



IUT de Vélizy-Rambouillet

CAMPUS DE VÉLIZY-VILLACOUBLAY

*Durand Antonin*

*Jougla Maxime*

*Parciany Benjamin*

*Zehren William*

## Dossier de conception

---

## Table des matières

---

1. Introduction
2. Conception Livrable 1
  1. Conception architecturale
  2. Conception détaillée

## Introduction

Ce document est un dossier de conception divisé en plusieurs parties. Il sera constitué de la conception architecturale et détaillée de chaque livrable. Il aura pour but de donner de renseigner toutes les informations sur la conception du site ainsi que des diagrammes permettant de la représenter graphiquement.

Le premier livrable consiste en la réalisation des bases du site web, à savoir les pages principales. Il consiste également en la configuration du kit cluster hat, et en l'installation du site web sur ce dernier.

# Conception Livrable 1

## Conception architecturale

Le domaine du problème est le site web BlitzCalc, on décompose donc ce système en sous-systèmes.

On définit ci-dessous une conception architecturale du site web avec : le serveur web côté serveur, le navigateur web côté client et la base de données.

On utilise donc l'approche orientée objet pour repérer les différents objets du problème. On utilise également la vue composant-connecteur pour représenter le système ainsi que l'ensemble des composants du système d'un point de vue statique.

On introduit également le serveur, le navigateur web et la base de données.

Le composant serveur est dans le kit cluster hat et le navigateur est un composant présent dans les machines des utilisateurs. Le composant base de données quant à lui est installé sur le kit cluster hat.

Objet	Etat	Comportement
page_inscription.php		
page_connexion.php		
page_accueil.php		
page_accueil_user.php		
page_accueil_admin.php		
style.css		
Enum_fic_logs.php		
Enum_niveau_logger.php		
Enum_role_user.php		
Logger.php		
LoggerInstance.php		
Logging.php		
MYSQLDataManagement.php		
page_deconnexion.php		
traitement_connexion.php		
traitement_inscription.php		
traitement_profil.php		
User.php		
Utility.php		

Objet	Etat	Comportement
verif_identite_page_admin.php		
verif_identite_page_user_inscrit_only.php		
verif_identite_page_user.php		
index.html		
Serveur Web	A tous les objets .php, .html, .css, .json, a une interface	Ecoute les requêtes du navigateur
Navigateur Web		Lit les fichiers, exécute des requêtes
Base de données	a les tables Logging, Users, Weak_passwords	

A partir de ce tableau, on peut définir les composants suivants : "page\_accueil.php", "page\_connexion.php", "page\_inscription.php", "page\_accueil\_user.php", "page\_accueil\_admin.html", "style.css", "Enum\_fic\_logs.php", "Enum\_niveau\_logger.php", "Enum\_role\_user.php", "Enum\_role\_user.php", "Logger.php", "LoggerInstance.php", "Logging.php", "MySQLDataManagement.php", "page\_deconnexion.php", "Pagination.php", "traitement\_connexion.php", "traitement\_inscription.php", "traitement\_profil.php", "User.php", "Utility.php", "verif\_identite\_page\_admin.php", "verif\_identite\_page\_user\_inscrit\_only.php", "verif\_identite\_page\_user.php", "index.html".

Ces derniers sont des abstractions respectives des objets : "page\_accueil.php", "page\_connexion.php", "page\_inscription.php", "page\_accueil\_user.php", "page\_accueil\_admin.html", "style.css", "Enum\_fic\_logs.php", "Enum\_niveau\_logger.php", "Enum\_role\_user.php", "Enum\_role\_user.php", "Logger.php", "LoggerInstance.php", "Logging.php", "MySQLDataManagement.php", "page\_deconnexion.php", "Pagination.php", "traitement\_connexion.php", "traitement\_inscription.php", "traitement\_profil.php", "User.php", "Utility.php", "verif\_identite\_page\_admin.php", "verif\_identite\_page\_user\_inscrit\_only.php", "verif\_identite\_page\_user.php", "index.html".

On définit ensuite le composant "Navigateur", abstraction de l'objet "Navigateur web", qui correspond au client voulant accéder au site web. On définit également le composant "Serveur", abstraction de l'objet "Serveur web". Enfin, on définit le composant "Base de données", abstraction de l'objet "Base de données".

