6 Annexe 5](#_Toc127444693)

[6.1 refSheet PHP 5](#_Toc127444694)

[6.1.1 Including PHP in a File 5](#_Toc127444695)

[6.1.2 Comments 5](#_Toc127444696)

[6.1.3 Outputting Data 5](#_Toc127444697)

[6.1.4 Writing PHP Functions 5](#_Toc127444698)

[6.1.5 Variables and Constants 6](#_Toc127444699)

[a) Defining Variables 6](#_Toc127444700)

[b) Types of Data 6](#_Toc127444701)

[c) Variable Scope 7](#_Toc127444702)

[d) Predefined Variables 7](#_Toc127444703)

[e) Constants 7](#_Toc127444704)

[6.1.6 PHP Arrays – Grouped Values 8](#_Toc127444705)

[a) Declaring an Array in PHP 8](#_Toc127444706)

[6.1.7 PHP Strings 8](#_Toc127444707)

[a) Defining Strings 8](#_Toc127444708)

[b) Escape Characters 8](#_Toc127444709)

[6.1.8 Arithmetic Operators 9](#_Toc127444710)

[a) Addition 9](#_Toc127444711)

[b) Subtraction 9](#_Toc127444712)

[c) Multiplication 9](#_Toc127444713)

[d) Division 10](#_Toc127444714)

[e) Modulus 10](#_Toc127444715)

[f) Exponentiation 10](#_Toc127444716)

[6.1.9 PHP Assignment Operators 10](#_Toc127444717)

[a) x = y 10](#_Toc127444718)

[b) x += y 10](#_Toc127444719)

[c) x -= y 10](#_Toc127444720)

[d) x \*= y 10](#_Toc127444721)

[e) x /= y 10](#_Toc127444722)

[f) x %= y 10](#_Toc127444723)

[6.2 Exercice 1 10](#_Toc127444724)

[6.2.1 Partie 1 : Les bases du PHP 11](#_Toc127444725)

[6.2.2 Partie 2 : Exécuter du code PHP sur WAMP 11](#_Toc127444726)

[a) Informations 11](#_Toc127444727)

[b) Tâches 12](#_Toc127444728)

[6.2.3 Partie 3 : Transmettre des données de page en page 12](#_Toc127444729)

[6.2.4 Partie 4: Trouver et corriger les erreurs de mon code PHP 13](#_Toc127444730)

[6.3 Exercice 2 14](#_Toc127444731)

[6.4 Exercice 3 14](#_Toc127444732)

[6.5 Exercice 4 15](#_Toc127444733)

[6.6 Exercice 5 16](#_Toc127444734)

[6.6.1 wrk.php 17](#_Toc127444735)

[6.6.2 equips.php 17](#_Toc127444736)

[6.6.3 ctrl.php 17](#_Toc127444737)

[6.7 Exercice 6 18](#_Toc127444738)

[6.8 Exercice 7 19](#_Toc127444739)

[6.9 Exercice 8 22](#_Toc127444740)

[6.10 Exercice 9 25](#_Toc127444741)

# Introduction

Voici les objectifs que je dois acquérir pendant ce module :

* Analyser les exigences d’une application Web et de la base de données, respectivement des éléments de données à lier, définir et documenter la technique de liaison.
* Identifier les informations importantes de protection et de sécurité en tenant compte de la protection des données, et définir les mesures.
* Réaliser l’intégration de l’application Web avec la base de données, respectivement aux éléments de données, en prêtant attention aux transactions, à la protection et la sécurité des données.
* Mettre en œuvre les souhaits de modifications conformément au déroulement prescrit des modifications.
* Définir et exécuter la procédure de test et de remise, la documenter dans un procès-verbal de tests. Si nécessaire, entreprendre les corrections.

