

# TASKSTEP : AUDIT DE QUALITE LOGICIELLE

1PEUDINSPI :

- Tristan DAL MOLIN
- Moulay-Wassim ALAOUI
- Jules DUTRION
- Matteo DE MARCO
- Wassim DIOURI

BUT2 Informatique



## I. Conception actuelle du projet

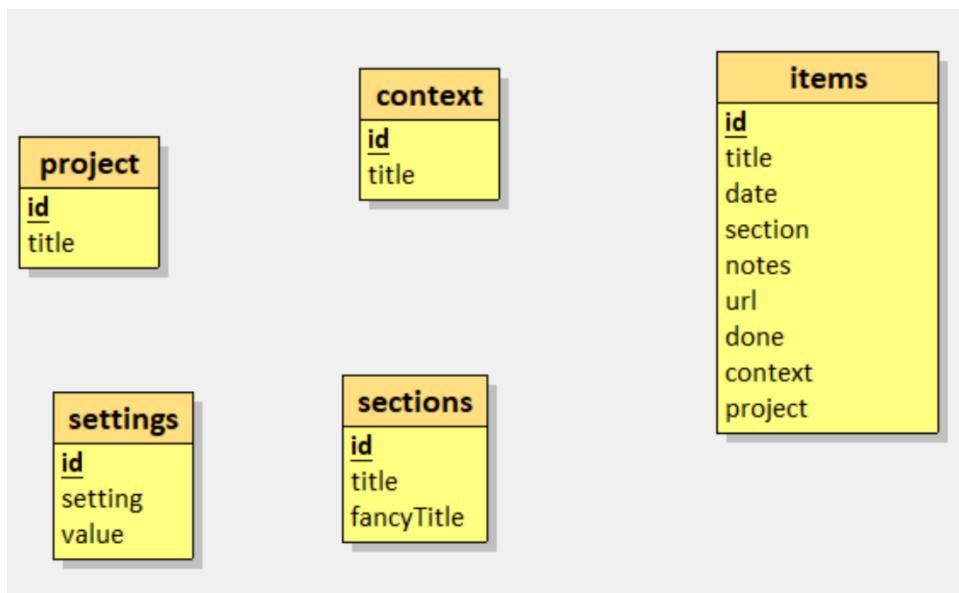


Figure 1 : Modèle de Conception de Données du projet

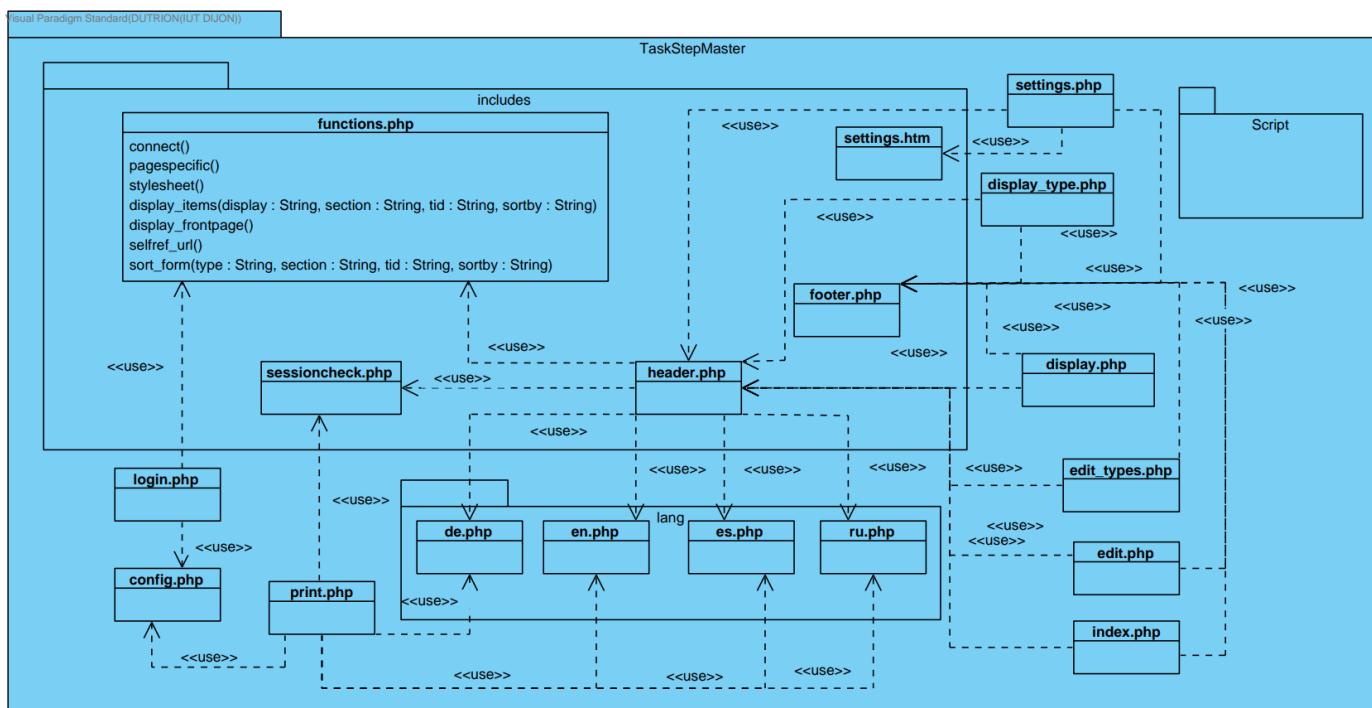


Figure 2 : Conception de l'application (UML)

## II. Audit de qualité logicielle

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
<b>A – Documentation technique</b>				
<b>A.1 – Documentation externe</b>				
A.1.1 Document indiquant la plateforme d'exécution (Qu'installer sur une VM pour exécuter l'appli)	Visual Studio Code	Lecture du fichier README.htm pour analyser les étapes d'installation locale et vérifier les composants nécessaires (AMP).	BP – L'application doit être exécutée sur un serveur local type XAMPP contenant Apache, MySQL et PHP et tout est indiqué en README.	N/A
A.1.2 Document indiquant la plateforme de développement	Visual Studio Code	Lecture du README.md pour vérifier la plateforme de développement	BP – Le projet est développé en PHP. Aucun environnement spécifique n'est imposé, tout est indiqué en README	N/A
A.1.3 Document décrivant l'architecture globale de l'application	Visual Studio Code	Vérification que l'architecture du projet est présente dans sa documentation (UML, conception, ...)	N/A – Aucune architecture, non applicable	N/A
A.1.4 Document décrivant les dépendances aux outils tiers de l'application	Visual Studio Code	Lecture du README.md pour vérifier les dépendances aux outils tiers de l'application (API extérieures, logiciels à installer en plus de l'application de base)	BP – Présence des dépendances du projet en README. La plateforme d'exécution nécessite PHP ≥ 7.0, MariaDB et un serveur local pour exécuter le projet	N/A
A.1.5 Cahier des charges ou document décrivant les fonctionnalités de l'application	Visual Studio Code	Lecture du README.md pour vérifier si les fonctionnalités y sont décrites. (Diagrammes de séquences, descriptions des fonctionnalités)	CF – Les fonctionnalités sont listées dans le README : organisation des tâches, filtres contextuels/projets, affichage auto des tâches du jour, impression, surlignage des tâches urgentes ou en retard, multilingue (Anglais, Russe, Allemand, Espagnol).	N/A

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
A – Documentation technique				
A.2 – Documentation interne				
A.2.1 Documentation in-situ du code présente (Code documenté)	Visual Studio Code	Analyse du code existants des différents fichiers présents dans le projet.	CR – Il semble qu'il n'y ait vraiment aucune documentation dans tout le projet. Seulement des commentaires rarement.	
A.2.2 Documentation du code cohérente avec le code et à jour	N/A	Vérification que la documentation a du sens et a une version, que les paramètres y soient ajoutés si ajoutés à une version du site	N/A (Pas de documentation)	
A.2.3 Documentation du code cohérente avec les spécifications fonctionnelles (cf A.1.5., si ça respecte les fonctionnalités décrites)	N/A	Présence de documentation cohérente avec les spécifications	N/A (Pas de documentation)	1h
A.2.4 Documentation du code claire, lisible, dans la langue souhaitée	N/A	Documentation claire, lisible et dans une même langue	N/A (Pas de documentation)	

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
B – Tests de l'application				
B.1 – Tests fonctionnels				
B.1.1 Les cas d'usage de l'application sont tous implémentés (Respect des diagrammes d'usage en conception)	N/A	Vérification que les cas d'usage sont tous respectés	N/A -- Pas de diagrammes de conceptions	N/A
B.1.2 Les spécifications fonctionnelles sont respectées	Firefox/ Edge	Vérifications que les fonctionnalités de la documentation sont bien respectées	CF – La plupart sont conformes, excepté le changement de langue du navigateur (changement de langue possible via le fichier config)	30 min
B.1.3 L'application fonctionne sur les plateformes d'exécution spécifiées	Oracle vm virtualbox	Vérifier que l'application fonctionne sous machine virtuelle en suivant le guide d'installation et d'utilisation.	CF – Pas de tuto pour un Xampp ou autre chose dans le genre, sinon tuto fonctionnel	10 min
B.1.4 L'application est robuste aux cas particuliers	Firefox	Vérification qu'aucune mauvaise saisie n'est possible (lettres à la place d'un nombre par exemple)	NC – Insertion possibles avec des champs nécessaires vides, même nom de contexte et de projet possibles, une date peut être entrée en tant que texte et non que date	1h

Add context      Add project

SampleContext    215

Item added!