On peut constater que certains composants d'objets sont similaires, on externalise donc le style de chaque page pour le mettre dans un objet, fichier appelé "style.charte.css". Cela va être montré dans la partie conception détaillée.

On crée aussi 3 packages, le premier appelé CSS contenant l'objet « style.css », le deuxième appelé HTML contenant l'objet « index.html », le troisième intitulé PHP contenant tous les objets en .php.

On représente les relations entre "BD" et "Logging", "Users", "Weak\_passwords".

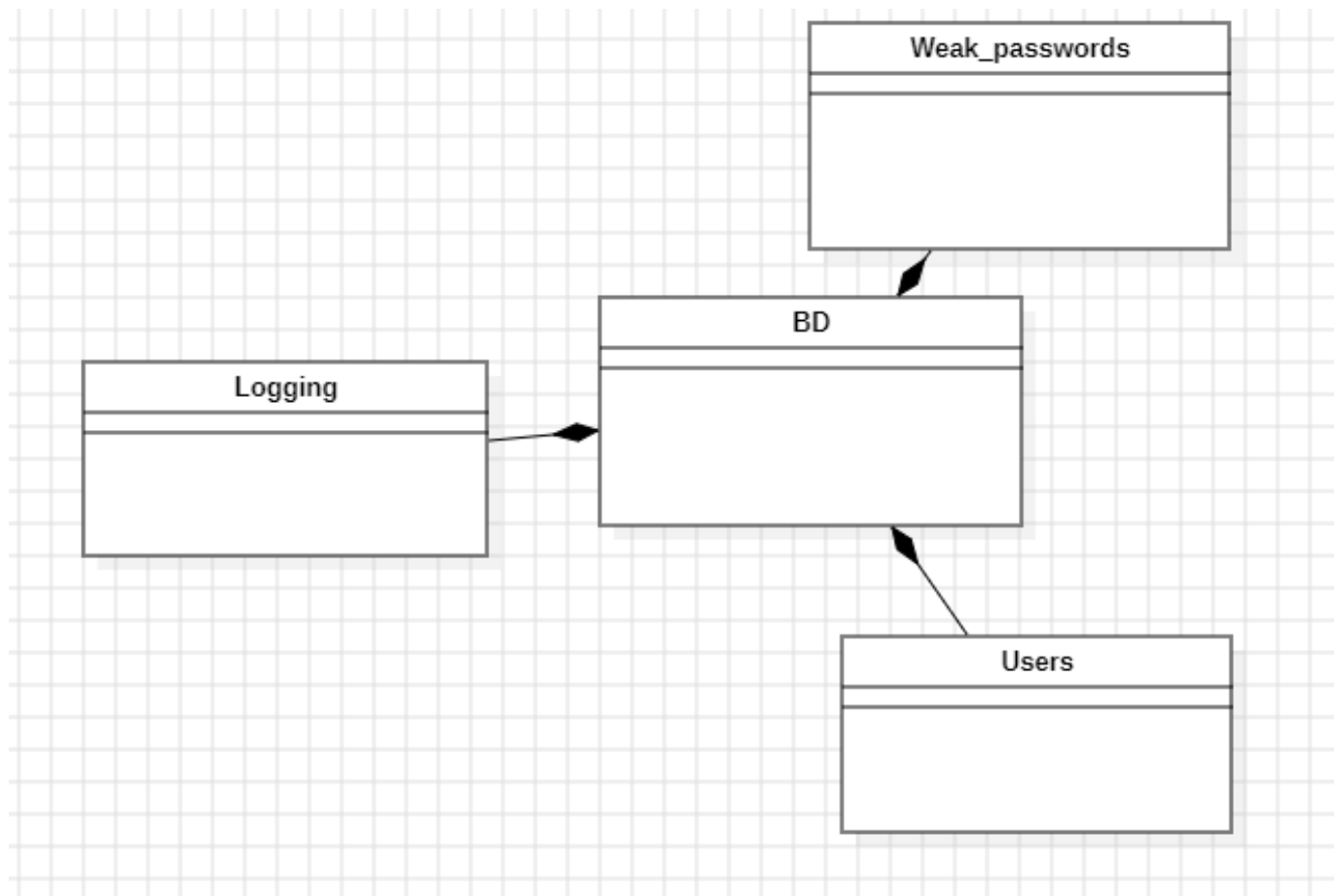


Diagramme BD du livrable 1

Ensuite, représente les relations entre les packages dans le diagramme de packages ci-dessous.