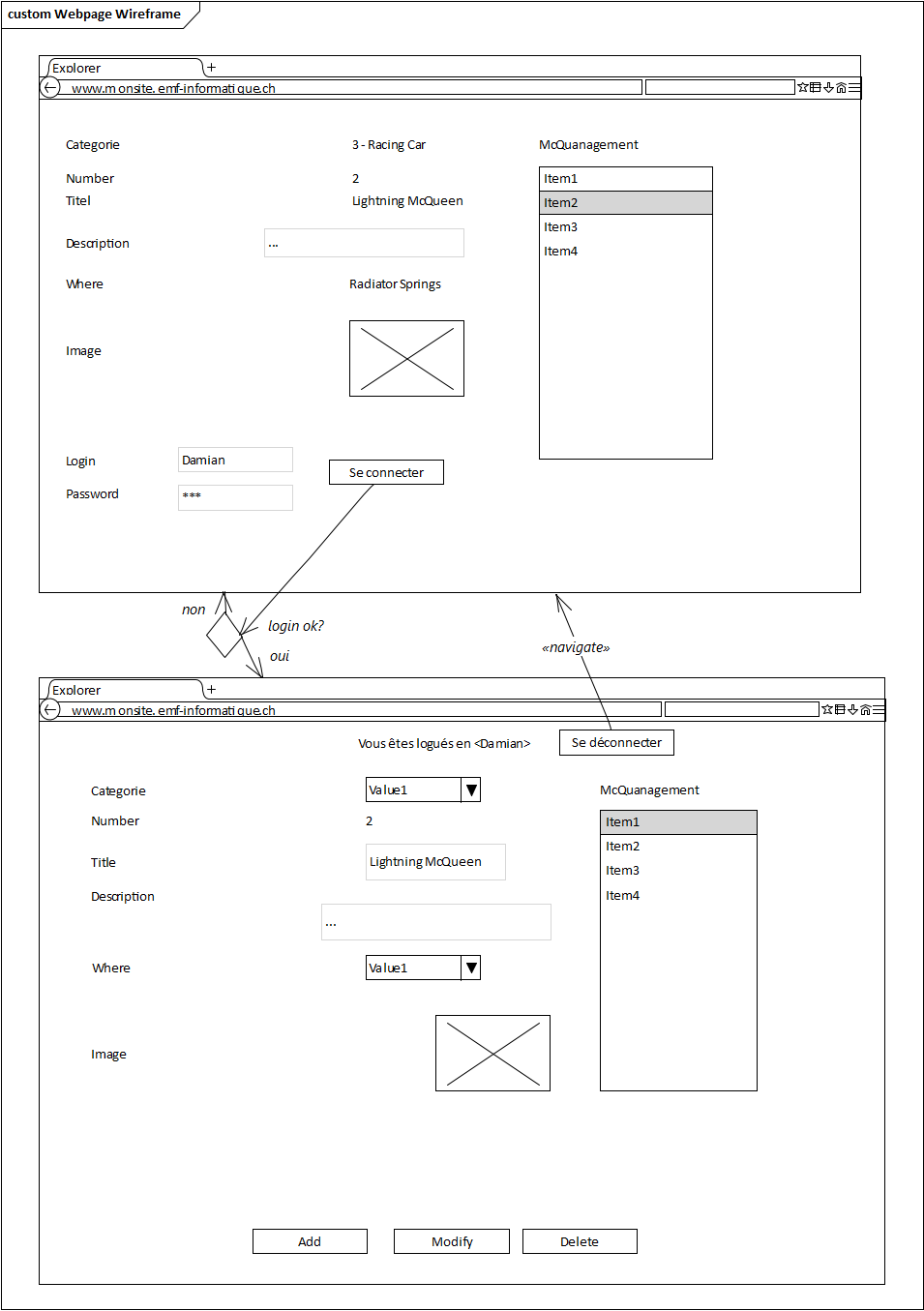


# Analyse

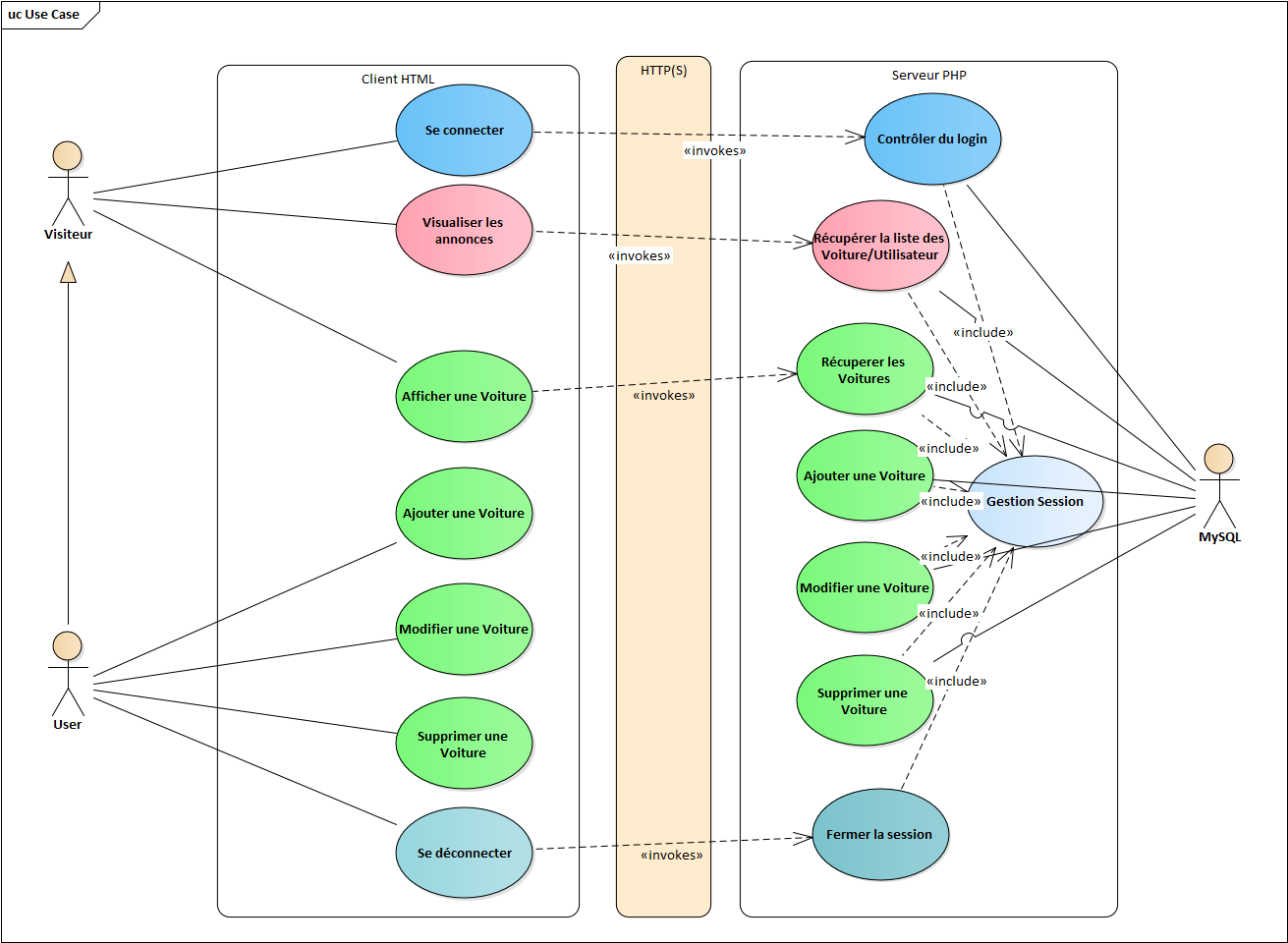
## Présentation du projet

Dans mon projet, il y a la possibilité de voir les différentes voitures de la série de films Cars. Ces différentes voitures sont affichées dans une liste. Dans cette liste, on peut choisir une voiture et voir les informations la concernant. Par exemple, une photo de chaque voiture, son origine, une brève description, son nom, la catégorie à laquelle elle appartient, etc. En outre, il est possible de se connecter avec un login et de modifier les entrées. On ne peut pas tout modifier, mais seulement la catégorie, le nom et l'origine de la voiture. Mon application s'appelle "McQuanagement".