Title:

Notes:

Section:

Context:

Project:

Edit contexts    Edit projects

Due date:

Url:

Taskstep 1.1 - By Rob Lowcock, Ethan Romba, and Thomas Hooge

 Add project



215

215

 Add context

 SampleContext

 SampleContext

Due date: azertyuiop

HTTP Status: 404 (not found)

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
B – Tests de l'application				
B.2 – Tests unitaires				
B.2.1 Les tests unitaires sont présents et cohérents avec les spécifications	Visual Studio Code	Vérification que l'application a des tests unitaires cohérents pour les cas généraux et particuliers	NC – Pas de tests unitaires	
B.2.2 Les tests unitaires passent tous	Visual Studio Code	Vérification que tous les tests unitaires passent	N/A -- Pas de tests unitaires	
B.2.3 Les tests unitaires sont indépendants les uns des autres	Visual Studio Code	Tests unitaires indépendants entre eux, simulation de couches inférieures, fakes DAO par exemple	N/A -- Pas de tests unitaires	2h
B.2.4 les cas particuliers (entrées invalides, effets de bord) sont testés	Visual Studio Code	Vérification que les cas particuliers sont testés	N/A -- Pas de tests unitaires	
B.2.5 la couverture des tests est correcte	Visual Studio Code	Couverture des tests correcte (pourcentage éventuel)	N/A -- Pas de tests unitaires	

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
C – Format du code source				
C.1 – Construction du code				
C.1.1 Lors de la construction du code (compilation) il n'y a pas d'avertissement (À modifier si c'est dangereux, peuvent être laissés de côté sinon)	« phpstan » sous Composer et Linux, Visual Studio Code	Commande « phpstan analyse » effectuée sur chaque classe PHP du projet	NC – Plus de 700 erreurs sont trouvées au total, dont beaucoup de méthodes appelées à des éléments	2h (correction des différents bugs)
<pre>         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid] 49   [Cannot call method query() on mixed.]         [method. nonObject] 51   [Cannot call method fetch_assoc() on mixed.]         [method. nonObject]         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid]         [binaryOp.invalid]         [offsetAccess. nonOffsetAccessible]         [property. nonObject] 20   [Cannot call method fetch_row() on bool mysqli_result.]         [method. nonObject] 14   [Cannot call method query() on mixed.]         [method. nonObject]         [property. nonObject]         [binaryOp.invalid]         [binaryOp.invalid] 26   [Cannot call method query() on mixed.]         [method. nonObject] 27   [Cannot call method fetch_array() on mixed.]         [method. nonObject]         [offsetAccess. nonOffsetAccessible]         [binaryOp.invalid]         [binaryOp.invalid]         [offsetAccess. nonOffsetAccessible]         [offsetAccess. nonOffsetAccessible] </pre>				

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
C.1.2 Lors de la construction du code il n'y a pas d'erreurs	« phpstan » sous Composer et Linux, Visual Studio Code	Commande « phpstan analyse » effectuée sur chaque classe PHP du projet, sous une configuration nous permettant d'obtenir tous les résultats possibles	NC – Plus de 700 erreurs sont trouvées au total, beaucoup de non-typage, de méthodes appelées sur des éléments « mixed »	2h (correction des différents bugs)

analyse.txt x

analyse.txt

175

328 143 Parameter #1 (mixed) of echo cannot be converted to string.

330 143 Variable \$id might not be defined.

331 | variable.undefined

332

333

334

335 Line edit\_types.php

336

337 12 Parameter #1 \$string of function addslashes expects string, mixed given.

338 | argument.type

339 13 Cannot call method query() on mixed.

340 | method.nonObject

341 13 Part \$eid (mixed) of encapsulated string cannot be cast to string.

342 | encapsaledStringPart.nonString

343 13 Part \$type (mixed) of encapsulated string cannot be cast to string.

344 | encapsaledStringPart.nonString

345 13 Variable \$mysqli might not be defined.

346 | variable.undefined

347 14 Binary operation "." between '<div id=\\'updated\\\" and mixed results in an

PROBLÈMES 1 SORTIE CONSOLE DE DÉBOGAGE TERMINAL PORTS

wsl + □ 🗑 ... ^ x

71 Cannot access offset 'e69f7f6ee287b969198c3c9d6777bd38' on mixed.  
| offsetAccess.nonOffsetAccessible

78 Cannot access offset '8825ede83f2f289127722d4e842cf7e8' on mixed.  
| offsetAccess.nonOffsetAccessible

81 Cannot access offset '8825ede83f2f289127722d4e842cf7e8' on mixed.  
| offsetAccess.nonOffsetAccessible

88 Cannot access offset '23c18046f52bef3eea034657bafda50f' on mixed.  
| offsetAccess.nonOffsetAccessible

91 Cannot access offset '23c18046f52bef3eea034657bafda50f' on mixed.  
| offsetAccess.nonOffsetAccessible

[ERROR] Found 759 errors

td@info-146-25-15:/mnt/c/xampp/htdocs/taskstep\$ vendor/bin/phpstan analyse > analyse.txt  
Note: Using configuration file /mnt/c/xampp/htdocs/taskstep/phpstan.neon.  
32/32 [██████████] 100%

*Figure 3 : Analyse de construction du code (php -l)*

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
C – Format du code source				
C.2 – Aspect du code source				
C.2.1 L'indentation du code est correcte	Visual Studio Code, Webimpress Coding Standard	On effectue une analyse du code sous Webimpress Coding Standard afin de vérifier le nombre d'erreurs d'indentation	NC- On remarque donc la présence de presque 2000 erreurs d'indentation dans ce projet (voir figure ci-dessous)	2h