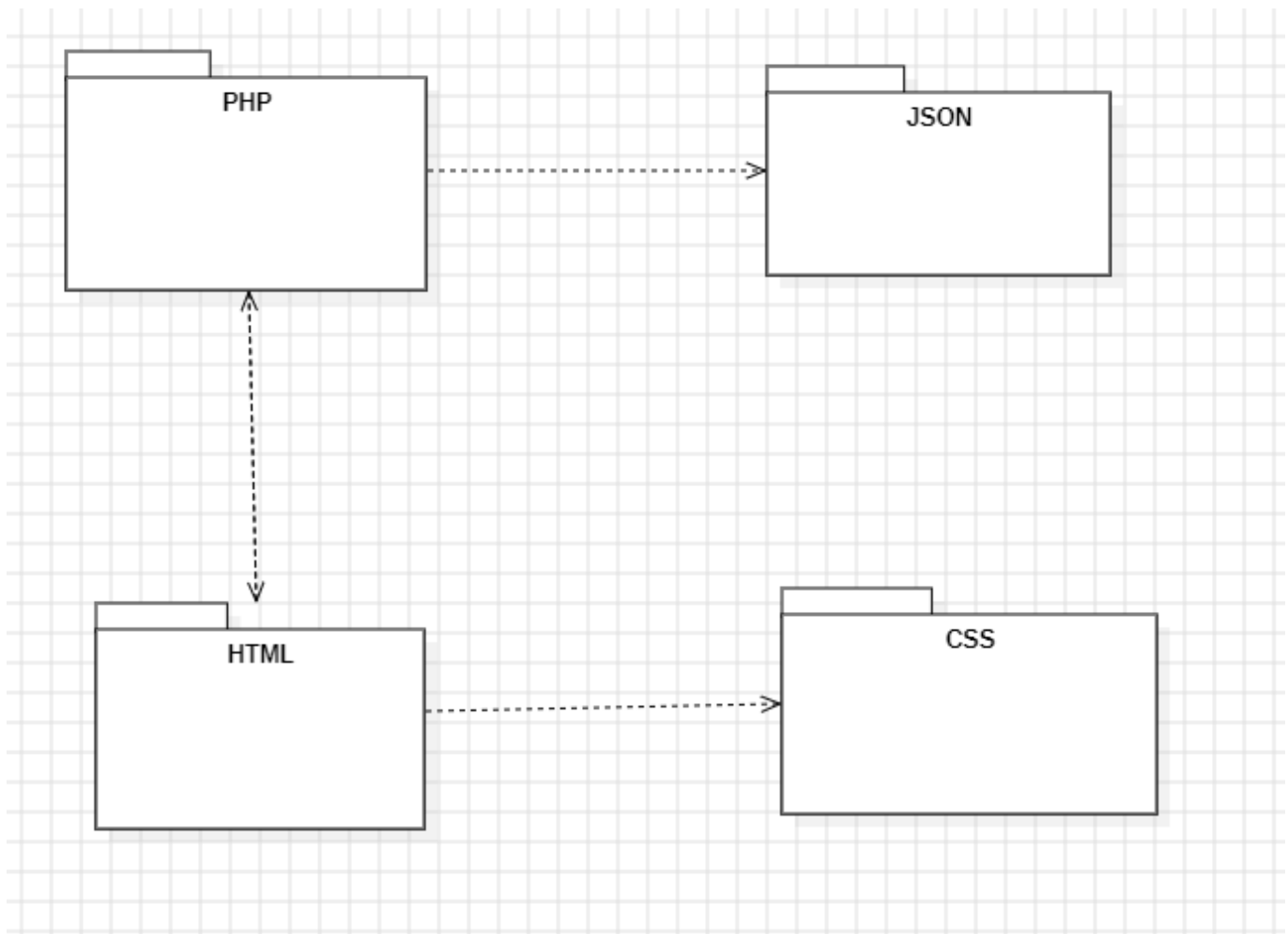


Diagramme de packages du livrable 1

On va utiliser la vue composants-connecteurs pour modéliser le système.

Les composants "Serveur" et "Navigateur" sont de stéréotype "executable", le composant "DB" est de stéréotype "database" avant les composants "utilisateur", "utilisation\_module" qui sont de stéréotype "table" et enfin tous les autres composants sont de stéréotype "file".