## Maquette



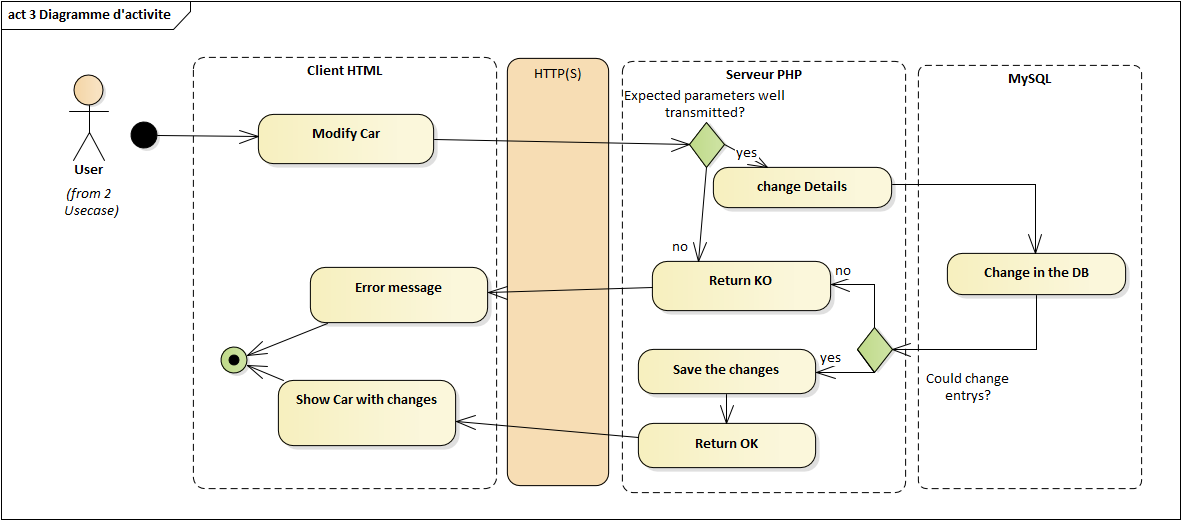
## Use Case



## Schéma entité relations

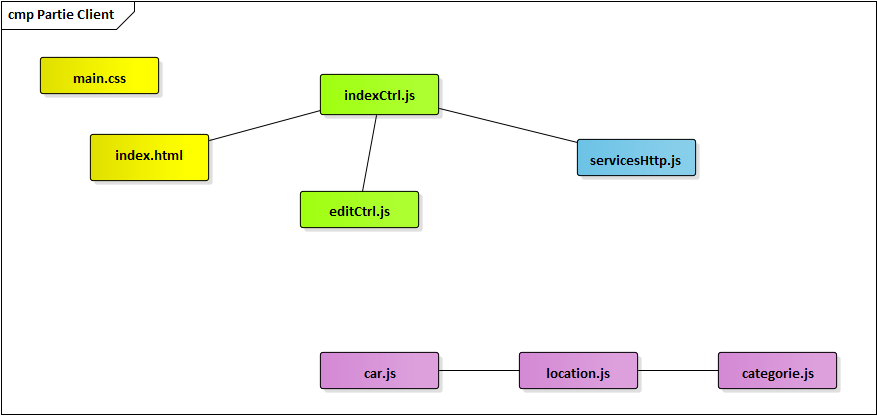


## Diagramme d’activite

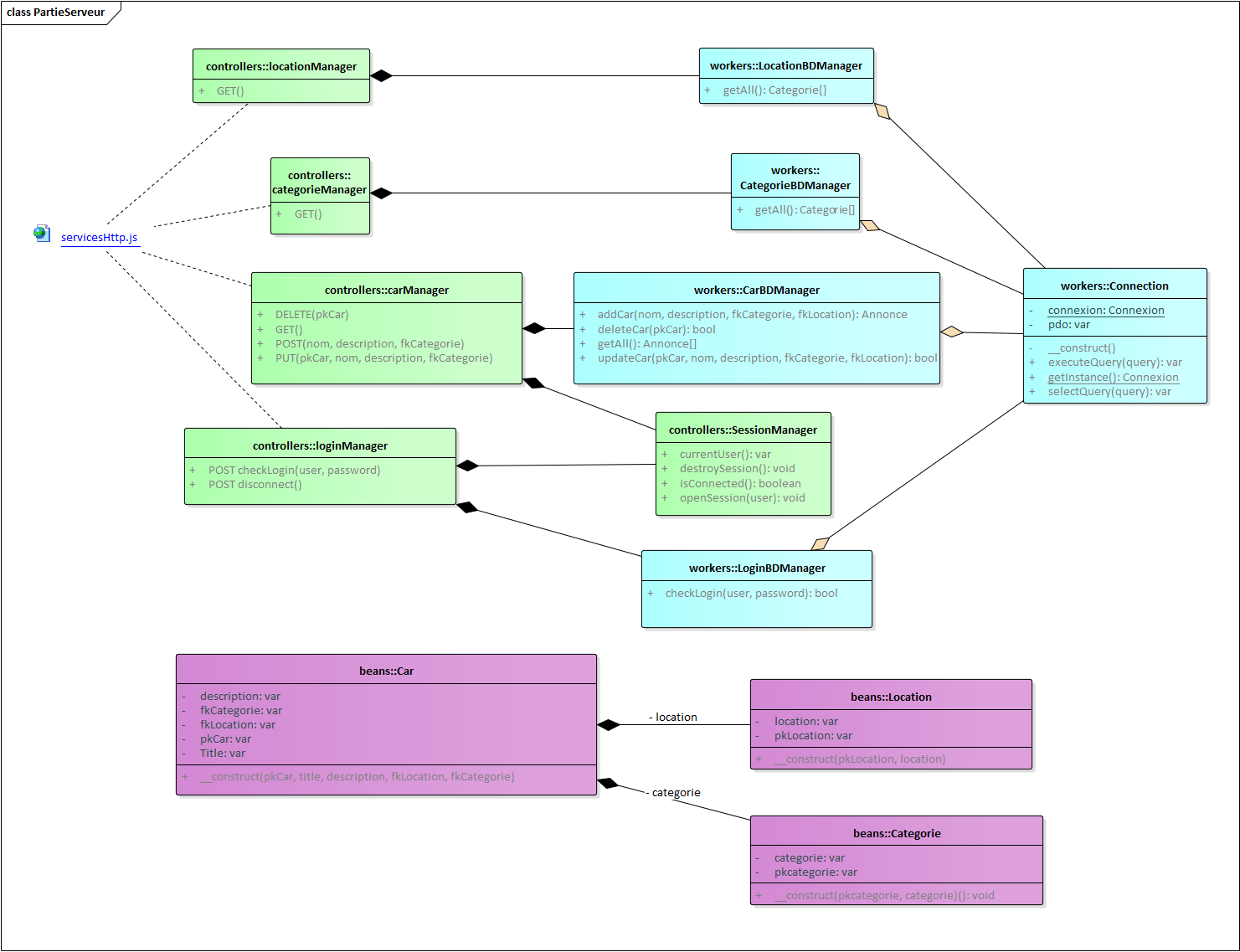


# Conception

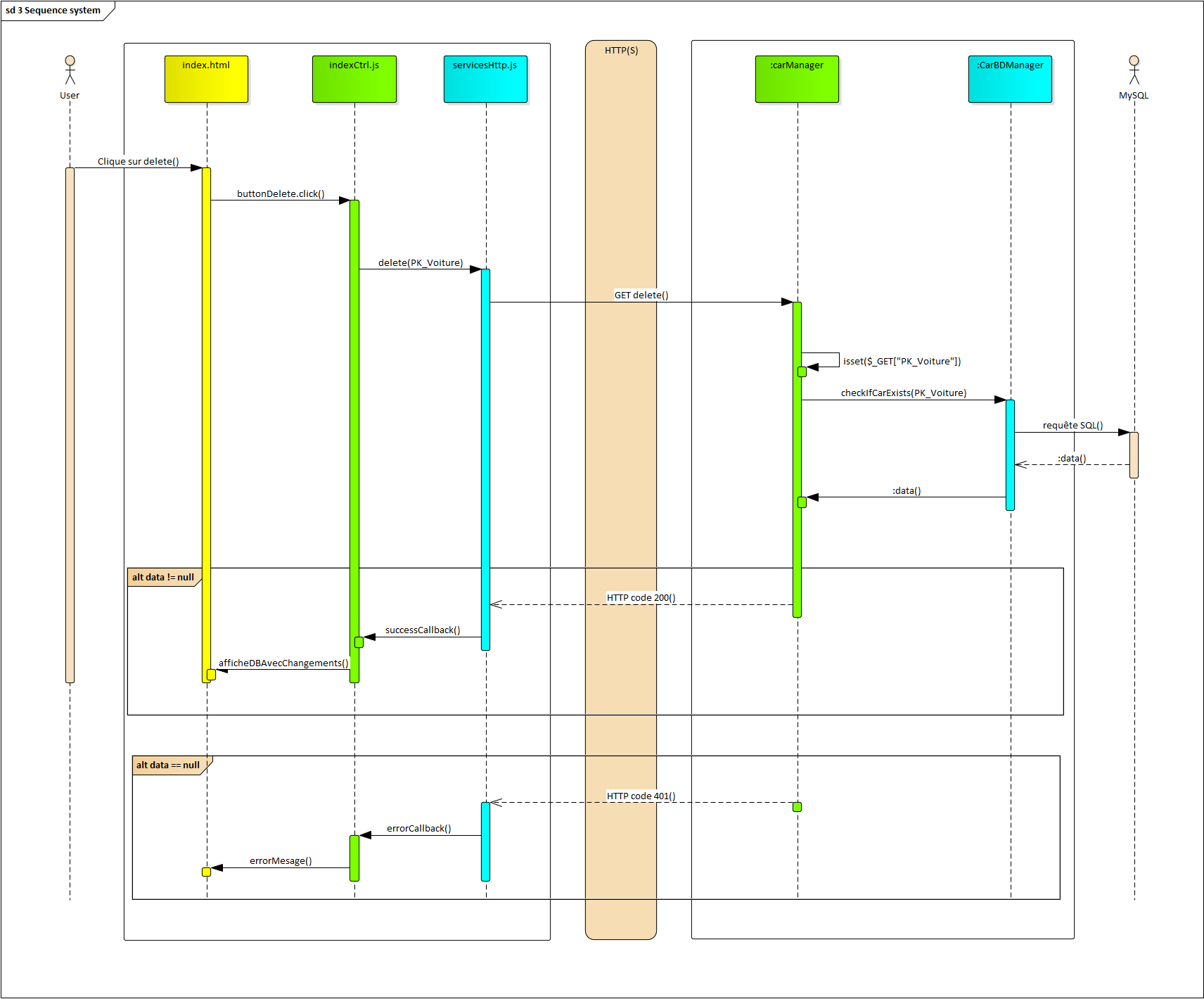
## Partie Client



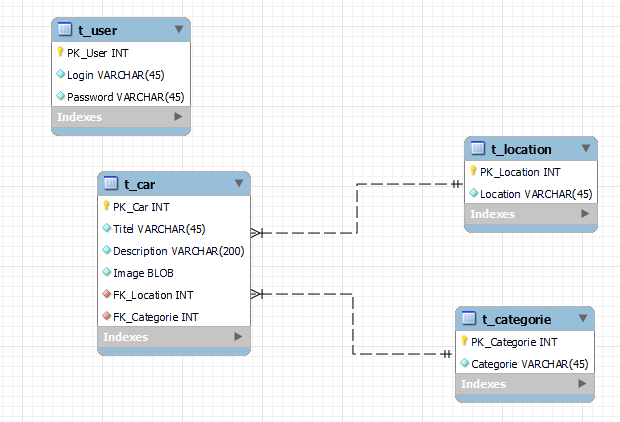
## Partie Serveur



## Diagramme de séquence interaction



## Schéma relationnel



## Conception de test

# Réalisation

# Conclusion

# Annexe

## refSheet PHP

### Including PHP in a File

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Comments

* // - Comments only on one line
* # - Also for one line
* /\*…\*/ - Multi line comment

### Outputting Data

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Writing PHP Functions

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### Variables and Constants

#### Defining Variables

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

#### Types of Data

* Integers
  + Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
* Floats
  + Ein Bild, das Text enthält.