RECHERCHE    ⌂ ⌚ ⌚ ⌚ ⌚ ⌚ ⌚

indent    Aa ab \*

Remplacer AB 🔍

1987 résultats dans 1 fichier - Ouvrir dans l'éditeur

analyseCS.txt 1987

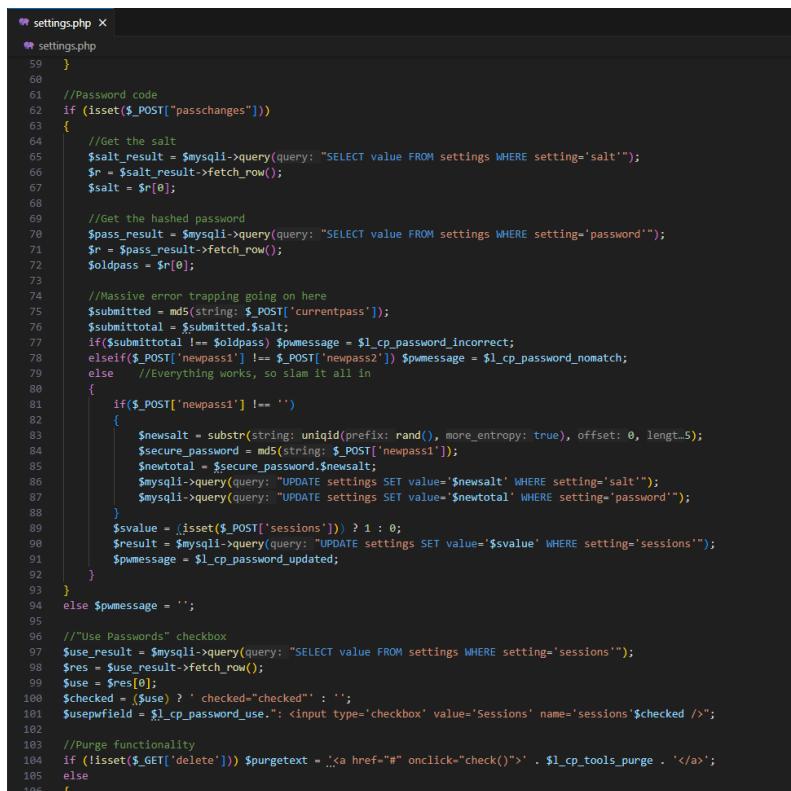
- ...x] Spaces must be used to indent...
- ...31mERROR[0m | [x] Invalid ind...
- ...x] Spaces must be used to indent...
- ...31mERROR[0m | [x] Invalid ind...
- ...x] Spaces must be used to indent...
- ...31mERROR[0m | [x] Invalid ind...
- ...x] Spaces must be used to indent...
- ...31mERROR[0m | [x] Invalid ind...
- ...x] Spaces must be used to indent...
- ...31mERROR[0m | [x] Invalid ind...
- ...x] Spaces must be used to indent...

```

includes > functions.php ...
173 function sort_form($type = '', $section = '', $tid = '', $sortby = ''): void{
196     }
197     ?>
198     <div class="sortform">
199         <a href="print.php?<?php echo $printurl; ?>"><img src='images/printer.png' alt=''/> <?php echo $l_items_print; ?></a></span></p>
200         <form action="display.php" method="get">
201             <div>
202                 <input type="hidden" name="display" value="<?php echo $type ?>" />
203                 <?php echo $hidden . $l_items_sorttext ?>
204                 <select name="sort">
205                     <?php
206                         foreach ($l_items_sort AS $key=>$value)
207                         {
208                             if ($key != $type)
209                             {
210                                 $selected = ($sortby == $key) ? ' selected="selected"' : '';
211                                 echo ...<option value="<?php echo $key.'_'.$selected.'>'.<?php echo $value.'>'</option>';
212                             }
213                         ?>
214                     </select>
215                     <input type="submit" value="<?php echo $l_items_sortbutton; ?>" />
216                 </div>
217             </form>
218         </div><?php
220     }
221     ?>

```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
C.2.2 Il n'y a pas d'espace inutiles (blancs) dans le code	Visual Studio Code	Vérification dans chaque fichier qu'aucun grand espace inutile n'est laissé	BP – Aucun réel espace inutile ne se trouve dans le code	/
C.2.3 Le code est facile à lire	Visual Studio Code	On vérifie la présence de commentaires si une fonction n'est pas assez claire sur son propos, la bonne nomenclature de variables et de fonctions, etc.	NC – Les variables sont bien nommées, certains scripts PHP sont correctement commentés (bien que manquant de documentation), mais beaucoup manquent de clarifications et il est difficile de comprendre quelle classe fait quoi (voir figures ci-dessous)	1h



```

settings.php X
settings.php
59 }
60
61 //Password code
62 if (isset($_POST["passchanges"]))
63 {
64     //Get the salt
65     $salt_result = $mysqli->query(query: "SELECT value FROM settings WHERE setting='salt'");
66     $r = $salt_result->fetch_row();
67     $salt = $r[0];
68
69     //Get the hashed password
70     $pass_result = $mysqli->query(query: "SELECT value FROM settings WHERE setting='password'");
71     $r = $pass_result->fetch_row();
72     $oldpass = $r[0];
73
74     //Massive error trapping going on here
75     $submitted = md5(string: $_POST['currentpass']);
76     $submittotal = $submitted.$salt;
77     if($submittotal !== $oldpass) $pumessage = $l_cp_password_incorrect;
78     elseif($_POST['newpass1'] !== $_POST['newpass2']) $pumessage = $l_cp_password_nomatch;
79     else //Everything works, so slam it all in
80     {
81         if($_POST['newpass1'] !== '')
82         {
83             $newsalt = substr(string: uniqid(prefix: rand(), more_entropy: true), offset: 0, length:5);
84             $secure_password = md5(string: $_POST['newpass1']);
85             $newtotal = $secure_password.$newsalt;
86             $mysqli->query(query: "UPDATE settings SET value='$newsalt' WHERE setting='salt'");
87             $mysqli->query(query: "UPDATE settings SET value='$newtotal' WHERE setting='password'");
88         }
89         $value = (isset($_POST['sessions'])) ? 1 : 0;
90         $result = $mysqli->query(query: "UPDATE settings SET value='$value' WHERE setting='sessions'");
91         $pumessage = $l_cp_password_updated;
92     }
93 }
94 else $pumessage = '';
95
96 //Use Passwords" checkbox
97 $use_result = $mysqli->query(query: "SELECT value FROM settings WHERE setting='sessions'");
98 $res = $use_result->fetch_row();
99 $use = $res[0];
100 $checked = ($use) ? ' checked="checked"' : '';
101 $usepfield = $l_cp_password_use . ': <input type="checkbox" value="Sessions" name="sessions$checked />';
102
103 //Purge functionality
104 if (!isset($_GET['delete'])) $purgetext = ..<a href="#" onclick="check()>' . $l_cp_tools_purge . '</a>';
105 else
106 {

```

Code lisible

```
functions.php X
includes > functions.php > display_items()
167 function selfref_url(): void
168 {
169     $dirstuff = str_replace($search: basename(path: $_SERVER['PHP_SELF']), replace: '', subject: $_SERVER['PHP_SELF']);
170     $full = "http://". $_SERVER['HTTP_HOST']. $dirstuff;
171     echo $full;
172 }
173 1 reference
function sort_form($type = '', $section = '', $tid = '', $sortby = ''): void{
174
175     global $l_items_sorttext, $l_items_sort, $l_items_sortbutton, $l_items_print;
176     $sortby = ($sortby) ? $sortby : 'done';
177
178     switch($type) {
179         case 'section':
180             $printurl = "print=section&section=$section";
181             $hidden = "<input type='hidden' name='section' value='$section' />";
182             break;
183         case 'context':
184             $printurl = "print=$type&id=$tid";
185             $hidden = "<input type='hidden' name='tid' value='$tid' />";
186             break;
187         case 'today':
188             $printurl = "print=today";
189             $hidden = "";
190             break;
191         case 'all':
192             $printurl = "print=all";
193             $hidden = "";
194             break;
195     }
196 }
197 ?>
<div class="sortform">
<p><span class='printer'><a href="print.php?<?php echo $printurl; ?>"><img src='images/printer.png' alt=''/> <?php echo $l_items_print; ?></a></span></p>
<form action="display.php" method="get">
<div>
<input type="hidden" name="display" value="<?php echo $type ?>"/>
<?php echo $hidden . $l_items_sorttext ?>
<select name="sort">
    <?php
        foreach ($l_items_sort AS $key->$value)
        {
            if ($key != $type)
            {
                $selected = ($sortby == $key) ? ' selected="selected"' : '';
                echo "<option value='".$key."' $selected .'>".$value."</option>";
            }
        }
    </?php
</select>
</div>
</form>
</div>
```

## Code illisible

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
C.2.4 Le code est dans la langue spécifiée	Visual Studio Code	On vérifie que le code du projet entier est écrit en une même langue	BP – Le code entier est en anglais, jusque dans sa documentation et ses commentaires	/
C.2.5 La taille du code est correcte	Visual Studio Code	Vérification que chaque fonction fait moins de 30 lignes et qu'une classe/un fichier fasse moins de 300 lignes	CF – Chaque fichier fait effectivement moins de 300 lignes mais beaucoup de fonctions dépassent les 30 lignes (voir figure ci-dessous), refactoring et séparation des responsabilités nécessaires	2h

```

1 reference
173 function sort_form($type = '', $section = '', $tid = '', $sortby = ''): void{
174
175     global $l_items_sorttext, $l_items_sort, $l_items_sortbutton, $l_items_print;
176     $sortby = ($sortby) ? $sortby : 'done';
177
178     switch($type) {
179         case 'section':
180             $printurl = "print=section&section=$section";
181             $hidden = "<input type='hidden' name='section' value='$section' />";
182             break;
183         case 'context':
184             case 'project':
185                 $printurl = "print=$type&id=$tid";
186                 $hidden = "<input type='hidden' name='tid' value='$tid' />";
187             break;
188         case 'today':
189             $printurl = "print=today";
190             $hidden = "";
191             break;
192         case 'all':
193             $printurl = "print=all";
194             $hidden = "";
195             break;
196     }
197     ?>
198     <div class="sortform">
199     <><span class="printer"><a href="print.php?<?php echo $printurl; ?>"> <?php echo $l_items_print; ?></a></span></p>
200     <form action="display.php" method="get">
201     <div>
202         <input type="hidden" name="display" value="<?php echo $type ?>" />
203         <?php echo $hidden . $l_items_sorttext ?>
204         <select name="sort">
205             <?php
206             foreach ($l_items_sort AS $key=>$value)
207             {
208                 if ($key != $type)
209                 {
210                     $selected = ($sortby == $key) ? ' selected="selected"' : '';
211                     echo "<option value='".$key."'". $selected .'>'.$value.'