On a la même interface nommée « IRequeteWeb » entre « Navigateur » et « Serveur ». On a des relations de dépendances entre « Serveur » et les composants de stéréotype « file ». De même entre le "Navigateur" et les composants de stéréotype "html".

Le composant « Serveur » interprète les composants en .php et envoi les fichiers html correspondants au « Navigateur ».

On a une interface entre les composants "Navigateur" et "Serveur" que l'on va nommer "IRequeteWeb". On a des relations de dépendances entre "Serveur" et les composants de stéréotype "file", de même entre le "Navigateur" et les composants de stéréotype html.

On modélise cette vue à l'aide d'un diagramme de composants ci-dessous :

On a une interface que l'on va nommer "MySql" reliant le composant "MySQLDataManagement.php" avec le composant "Base de données".

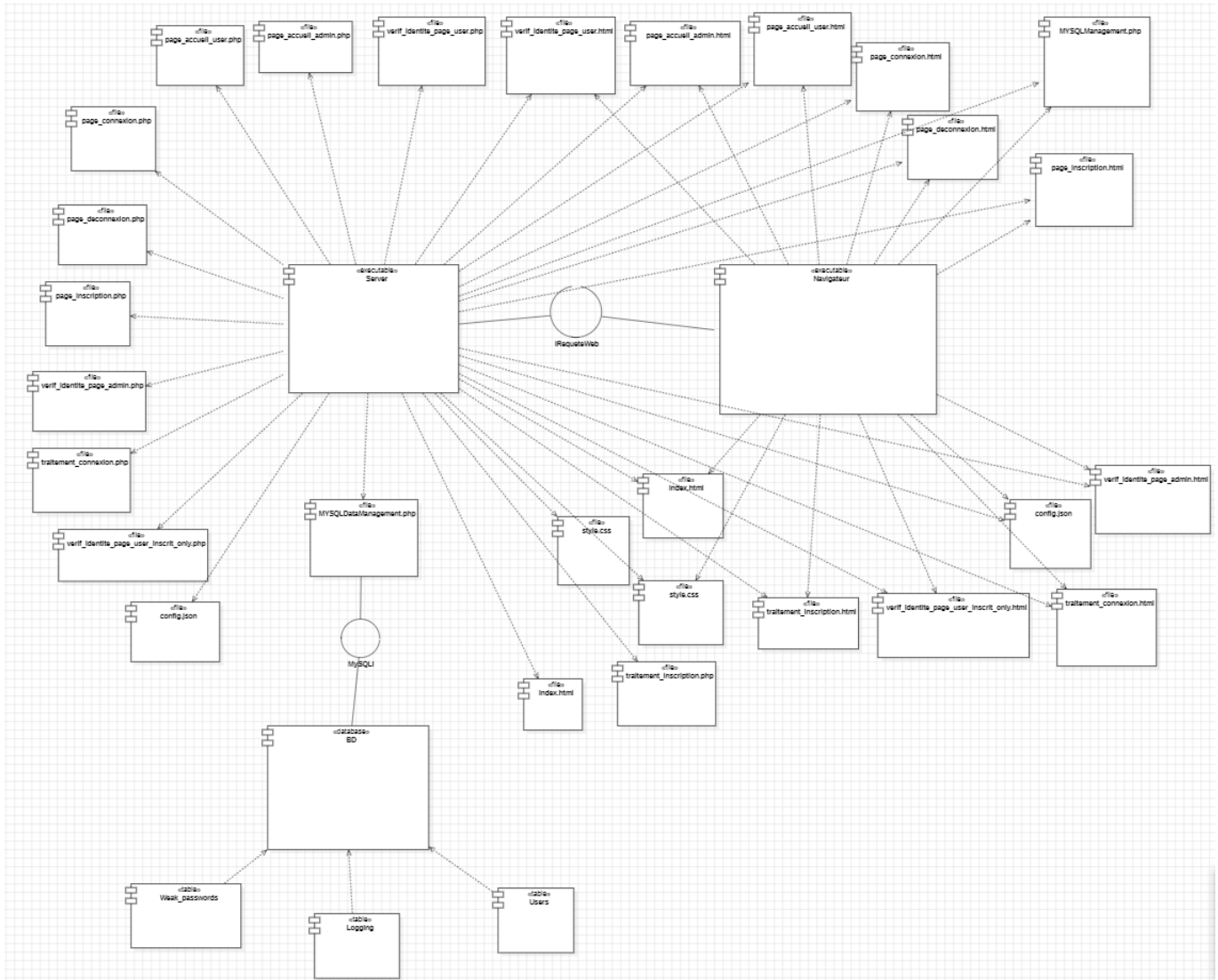
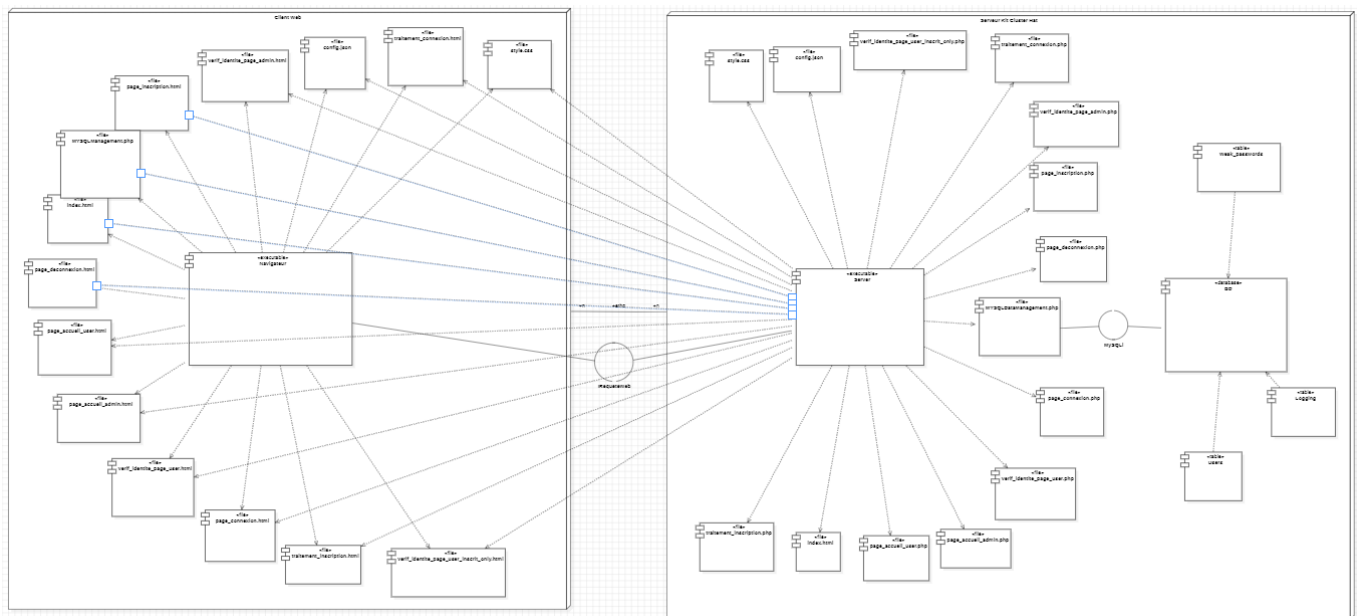