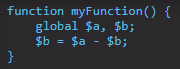
    Automatisch generierte Beschreibung
* Strings
  + Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
* Boolean values
  + True or False
* Arrays
  + Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
* Object
  + Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
* Resources – These are references to functions and resources outside of PHP
* NULL

#### Variable Scope



#### Predefined Variables

* $GLOBALS – Used to access global variables from anywhere inside a PHP script
* $\_SERVER – Contains information about the locations of headers, paths and scripts
* $\_GET – Can collect data that was sent in the URL or submitted in an HTML form
* $\_POST – Used to gather data from an HTML form and to pass variables
* $\_REQUEST – Also collects data after submitting an HTML form

#### Constants



* \_\_LINE\_\_ - Denotes the number of the current line in a file
* \_\_FILE\_\_ - Is the full path and filename of the file
* \_\_DIR\_\_ - The directory of the file
* \_\_FUNCTION\_\_ - Name of the function
* \_\_CLASS\_\_ - Class name, includes the namespace it was declared in
* \_\_TRAIT\_\_ - The trait name, also includes the namespace
* \_\_METHOD\_\_ - The class methode name
* \_\_NAMESPACE\_\_ - Name of the current namespace

### PHP Arrays – Grouped Values

#### Declaring an Array in PHP

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

### PHP Strings

#### Defining Strings

* Single quotes - This is the simplest way. Just wrap your text in ' markers and PHP will handle it as a string
* Double quotes - As an alternative you can use ". When you do, it’s possible to use the escape characters below to display special characters
* heredoc - Begin a string with <<< and an identifier, then put the string in a new line. Close it in another line by repeating the identifier. heredoc behaves like double-quoted strings
* nowdoc - Is what heredoc is for double-quoted strings but for single quotes. It works the same way and eliminates the need for escape characters

#### Escape Characters

* \n – Linefeed
  + Ein Bild, das Text, Gerät, Anzeige enthält.

    Automatisch generierte Beschreibung
* \r - Carriage return
  + Ein Bild, das Text, Gerät, Anzeige, Messanzeige enthält.

    Automatisch generierte Beschreibung
* \t - Horizontal tab
  + Ein Bild, das Text, Gerät, Anzeige, Messanzeige enthält.

    Automatisch generierte Beschreibung
* \v - Vertical tab
  + Ein Bild, das Text, Gerät, Anzeige, Messanzeige enthält.

    Automatisch generierte Beschreibung
* \e – Escape
  + Ein Bild, das Text, Gerät, Anzeige, Messanzeige enthält.

    Automatisch generierte Beschreibung
* \f - Form feed
  + Ein Bild, das Text, Gerät, Anzeige, Messanzeige enthält.

    Automatisch generierte Beschreibung
* \\ - Backslash
  + Ein Bild, das Text, Gerät, Anzeige, Messanzeige enthält.

    Automatisch generierte Beschreibung
* \$ - Dollar sign
  + Ein Bild, das Text, Gerät, Anzeige, Messanzeige enthält.

    Automatisch generierte Beschreibung
* /’ – Single quote
  + Ein Bild, das Text enthält.

    Automatisch generierte Beschreibung
* \’’ – Double quote
  + Ein Bild, das Text, Gerät, Anzeige, Messanzeige enthält.

    Automatisch generierte Beschreibung3
* \[0-7]{1,3} – Character in octal notation
* \x[0-9A-Fa-f]{1,2} – Character in hexadecimal notation
* \u{[0-9A-Fa-f]+} – String as UTF-8 representation

### Arithmetic Operators

#### Addition

Sum of $x and $y



#### Subtraction

Difference of $x and $y



#### Multiplication

Product of $x and $y



#### Division

Quotient of $x and $y



#### Modulus

The remainder of $x divided by $y



#### Exponentiation

Result of raising $x to the $y’th power



### PHP Assignment Operators

#### x = y

The left operand gets set to the value of the expression on the right

#### x += y

Addition

#### x -= y

Substraction

#### x \*= y

Multiplication

#### x /= y

Division

#### x %= y

Modulus

## Exercice 1

### Partie 1 : Les bases du PHP

*Veuillez chercher les réponses dans le chapitre "Écrivez votre premier script"*

1. A quoi sert la balise echo dans un fichier php ?

Elle permet d'insérer du texte dans une page web

1. A quoi sert le point-virgule dans un fichier php ?

C'est la fin d'une ligne

*Veuillez chercher les réponses dans les chapitres "les variables", "les conditions" et "les boucles".*

1. Par quel caractère commence une variable dans un fichier php ?

Par un "$"

1. Trouvez les erreurs dans le code php suivant :

|  |
| --- |
| <?php  $agePierre = 17;  $difference = 3;  $ageMarie = $agePierre + $difference;    echo 'Pierre a ' .$agePierre ' ans';  echo 'Marie a ' .$ageMarie ' ans';  ?> |

1. Veuillez indiquer ce qu’affichera le code suivant :

|  |
| --- |
| <?php  calculTest(5);  function calculTest($nombre)  {  for ($nombreIterations = 1 ; $nombreIterations <= $nombre ; $nombreIterations++)  {  echo 'passage ' .$nombreIterations. ' sur ' .$nombre. '<br/>' ;  }  }  ?> |

Affichage :

passage 1 sur 5

passage 2 sur 5

passage 3 sur 5

passage 4 sur 5

passage 5 sur 5

### Partie 2 : Exécuter du code PHP sur WAMP

#### Informations

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

Le logiciel WAMP (open source) permet d’installer, en une seule application, les éléments suivants :

* Un serveur Web (Apache)
* Une base de données (MySQL et/ou MariaDB)
* Un interpréteur PHP

Vous l’aurez compris, WAMP est donc l’acronyme de Windows Apache MySQL PHP.

Pour télécharger WAMP, rendez-vous sur le site [www.wampserver.com](http://www.wampserver.com). Il suffit ensuite d’exécuter l’installeur et vous devriez obtenir l’icône suivante dans la barre de notification :

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

ATTENTION : L’icône doit être verte, signe que tous les services ont pu démarrer correctement.