```

Fonction prenant plus de 60 lignes

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
C.2.6 Le code ne présente pas d'élément redondant	Visual Studio Code	Vérification qu'il n'existe pas 2 classes ou fonctions faisant la même chose (principe DRY), vérifier l'usage d'héritage ou d'interface dans ce cas-là	NC – Les récupérations en BDD depuis du code PHP pourraient être factorisées en un même appel sécurisé	1h

```

26 //Show/Hide Tips checkbox
27 $result = $mysqli->query(query: "SELECT * FROM settings WHERE setting='tips'");
28 <while ($r=$result->fetch_array())
29 {
30     $checked = ($r['value']) ? ' checked="checked"' : '';
31     $tipsfield = $l_cp_display_tips.": <input type='checkbox' value='Display tips' name='tips'$checked />";
32 }
33
34 //Stylesheets code
35 $result = $mysqli->query(query: "SELECT * FROM settings WHERE setting='style'");
36 $styleoptions = '';
37 > while($r=$result->fetch_array())
38 {
39 }
40
41 //Password code
42 <if (isset($_POST["passchanges"]))
43 {
44     //Get the salt
45     $salt_result = $mysqli->query(query: "SELECT value FROM settings WHERE setting='salt'");
46     $r = $salt_result->fetch_row();
47     $salt = $r[0];
48
49     //Get the hashed password
50     $pass_result = $mysqli->query(query: "SELECT value FROM settings WHERE setting='password'");
51     $r = $pass_result->fetch_row();
52     $oldpass = $r[0];
53
54     //Massive error trapping going on here
55     $submitted = md5(string: $_POST['currentpass']);
56     $submittotal = $submitted.$salt;
57     if($submittotal != $oldpass) $pwmmessage = $l_cp_password_incorrect;
58     elseif($_POST['newpass1'] != $_POST['newpass2']) $pwmmessage = $l_cp_password_nomatch;
59     else    //Everything works, so slam it all in
60     {

```

Script de requêtes BDD

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
C.2.7 La norme de casse du langage utilisé est respectée	Visual Studio Code, Webimpress Coding Standard	Le camel case est respecté (attributs en minuscule, propriété en majuscule au début, pas d'underscores (ils peuvent être remplacé par une majuscule dans le mot), constantes en majuscules)	NC – Par analyse Webimpress Coding Standard, on détecte déjà presque 600 erreurs de camel case dans le code	1h

camel

Remplacer AB ⌂ ...

590 résultats dans 2 fichiers - [Ouvrir dans l'éditeur](#)

▼ analyseCS.txt 588

- ...l\_index\_welcome" is not in valid camel caps
- ...l\_index\_introm" is not in valid camel caps
- ...l\_index\_introa" is not in valid camel caps
- ...l\_index\_introe" is not in valid camel caps
- ...l\_index\_introtext" is not in valid camel caps
- ...l\_index\_1task" is not in valid camel caps format
- ...l\_index\_mtasks" is not in valid camel caps
- ...l\_index\_mtaske" is not in valid camel caps
- ...l\_index\_tip" is not in valid camel caps format
- ...arry\_txt" is not in valid camel caps format
- ...arry\_txt" is not in valid camel caps format
- ...arry\_txt" is not in valid camel caps format
- ...l\_dbp\_l1" is not in valid camel caps format
- ...l\_dbp\_add" is not in valid camel caps format
- ...l\_dbp\_edit" is not in valid camel caps format
- ...l\_msg\_itemdo" is not in valid camel caps format
- ...l\_msg\_itemundo" is not in valid camel caps
- ...l\_msg\_actionerror" is not in valid camel caps
- ...l\_sectionlist" is not in valid camel caps format
- ...l\_nav\_allitems" is not in valid camel caps
- ...menu\_date\_format" is not in valid camel caps
- ...l\_nav\_today" is not in valid camel caps format
- ...l\_msg\_notoday" is not in valid camel caps format

Soucis de camel case du projet

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
D – Principes de qualité logicielle				
D.1 – Principes SOLID				
D.1.1 Le principe de responsabilité unique (S) est respecté	Vscode	On vérifie si chaque fichier fait une seule tâche, grâce à ça il serait plus simple de maintenir l'application dans le futur (Une classe, une responsabilité. On cherche différentes classes faisant la même chose ou une faisant plusieurs choses à la fois)	CR - Le fichier display_type.php ne respecte pas le principe S pour la plupart des fichiers les appels à la base de données et la gestion de l'affichage sont mélangés dans le code ce qui est contraire au principe S : exemple ci-dessous	10h
<pre>//display all the projects/contexts \$result = \$mysqli-&gt;query(query: "select * from {\$type}s order by title"); echo "&lt;div id='editlist'&gt;\n&lt;p&gt;".\\$1_dbp_l1[\$type]."&lt;/p&gt;";</pre>				
D.1.2 Le principe ouvert-fermé (O) est respecté	Vs code	On vérifie s'il y a une isolation des couches et si pour ajouter une fonctionnalité on peut	CR - il n'y a aucune interface ni de séparation des packages donc si on veut ajouter une fonctionnalité on est obligé de modifier du code partout pour que ça marche	10h
D.1.3 Le principe de substitution de Liskov (L) est respecté	Visual Studio Code	Vérification qu'en cas d'héritage, le module de bas peut remplacer celui de haut niveau sans changer le fonctionnement (exemple des exceptions levées par les classes filles)	CR – Pas d'objets donc à mettre en place dans le projet	10h
D.1.4 Le principe de ségrégation des interfaces (I) est respecté	Visual Studio Code	Une classe n'implémente pas une interface dont elle n'a pas besoin de toutes les méthodes, éviter les interfaces pour tout, qui ont trop de méthodes (principe S)	CR – Pas d'objets donc à mettre en place dans le projet	
D.1.5 Le principe d'inversion des dépendances (D) est respecté		(Code de haut niveau ne dépend pas de bas niveau (métier qui parle à IHM directement))	CR – Pas d'objets donc à mettre en place dans le projet	

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
<b>D – Principes de qualité logicielle</b>				
<b>D.2 – Gestion des erreurs</b>				
D.2.1 – Les cas d'erreurs sont traités à l'aide des exceptions	VsCode	Test via l'IHM et analyse du code pour trouver des erreurs générées	CR – Aucune présence d'exception	1heure
D.2.2 Le point d'entrée du programme capture les exceptions non encore capturées	VsCode	Test via l'IHM et analyse du code pour trouver des erreurs générées et trouver des blocs de try catch	CR – Vu qu'il n'y a aucune exception les seules erreurs affichées viennent d'if/else	1heure
D.2.3 Les exceptions ne sont pas muselées mais capturées pour être traitées (catchs vides, à documenter si c'est volontaire)	VsCode	Test via l'IHM et analyse du code	CR - Aucune présences de try / cacth sauf dans la bibliothèque	1heure

The screenshot shows the TaskStep application interface. At the top, there is a navigation bar with links for 'Home', 'All Items', 'By context', 'By project', 'Settings', 'Help', and 'Logout'. Below the navigation bar, there are three main sections: 'Add item', 'Ideas (0) (9)', and 'Might Want to Buy (0) (1)'. A prominent 'Fatal error' message is displayed in the center of the screen, stating:

```
Fatal error: Uncaught mysqli_sql_exception: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near '';SELECT * FROM SESSIONS' at line 1 in C:\xampp\htdocs\taskstep-master\display.php:48
Stack trace: #0 C:\xampp\htdocs\taskstep-master\display.php(48): mysqli->query('SELECT *
FROM i...')
#1 {main} thrown in C:\xampp\htdocs\taskstep-master\display.php on line 48
```