Diagramme de composants du livrable 1

L'étude du système ainsi que sa modélisation à l'aide de la vue composant-connecteur nous a permis de représenter le système qui est le site web. Nous allons représenter maintenant le système du point de vue de son déploiement à l'aide de la vue d'attribution.

Le serveur, la base de données et les pages sont stockées dans une machine Kit Cluster Hat qu'on va appeler "Serveur Kit Cluster Hat" contenant ces composants. De plus, comme le site web doit être accessible pour tous les ordinateurs étant sur le même réseau que la machine "Serveur Kit Cluster Hat", on a le composant Navigateur ainsi que les pages web envoyées par le serveur qui sont dans la machine du client, qu'on va appeler Client Web. Il peut y avoir 0 ou n Client Web car plusieurs utilisateurs peuvent se connecter au serveur. Il peut également y avoir 0 ou n "Serveur Kit Cluster Hat", car des répliques de ce serveur peuvent être créées.

Enfin, les 2 machines sont reliées par une connexion internet, représentée par une relation d'association dans le diagramme.



### Diagramme de déploiement du livrable 1



## Conception détaillée

En ayant défini les dépendances et les associations entre tous les objets, on obtient alors une structure telle que :

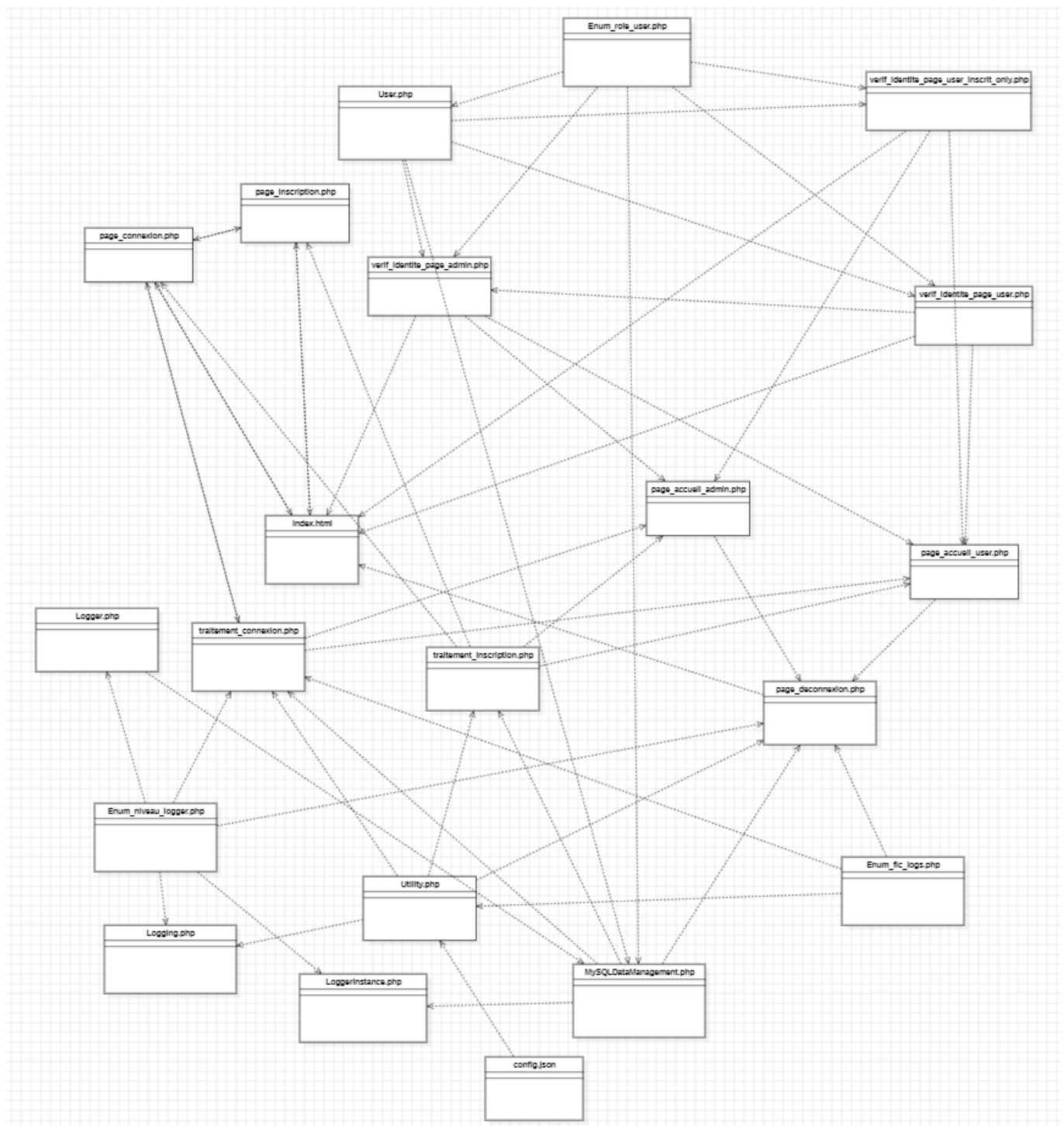


Diagramme de classes du livrable 1 (1)

On commence par analyser les différentes classes .php, à savoir "Logger.php", "LoggerInstance.php", "Logging.php", "MySQLDataManagement.php", "User.php" et "Utility.php".

On analyse aussi les énumérations "Enum\_fic\_logs.php", "Enum\_niveau\_logger.php", "Enum\_role\_user.php".