A partir de ce moment, vous pouvez placer vos fichiers PHP dans le répertoire C:\wamp64\www et l’accès au travers du serveur Web (avec l’adresse localhost) vous permettra de les exécuter.

Durant ce module, nous utiliserons donc WAMP pour le développement sur vos postes et, ensuite, vous utiliserez votre espace personnel emf-informatique pour héberger votre solution.

#### Tâches

1. Télécharger WAMP et installer-le sur votre PC.

1. Reprenez le code du point 5 pour en tester le fonctionnement sur votre installation de WAMP

### Partie 3 : Transmettre des données de page en page

*Veuillez chercher les réponses dans les chapitres "Ecoutez la requête de vos utilisateurs grâce aux URL" et "Administrez des formulaires de façon sécurisée"*

1. Voici un appel à une page PHP : <http://www.sitePerso.com/fichier.php?nom=Olivier>

Veuillez répondre aux questions suivantes :

* S’agit-il d’un POST ou d’un GET ? Quelle est la différence ?

C'est un GET car il y a un ? dans la URL. GET contient les donnes issues de l'URL et POST les donnes issues la forumulair.

* Comment (avec quel code ?) peut-on récupérer la valeur du paramètre « nom » dans le php ?

$\_GET('nom')

* Comment (avec quel code ?) peut-on tester si le paramètre « nom » a bien été donné ?

Isset($\_GET('nom'))

*Veuillez chercher les réponses dans le chapitre "Conservez des données grâce aux Sessions & Cookies"*

1. Voici 3 fichiers PHP :

|  |  |  |
| --- | --- | --- |
| fichier1.php | fichier2.php | fichier3.php |
| <?php  session\_start();  $\_SESSION['prenom'] = 'Jean';  $\_SESSION['nom'] = 'Dupont';  $\_SESSION['age'] = 24;  ?> | <?php  session\_start();  session\_destroy() ;  ?> | <?php  session\_start();  print\_r($\_SESSION);  ?> |

Que s’affichera-t-il si :

* L’utilisateur lance la page fichier1.php puis fichier3.php ?

[prenom] => Jean

[nom] => Dupont

[age] => 24

* L’utilisateur lance la page fichier1.php puis fichier2.php puis fichier3.php ?

rien

* L’utilisateur lance la page fichier2.php puis fichier1.php puis fichier3.php ?

[prenom] => Jean

[nom] => Dupont

[age] => 24

* L’utilisateur lance tout de suite le fichier3.php ?

rien

### Partie 4: Trouver et corriger les erreurs de mon code PHP

*Veuillez prendre note du chapitres "Au secours ! Mon script plante !" Pour corriger ce fichier.*

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Exercice 2

<!doctype html>

<html>

  <header>

    <link rel="stylesheet" type="text/css" href="stylesheets/main.css" />

</header>

  <body>

    <div id="conteneur">

      <h1>Les équipes de National League</h1>

      <table border= "1">

      <tr>

        <td>ID</td>

        <td>Club</td>

      </tr>

      <?php

        require('ctrl.php');

        $equipe = getEquipes();

        for($i = 0; $i < count($equipe); $i++){

          $j = $i + 1;

          echo "<tr><td> $j </td> <td> $equipe[$i] </td></tr>";

        }

      ?>

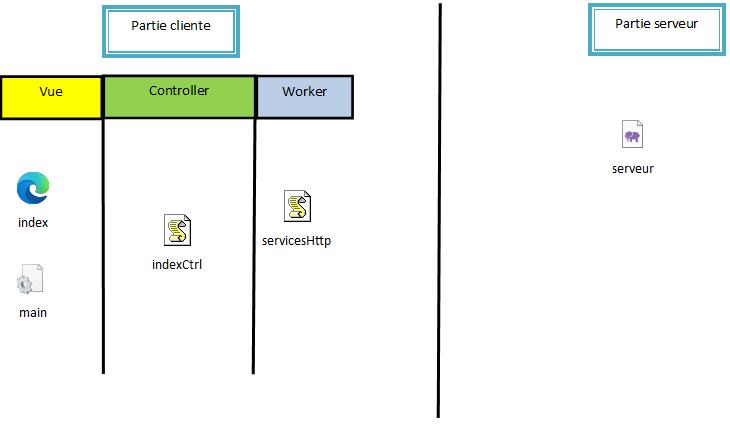
      </table>

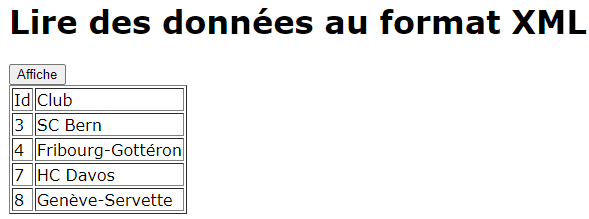
    </div>

  </body>

</html>

## Exercice 3





var BASE\_URL = "http://localhost/Exercice%201/client/php/serveur";

## Exercice 4

var cmbJoueurs = document.getElementById("cmbJoueurs");

    cmbJoueurs.options.length = 0;

    $(data).find("joueur").each(function() {

        var joueur = new Joueur();

        joueur.setNom($(this).find("nom").text());

        joueur.setPoints($(this).find("points").text());

        cmbJoueurs.options[cmbJoueurs.options.length] = new Option(joueur, JSON.stringify(joueur));

    });

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

## Exercice 5

|  |
| --- |
| <?php  class Membre  {  private $nom;  public $numero;  public function \_\_construct($nom, $numero)  {  $this->nom = $nom;  $this->numero = $numero;  }  public function getNom()  {  return $this->nom ;  }  }  ?> |

Le code suivant utilise cette classe :

|  |
| --- |
| <?php  include\_once('Membre')    $membre = new Membre('paul', 69);  $Nom = $membre->getNom();  $Numero = $membre->numero;    echo 'Un nouveau membre! Nom: ' $nom ', son âge: ' .$numero. '.';  ?> |

### wrk.php

<?php

class Wrk{

  public function getEquipesFromDB(){

    return array("Gotteron", "SC Bern", "Fribourg-Gottéron", "HC Davos");