Below this, another browser window is visible, showing a login page with a similar fatal error message:

```
Fatal error: Uncaught mysqli_sql_exception: Aucune connexion n'a pu être établie car l'ordinateur cible l'a expressément refusée in C:\xampp\htdocs\taskstep\includes\functions.php:6
Stack trace: #0 C:\xampp\htdocs\taskstep\includes\functions.php(6): mysqli->__construct('localhost', 'taskstep', Object(SensitiveParameterValue), 'taskstep')
#1 C:\xampp\htdocs\taskstep\login.php(7): connect()
#2 {main} thrown in C:\xampp\htdocs\taskstep\includes\functions.php on line 6
```

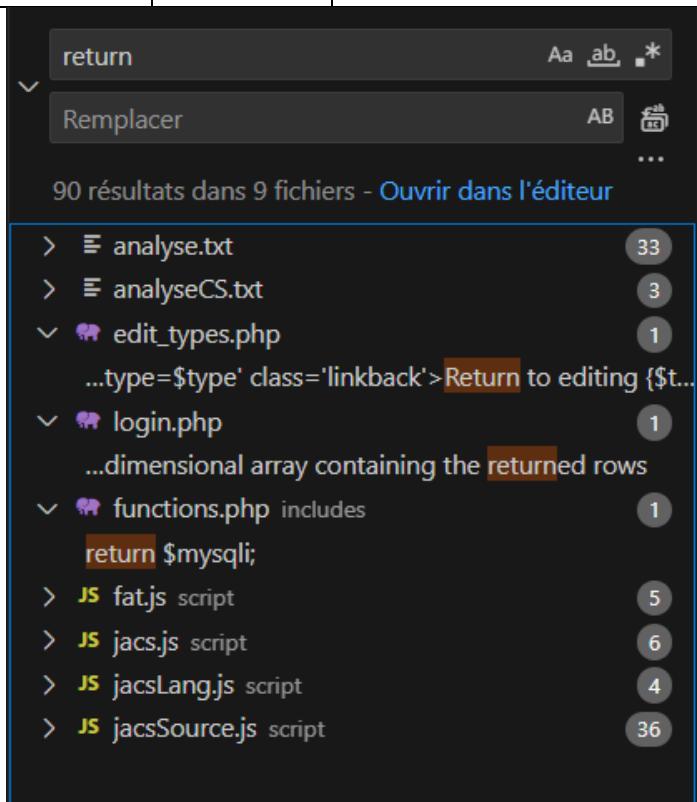
Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
D – Principes de qualité logicielle				
D.3 – Règles de codage				
D.3.1 – Il n'y a pas de « valeur magique » dans le code	Visual Studio Code	Il n'y a pas de valeurs arbitraires sorties de nulle part dans le code (limites de boucles par exemple)	CF – Une valeur magique se trouve au milieu des fonctions (display_frontpage() ) et devrait donc être échappée en dehors de la fonction (variable globale par exemple) ou en paramètre optionnel avec valeur par défaut (voir figures ci-dessous)	5 min

```

if ($urlfull == "") $url = "";
else
{
    $limit = 40; // set character limit
    $url = "<a href=\"$urlfull'>";
    // display URL up to character limit, shorten & add ellipsis if it is too long
    $url .= (strlen(string: $urlfull) > $limit) ? substr(string: $urlfull,offset: 0,length: $limit) . '...</a>';
}

```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
D.3.2 – chaque fonction ne possède qu'un return, sur la dernière ligne de la fonction	Visual Studio Code	Vérification qu'aucune fonction ne possède plus d'un return, ce dernier devant être placé en dernière ligne de fonction	BP – Aucune fonction ne possède plus d'un return dans le projet	/



Returns du projet

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
D.3.3 – aucune procédure ne contient de return	Visual Studio Code	Vérification qu'il n'y a pas de returns en procédure	BP – Il n'y a pas de returns en procédures du projet	/
D.3.4 – il n'y a aucun « goto » ou autre rupture du flux d'exécution	Visual Studio Code	Vérification qu'aucune rupture du flux n'a lieu dans le projet, qu'il n'y a pas de goto renvoyant à une autre ligne de code en particulier, ni de break en milieu de fonction	BP – Il n'y a pas de rupture du flux	/
D.3.5 – Le nom des variables est pertinent	Visual Studio Code	On vérifie que les noms donnés sont pertinents comme les noms de variable ou autre	NC – Certains noms ne sont absolument pas pertinents et sont non professionnels parfois par manque d'inspiration le nom est le même avec un numéro ce qui rend le code encore plus illisible	1h
<pre>\$editid = \$_GET["id"]; //DEBUG echo "This would produce an edit form for the context with id \$editid &lt;br /&gt;";  \$editquery = \$mysqli-&gt;query(query: "SELECT * FROM {\$type}s WHERE id=\$editid"); while(\$r=\$editquery-&gt;fetch_array()) {     \$edittitle = \$r["title"];     \$editid2 = \$r["id"]; } &lt;!--Open container--&gt; &lt;div id="sexyBG"&gt;&lt;/div&gt;&lt;div id="sexyBOX" onmousedown="document.onclick=function(){};" onmouseup="setTimeout('sexyTOG()',</pre>				
D.3.6 – le nom des opérations (fonctions, procédures) est pertinent	Visual studio code	Vérification que les noms des fonctions est pertinent et clair	NC – Certains noms de fonction ne sont pas très clair et ne possède pas de nomenclature	1h
<b>function pagespecific(): void</b>				
D.3.7 – il n'y a pas de code « mort » (commenté, non atteignable...) (fonctions jamais appelées, ou alors code qui n'est	Visual Studio Code	Vérification qu'il n'y a pas de code commenté ou non atteignable, de fonctions jamais appelées ou de code qui n'est	CF – Plusieurs zones de code sont commentées	5 min

jamais exécuté du programme		jamais exécuté du programme		
		<pre> 39 40 //if(\$_SESSION['loggedin'] == true) 41 //{ 42 // echo "You're already logged in! Either &lt;a href='logout.php'&gt;logout&lt;/a&gt; or continue to the &lt;a href='index.php'&gt;main page.&lt;/a&gt;"; 43 //} 44 </pre>		

debug

Aa ab \*

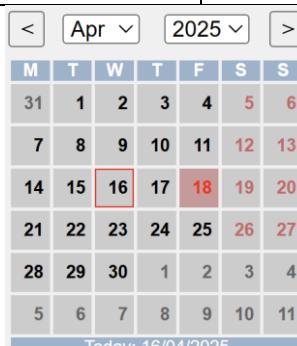
Remplacer AB

7 résultats dans 3 fichiers - [Ouvrir dans l'éditeur](#)