Pour faciliter l'analyse de ces dernières, on réalise un diagramme de classe ne contenant que ces classes, leurs paramètres et leurs méthodes et ces énumérations. Il servira à modéliser les dépendances et les associations entre ces dernières.

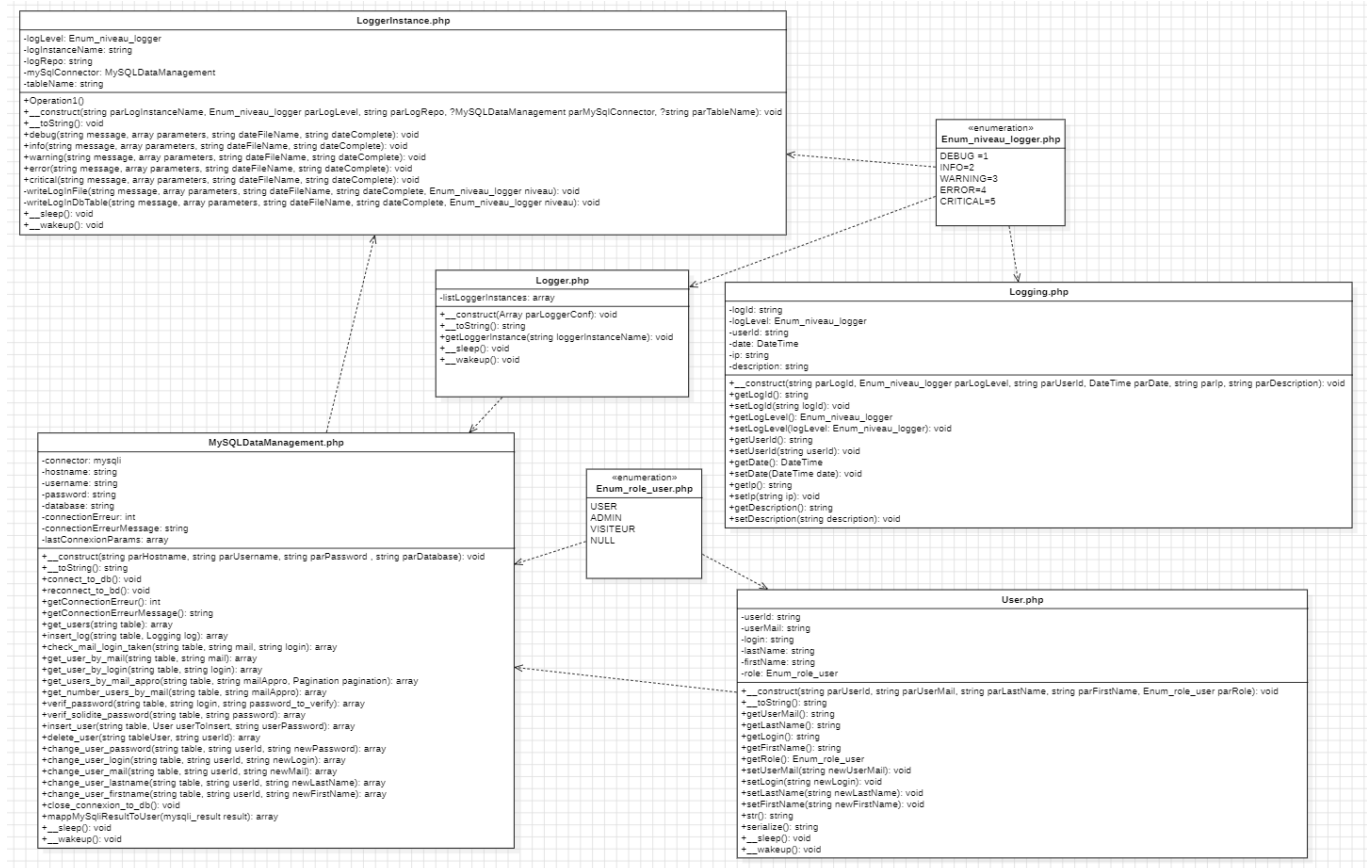


Diagramme de classes du livrable 1 (2)

On analyse maintenant comment sont constitués les objets en .php, à savoir "page\_accueil\_admin.php", "page\_accueil\_user.php", "page\_connexion.php", "page\_deconnexion.php", "page\_inscription.php".