  }

  public function ajouteCelluleHtml($id, $equipe){

    echo '<tr><td>' .$id. '</td><td>' .$equipe. '</td></tr>';

  }

}

?>

### equips.php

<?php

        require('ctrl.php');

        $ctrl = new Ctrl();

        $compteur = 1;

        foreach($ctrl->getEquipes() as $equipe){

          $ctrl->ajouteCellule($compteur, $equipe);

          $compteur = $compteur + 1;

        }

      ?>

### ctrl.php

<?php

include\_once('Wrk.php');

class Ctrl {

  public function \_\_construct(){

    $this->wrk = new Wrk;

  }

  public function getEquipes(){

    return $this->wrk->getEquipesFromDB();

  }

  public function ajouteCellule($id, $equipe){

    return $this->wrk->ajouteCelluleHtml($id, $equipe);

  }

}

?>

## Exercice 6

|  |
| --- |
| <?php  $bdd = new PDO('mysql:host=localhost;dbname=nomDB', 'root', 'pwd');  $reponse = $bdd->query('select pk\_jeux\_video, titre from jeux\_video');      while ($row = $response->fetch())  {    echo $row['titre'] . $row['pk\_jeux\_video'];      }  $reponse->closeCursor();  ?> |

Code pour connecter et afficher une DB

<?php

try

{

    // On se connecte à MySQL

    $mysqlClient = new PDO('mysql:host=localhost;dbname=jeux\_video;charset=utf8', 'root', '');

}

catch(Exception $e)

{

    // En cas d'erreur, on affiche un message et on arrête tout

        die('Erreur : '.$e->getMessage());

}

// Si tout va bien, on peut continuer

// On récupère tout le contenu de la table recipes

$sqlQuery = 'SELECT \* FROM jeux';

$jeuxStatement = $mysqlClient->prepare($sqlQuery);

$jeuxStatement->execute();

$jeux = $jeuxStatement->fetchAll();

// On affiche chaque recette une à une

foreach ($jeux as $jeu) {

?>

    <p><?php echo $jeu['titre']; ?></p>

<?php

}

?>

## Exercice 7

Dans cet exercice, il fallait faire une modification de l’exercice 4, mais cette fois les données n’était pas en XML mais stockés dans une BD. Voici le code du fichier php que j’ai ecrit :

<?php

require\_once('Wrk/configConnexion.php');

require\_once('Beans/Equipe.php');

require\_once('Beans/Joueur.php');

require\_once('Wrk/Wrk.php');

require\_once('Ctrl/Ctrl.php');

$ctrl = new Ctrl();

if($\_GET['action'] == "equipe")

{

    $ctrl->getEquipes();

}

if($\_GET['action'] == "joueur")

{

    if (isset($\_GET['equipeId']))

    {

        $ctrl->getJoueurs($\_GET['equipeId']);

    }

}

?>

Ensuite, il fallait créer un fichier contenant toutes les informations pour pouvoir accéder é la base de données (configConnexion.php) :

<?php

  //Informations de connexion à la base de données

  define('DB\_TYPE', 'mysql');

  define('DB\_HOST', 'localhost:3306');

  define('DB\_NAME', 'hockey\_stats');

  define('DB\_USER', 'root');

  define('DB\_PASS', '');

?>

Ensuite il fallait créer des Beans représentant les joueurs… :

<?php

class Joueur{

  function \_\_construct($nom, $points){

    $this->points = $points;

    $this->nom = $nom;

  }

  function getNom(){

    return $this->nom;

  }

  function getPoints(){

    return $this->points;

  }

  private $nom;

  private $points;

}

?>

… et un qui représente les équipes :

<?php

class Joueur{

  function \_\_construct($nom, $points){

    $this->points = $points;

    $this->nom = $nom;

  }

  function getNom(){

    return $this->nom;

  }

  function getPoints(){

    return $this->points;

  }

  private $nom;

  private $points;

}

?>

Puis il fallait créer le workeur, qui établit la connexion avec la BD et qui récupère les équipes et les joueurs de la BD :

<?php

class Wrk

{

  function \_\_construct()

  {

    try {

      $this->pdo = new PDO(DB\_TYPE . ':host=' . DB\_HOST . ';dbname=' . DB\_NAME, DB\_USER, DB\_PASS, array(

        PDO::MYSQL\_ATTR\_INIT\_COMMAND => 'SET NAMES utf8',

        PDO::ATTR\_PERSISTENT => true

      ));

    } catch (PDOException $e) {

      print "Erreur !: " . $e->getMessage() . "<br/>";

      die();

    }

  }

  function getEquipesFromDB()

  {

    $query = $this->pdo->prepare('select \* from t\_equipe');

    $query->execute();

    $rows = $query->fetchAll();

    $result = array();

    foreach($rows as $row){

      $equipe = new Equipe($row['PK\_equipe'], $row['Nom']);

      $result[] = $equipe;

    }

    $query->closeCursor();

    return $result;

  }

  function getJoueursFromDB($idEquipe){

    $query = $this->pdo->prepare('select \* from t\_joueur where fk\_equipe = :equipe');

    $query->execute(array('equipe'=>$idEquipe));

    $rows = $query->fetchAll();

    $result = array();

    foreach($rows as $row){

      $joueur = new Joueur($row['Nom'], $row['Points']);

      $result[] = $joueur;

    }

    $query->closeCursor();

    return $result;

  }

  private $pdo;

}

Pour la fin, il fallait créer le contrôleur, qui s’occupait de mettre en format XML les données extraites de la BD :

<?php

class Ctrl

{

  function getEquipes()

  {

    $equipes = $this->wrk->getEquipesFromDB();

    echo '<equipes>';

    foreach ($equipes as $equipe) {

      echo '<equipe>';

      echo '<id>';

      echo $equipe->getPk();

      echo '</id>';

      echo '<nom>';

      echo $equipe->getNom();

      echo '</nom>';

      echo '</equipe>';

    }

    echo '</equipes>';

  }

  function getJoueurs($idEquipe)