edit\_types.php 5

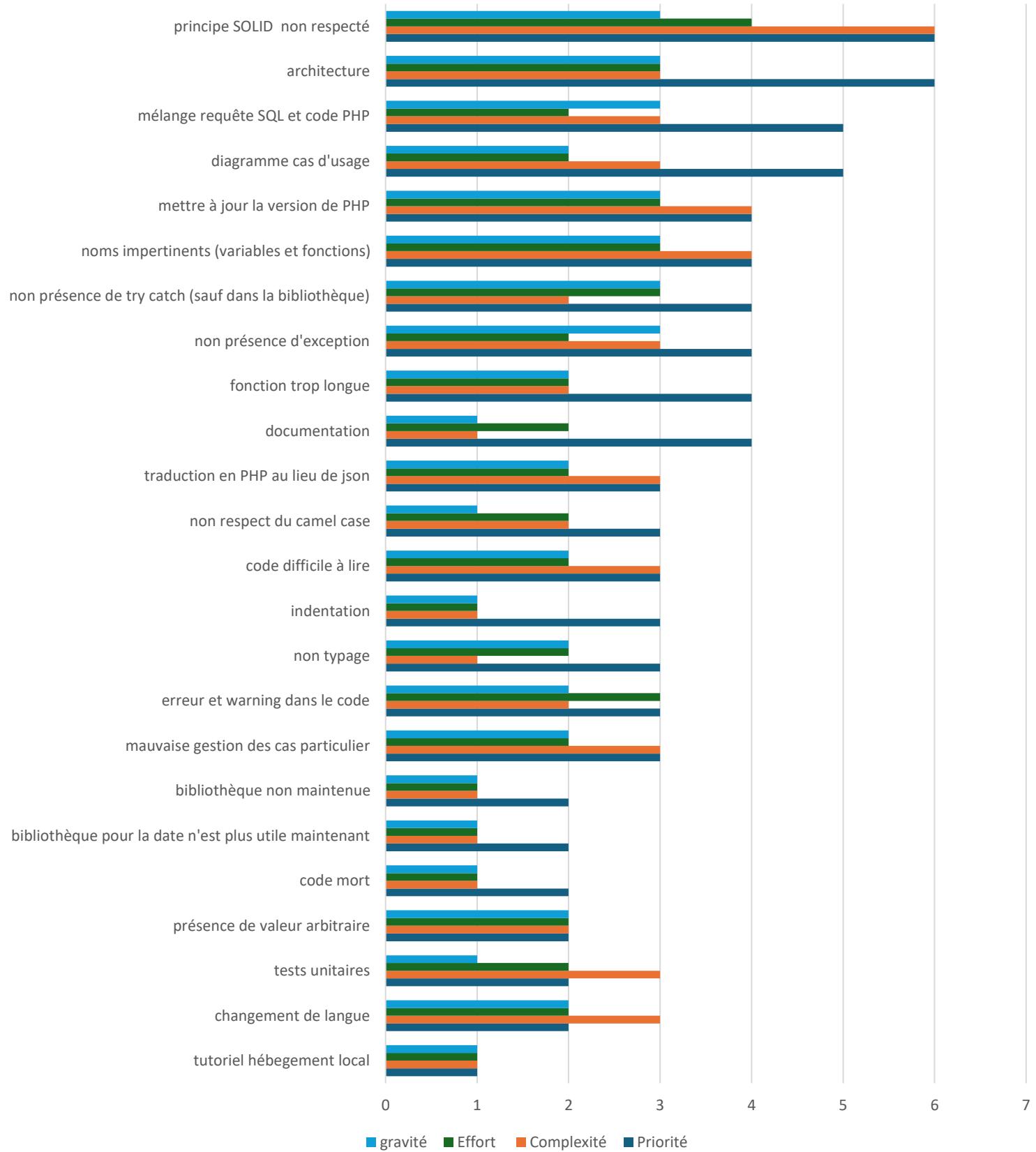
```
//DEBUG NB: quick error trap
//DEBUG echo "This would produce ...
//DEBUG echo "The MySQL code ha...
//DEBUG echo "ID: $editid2 <br />";
//DEBUG echo "Title: $edittitle <br /...
```

Critères d'audit	Outils utilisés	Descriptions des vérifications effectuées (tests, conditions de test, etc.)	Résultats de l'audit (constats)	Temps estimé d'optimisation
<b>E – Pile technique</b>				
<b>E.1 – Langage utilisé</b>				
E.1.1 – Le(s) langage(s) utilisés sont pertinents pour le problème donné (Exemple non pertinents : application en Python, IA en C#)	Visual Studio Code, XAMPP	Vérifications que les langages utilisés ne sont pas hors-sujet et sont pertinents à utiliser.	CF – Comme il s'agit d'un site web, le code PHP est adapté dans l'ensemble à de l'interaction côté serveur et à la communication avec une Base de Données Pour la base de données la communication se fait par l'utilisation de MySQL Pour la traduction, quant à elle, elle n'est pas réalisée dans le bon langage, elle est réalisée en PHP alors qu'elle devrait être en .json	1 heures
E.1.2 – La version utilisée du langage est supportée et maintenue (Ex : page en PHP5, failles non corrigées aujourd'hui)	Visual Studio Code, README	Vérification que le langage de programmation est supporté par les navigateurs web et maintenu encore aujourd'hui (lecture du README pour trouver les versions)	CR – La version utilisée est php 7.0 qui n'est plus maintenue, il faut mettre à jour les différentes méthodes n'étant plus d'actualité (PDO pour interactions BDD, POO, ...)	3 heures
<b>E.2 – Outils tiers</b>				
E.2.1 – Les bibliothèques/frameworks utilisés sont pertinents et réellement utiles	Visual Studio Code	Analyse du code, on vérifie que les bibliothèques utilisées servent réellement et n'ont pas un trop grand nombre de fonctionnalité pour un usage trop petit	NC – L'unique bibliothèque utilisée était utile en 2006 mais n'est plus utiles actuellement est pris en charge nativement par HTML depuis 2015 (voir figure ci-dessous)	1 minutes
E.2.2 – Les bibliothèques/frameworks utilisés sont dans une version supportée et maintenue	Visual Studio Code	Vérification que la ou les bibliothèques du projet sont encore d'actualité et maintenue(s), si besoin, les mettre à jour	NC – La seule bibliothèque utilisée n'est plus maintenue.	1 heure



### III. Tableau des tâches

Tache de l'audit de qualité logiciel



Recommandations	Priorité	Complexité	Effort	gravité
tutoriel hébergement local	1	1	1	1
changement de langue	2	3	2	2
tests unitaires	2	3	2	1
présence de valeur arbitraire	2	2	2	2
code mort	2	1	1	1
bibliothèque pour la date n'est plus utile maintenant	2	1	1	1
bibliothèque non maintenue	2	1	1	1
mauvaise gestion des cas particulier	3	3	2	2
erreur et warning dans le code	3	2	3	2
non typage	3	1	2	2
indentation	3	1	1	1
code difficile à lire	3	3	2	2
non respect du camel case	3	2	2	1
traduction en PHP au lieu de json	3	3	2	2
documentation	4	1	2	1
fonction trop longue	4	2	2	2
non présence d'exception	4	3	2	3
non présence de try catch (sauf dans la bibliothèque)	4	2	3	3
noms impertinents (variables et fonctions)	4	4	3	3
mettre à jour la version de PHP	4	4	3	3
diagramme cas d'usage	5	3	2	2
mélange requête SQL et code PHP	5	3	2	3
architecture	6	3	3	3
principe SOLID non respecté	6	6	4	3

## IV. Synthèse

La qualité du code a été jugée selon les principes SOLID, les bonnes pratiques de programmation et le PSR (PHP Standards Recommandations). Pour ce faire, des tests sous machine virtuelle vierge étaient effectuées afin de tester l'installation de l'application et le bon fonctionnement du manuel d'installation. Des analyses de qualité de code à l'aide de PHPStan et de Webimpress Coding Standards ont pu être réalisés. Et diverses lectures du code source permettant de déduire les points problématiques du projet (documentation interne, logique, répétitions, etc.)

Un sévère manque d'architecture se faisait présent durant toute l'analyse, impliquant donc une refonte architecturale du projet, ce qui nous permettra par la même occasion de respecter les principes SOLID qui se font inexistant sur tout le projet. L'usage de bases de données MySQL dans le projet se fait non seulement de façon non sécurisée mais aussi ne prenant pas en compte les éventuelles exceptions en plus de ne pas être testé.