La page "page\_deconnexion.php" ne contient pas d'html et n'apparaîtra jamais sur l'écran de l'utilisateur, elle permet uniquement de déconnecter la session de l'utilisateur.

La page "page\_accueil\_admin.php" ne contient qu'un lien de déconnexion contenu dans un `<ul><li>` lui même contenu dans un `<nav>`. Elle contient un `<h1>` qui permet d'afficher le nom de la page. Elle contient aussi un `<h2>` dans le `<main>` qui affiche le nom de l'admin.

La page "page\_accueil\_user.php" ne contient qu'un lien de déconnexion contenu dans un `<ul><li>` lui même contenu `<nav>`. Elle contient un `<h1>` qui permet d'afficher le nom de la page. Elle contient aussi un `<h2>` dans le `<main>` qui affiche le nom de l'utilisateur.

La page "page\_connexion.php" contient un `<h2>` qui permet d'afficher le nom de la page. Elle contient ensuite un `<form>` qui contient lui même deux `<div>` qui contiennent chacun un `<label>` et un `<input>` de type text. Ce sont les champs de notre formulaire. Ce `<form>` contient également un `<input>` de type submit qui correspond au bouton de validation. Cette page contient un deuxième `<form>` qui contient un `<input>` de type submit. Elle contient aussi un `<p>` servant à l'affichage des erreurs. Enfin, elle contient un `<ul>` dans un `<nav>` avec deux `<li>` qui contiennent chacun un `<a>` avec un href.

La page "page\_inscription.php" contient un `<h2>` qui permet d'afficher le nom de la page. Elle contient ensuite un `<form>` qui contient lui même six `<div>` qui contiennent chacun un `<label>` et un `<input>` de type text. Ce sont les champs de notre formulaire. Ce `<form>` contient également un `<input>` de type submit qui correspond au bouton de validation. Cette page contient un deuxième `<form>` qui contient un `<input>` de type submit. Elle contient aussi un `<p>` servant à l'affichage des erreurs. Enfin, elle contient un `<ul>` dans un `<nav>` avec deux `<li>` qui contiennent chacun un `<a>` avec un href.

Nous allons ensuite effectuer la conception détaillée de la base de données. La base de données se décompose de 3 tables :

La table Users comporte 7 colonnes, à savoir `userId` de type varchar, `userMail` de type varchar, `login` de type varchar, `lastName` de type varchar, `firstName` de type varchar, `password` de type varchar et `role` de type varchar. De plus, on a des contraintes d'intégrités sur les colonnes, le nombre de caractères maximum est de 36 pour la colonne `userId`, de 35 pour la colonne `userMail`, de 256 pour la colonne `password` et de 10 pour la colonne `role`. le nombre de caractères maximal est de 25 pour les colonnes `login`, `lastName` et `firstName`. On a également une clé primaire sur la colonne `userId` pour rendre les identifiants uniques et pour empêcher le fait d'avoir deux tuples ayant le même login. Les colonnes ne peuvent par ailleurs pas être nulles. De plus, pour la colonne `role`, la valeur "USER" est insérée par défaut.

La table Users comporte 6 colonnes, à savoir `logId` de type varchar, `logLevel` de type varchar, `userId` de type varchar, `date` de type datetime, `ip` de type varchar et `description` de type text. De plus, on a des contraintes d'intégrités sur les colonnes, le nombre de caractères maximum est de 36 pour les colonnes `logId` et `userId`, de 10 pour la colonne `logLevel`, de 15 pour la colonne `ip`. On a également une clé primaire sur la colonne `logId` pour rendre les identifiants uniques et pour empêcher le fait d'avoir deux tuples ayant le même login. On a aussi une clé étrangère sur la colonne `userId`, cette dernière fait référence à la clé primaire du même nom dans la table Users. Toutes les colonnes sauf `logLevel` ne peuvent par ailleurs pas être nulles. La colonne `logLevel`, doit être nulle ou doit contenir des valeurs de criticités d'alerte des logs ('DEBUG', 'INFO', 'WARNING', 'ERROR', 'CRITICAL'). De plus, pour la colonne `role`, la valeur "USER" est insérée par défaut.

La table Weak\_passwords comporte une seule colonne `password` de type varchar avec des contraintes d'intégrité telles que : le nombre de caractères contenus dans cette colonne doit être de 256 au maximum et elle ne peut pas être nulle.