  {

    $joueurs = $this->wrk->getJoueursFromDB($idEquipe);

    echo '<joueurs>';

    foreach ($joueurs as $joueur) {

      echo '<joueur>';

      echo '<nom>';

      echo $joueur->getNom();

      echo '</nom>';

      echo '<points>';

      echo $joueur->getPoints();

      echo '</points>';

      echo '</joueur>';

    }

    echo '</joueurs>';

  }

  private $wrk;

}

L’application fonctionne exactement comme celle de l’exercice 4, il n’est donc pas nécessaire d’ajouter un screenshot montrant son exécution. Voici la structure des fichiers côté serveur :

Une image contenant texte

Description générée automatiquement

## Exercice 8

Cet exercice avait pour but de nous familiariser avec les sessions et les cookies, il fallait donc s’occuper de la partie serveur pour qu’elle gère les droits d’accès à des informations.

|  |  |
| --- | --- |
| Voici ce qu’il se passe lorsqu’on entre le faux mot de passe. | Une image contenant texte  Description générée automatiquement |
| Dès que l’on entre le bon mot de passe, on peut voir la liste du personnel. | Une image contenant texte  Description générée automatiquement |
| On peut même se déconnecter. | Une image contenant texte  Description générée automatiquement |

Il fallait s’occuper de la partie serveur, qui s’occupe de gérer les requêtes POST :

<?php

if ($\_SERVER['REQUEST\_METHOD'] == 'POST'){

    if($\_POST['action'] == "connect")

    {

        $pw = $\_POST["password"];

        session\_start();

        // doit controller que le password donné (en post) est bien 'emf'

        if ($pw === "emf") {

            // si c'est ok => enregistrement de "emf" dans une variable de session et écrire '<result>true</result>'

            $\_SESSION["logged"] = $pw;

            echo '<result>true</result>';

        } else {

            // si c'est faux => effacer la variable de session et écrire '<result>false</result>'

            session\_destroy();

            echo '<result>false</result>';

        }

    }

    if($\_POST['action'] == "disconnect")

    {

        // effacer la variable de session et écrire <result>true</result>

        session\_start();

        unset($\_SESSION["logged"]);

        session\_destroy();

        echo '<result>true</result>';

    }

}

if ($\_SERVER['REQUEST\_METHOD'] == 'GET'){

    if($\_GET['action'] == "getInfos")

    {

        session\_start();

        // tout d'abord contrôler que la variable de session 'logged' contient bien "emf"

        if (isset($\_SESSION["logged"])) {

            if ($\_SESSION["logged"] == "emf") {

                // si oui => retourner <users><user><name>Victor Legros</name><salaire>9876</salaire></user><user><name>Marinette Lachance</name><salaire>7540</salaire></user><user><name>Gustave Latuile</name><salaire>4369</salaire></user><user><name>Basile Ledisciple</name><salaire>2384</salaire></user></users>

                echo "<users><user><name>Victor Legros</name><salaire>9876</salaire></user><user><name>Marinette Lachance</name><salaire>7540</salaire></user><user><name>Gustave Latuile</name><salaire>4369</salaire></user><user><name>Basile Ledisciple</name><salaire>2384</salaire></user></users>";

            }

        } else {

            // si non => écrire <message>DROITS INSUFFISANTS</message>

            echo '<message>DROITS INSUFFISANTS</message>';

        }

    }

}

?>

## Exercice 9

Les erreurs que j’ai corrigées pour que ça marchait :

Dans le indexCtrl.js :

function chargerSkieursSuccess(data, text, jqXHR)

{

    var txt = '';

    $(data).find("skieur").each(function() {

        var skieur = new Skieur();

        skieur.setNom($(this).find("nom").text());

        skieur.setPosition($(this).find("position").text());

        txt = txt + "<tr><td>" + skieur.getPosition() + "</td><td>" + skieur.toString() + "</td></tr>";

    });

    document.getElementById("tableContent").innerHTML = txt;

}

Dans le servicesHttp.js :

var BASE\_URL = "http://localhost/Exercice%201/ex9\_donnee/serveur/";

Dans le configConnexion.php :

define('DB\_TYPE', 'mysql');

    define('DB\_HOST', 'localhost:3306');

    define('DB\_NAME', 'bd\_kitzbuehl');

    define('DB\_USER', 'root');

    define('DB\_PASS', '');

Dans le Pays.php :

 public function toXML()

    {

      $result = '<pays>';

      $result = $result . '<pk\_pays>'.$this->getPkPays().'</pk\_pays>';

      $result = $result . '<nom>'.$this->getNom().'</nom>';

      $result = $result . '</pays>';

      return $result;

    }

Et dans le Skieur.php :

public function \_\_construct($pkSkieur, $nom, $position) {

    $this->pk\_skieur = $pkSkieur;

    $this->nom = $nom;

    $this->position = $position;

  }

public function toXML()

  {

    $result = '<skieur>';

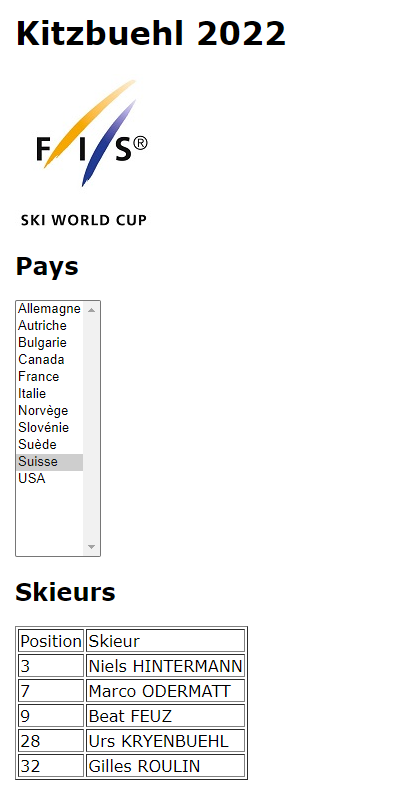
    $result = $result . '<nom>'.$this->getNom().'</nom>';

    $result = $result . '<position>'.$this->getPosition().'</position>';

    $result = $result . '</skieur>';

    return $result;

  }



Voici ou les différents fichiers se trouve :