## V. Annexe : Nouvel UML du projet

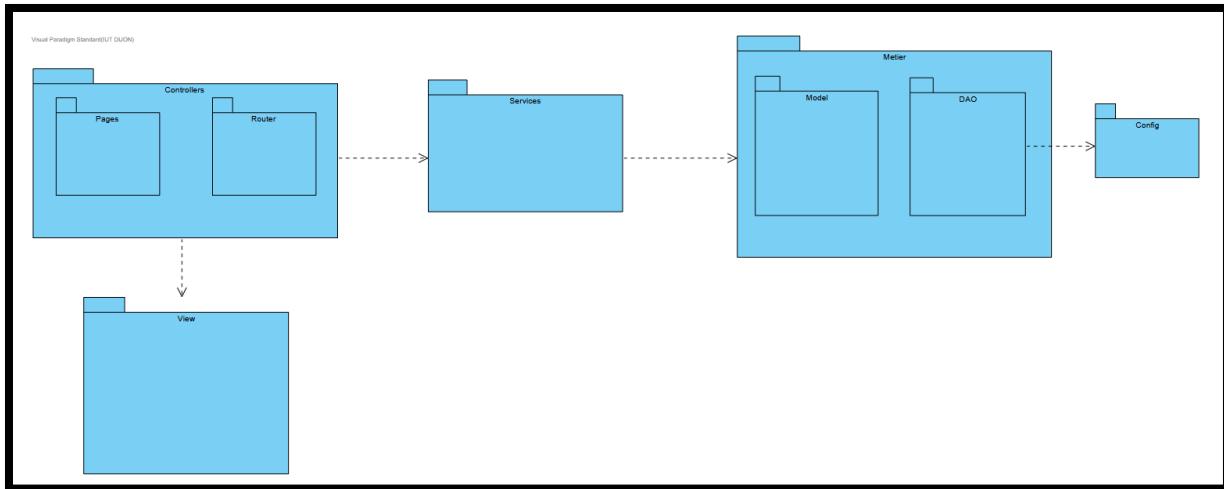


Figure 4 : Diagramme des packages

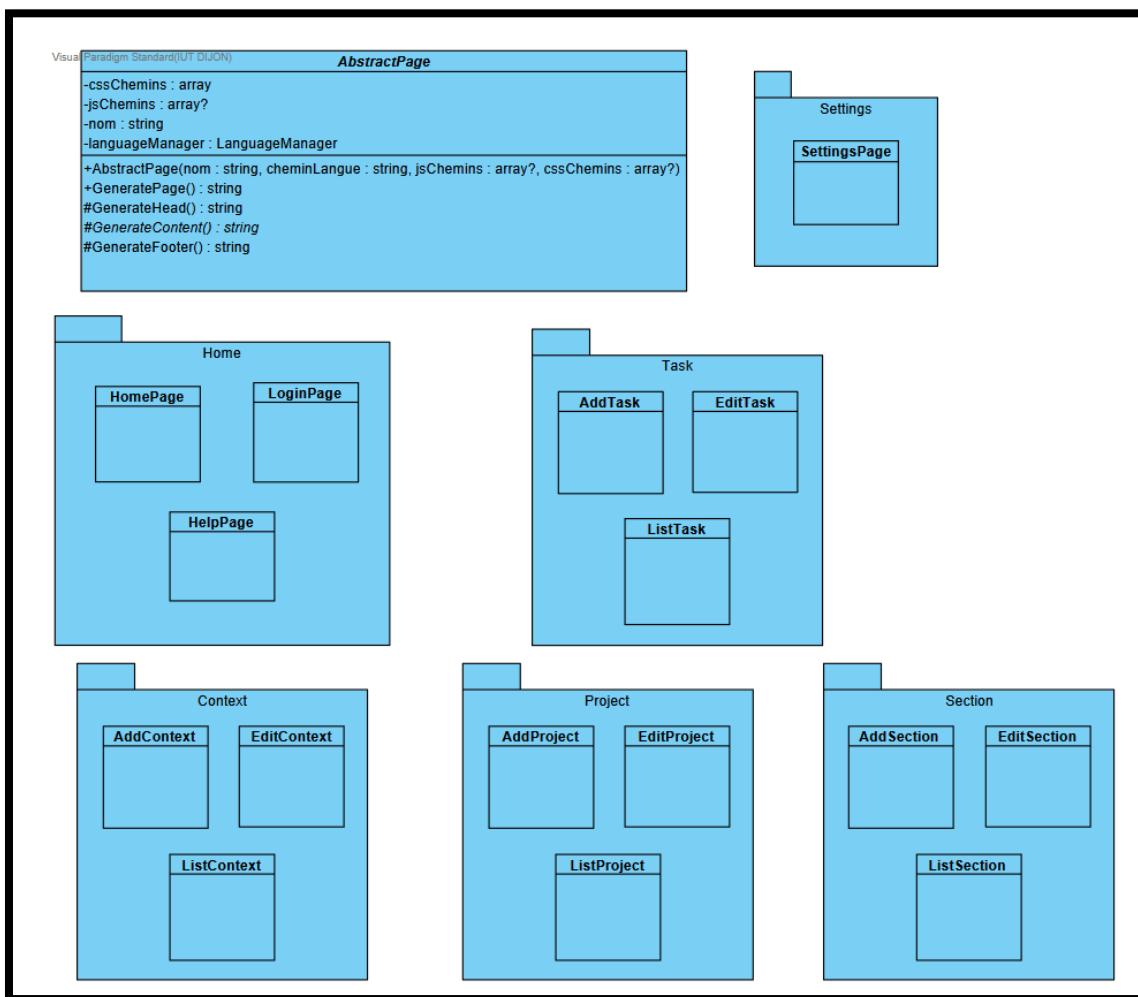


Figure 5 : UML de la couche Vue

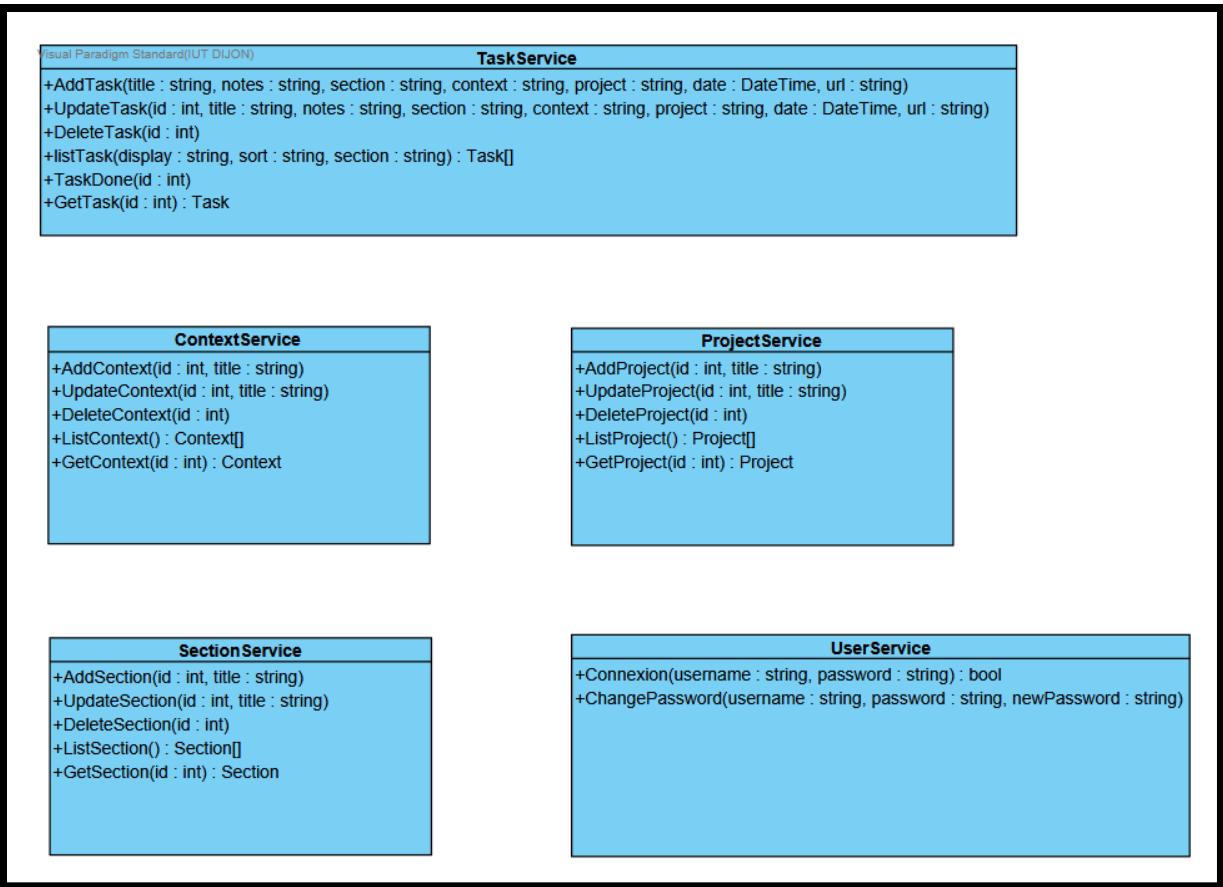


Figure 6 : UML de la couche Service

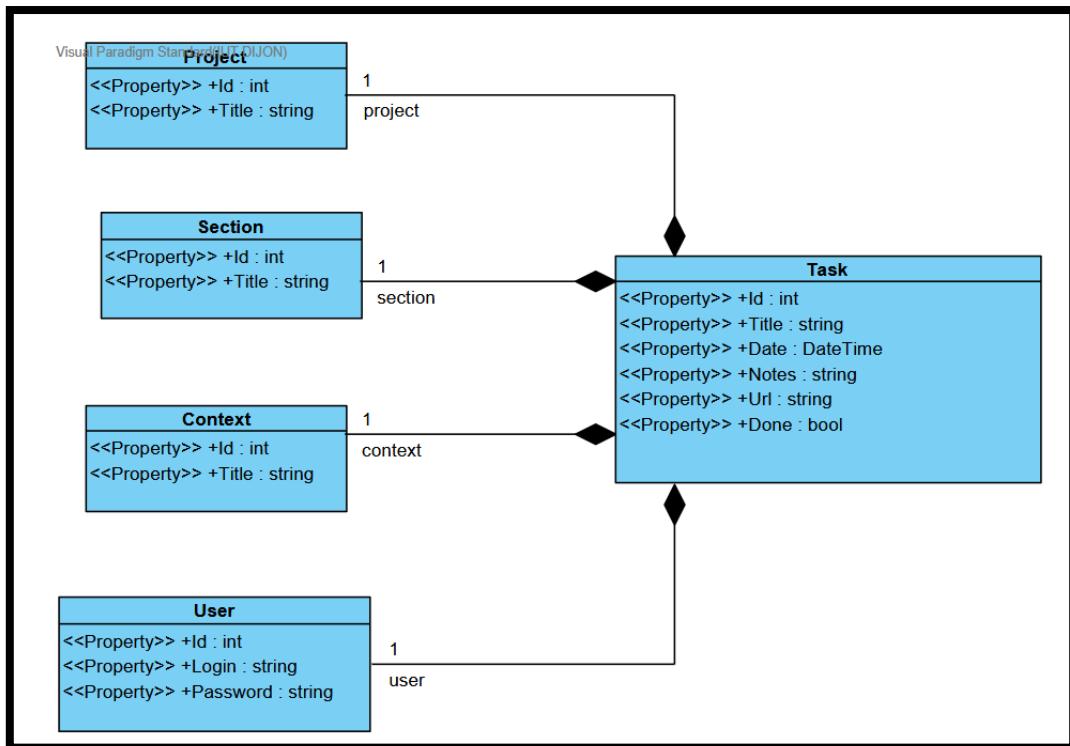


Figure 7 : UML de la couche Model

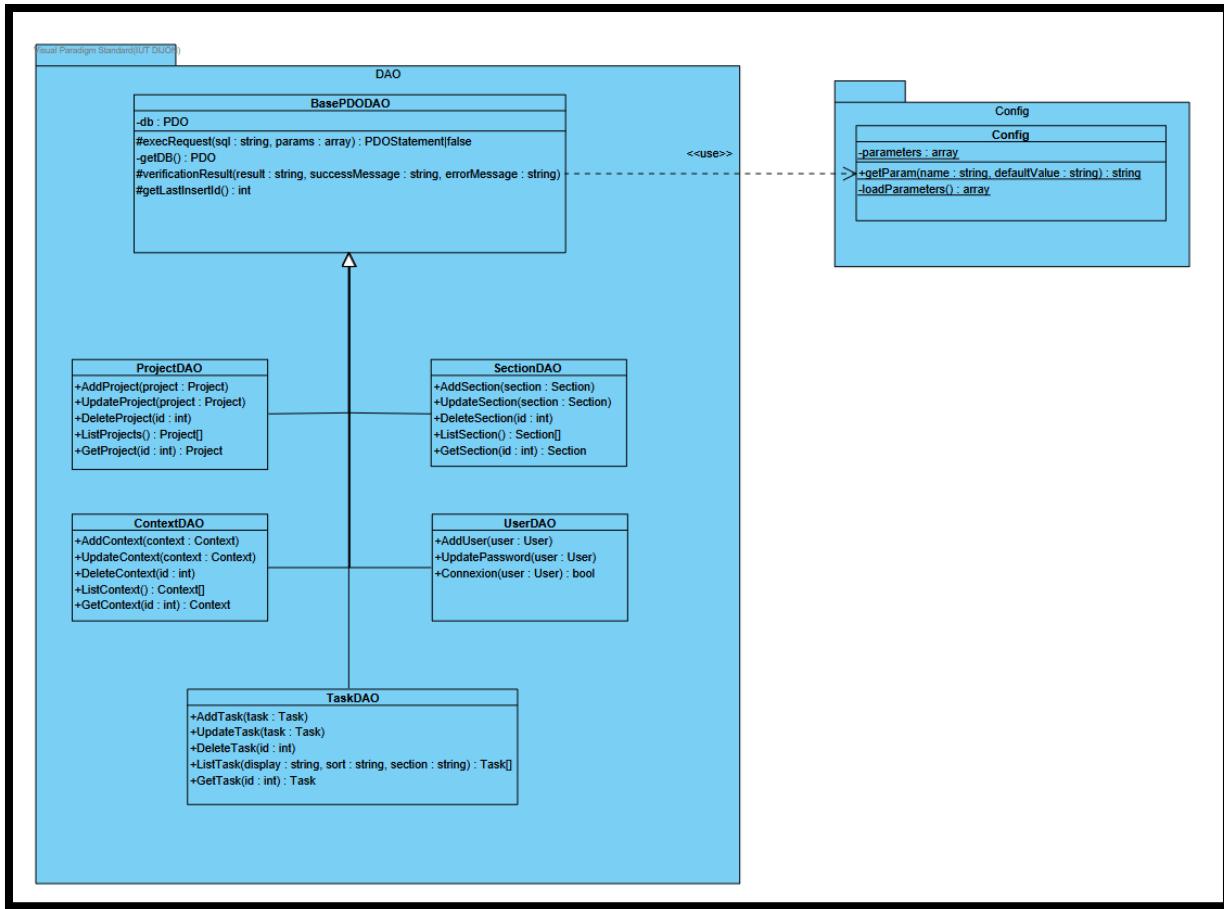


Figure 8 : UML de la couche DAO

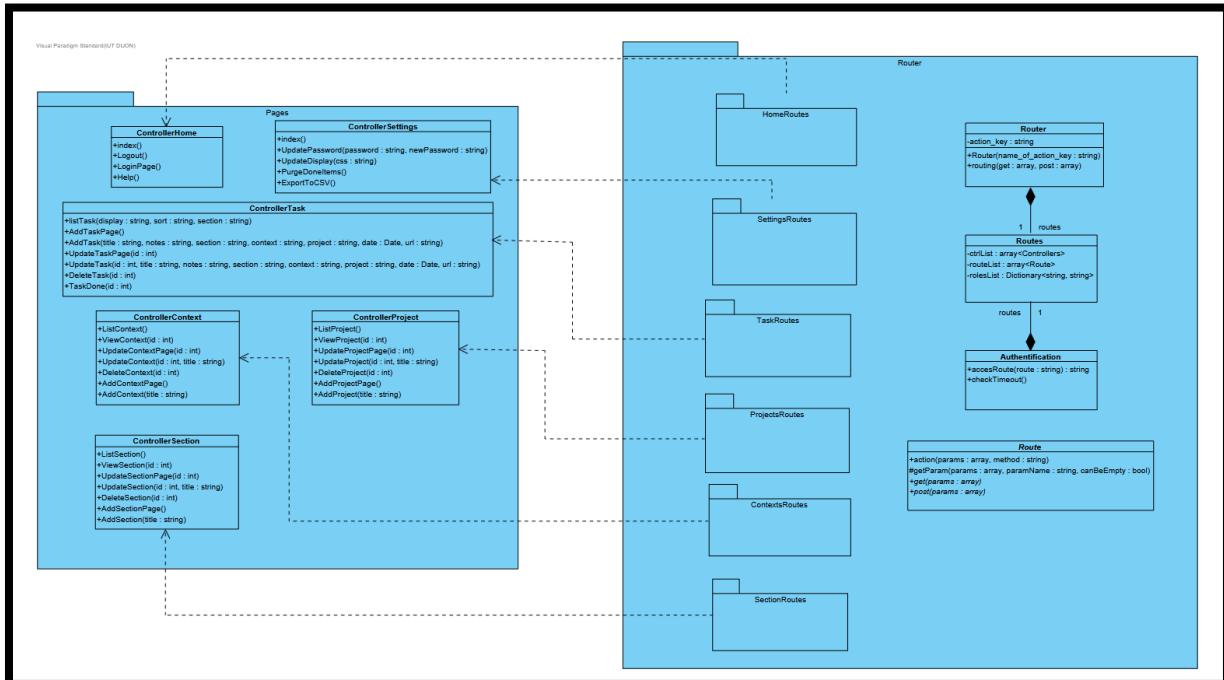


Figure 9 : UML de la couche Controller