

Réalisé par :

FEISTHAUER Simon
SCHOU Lilian
TOPAL Fatih
TOK Mikail



SAÉ 4.01

Placement lors des contrôles au département informatique

Sommaire :

Placement lors des contrôles au département informatique.....	1
Sommaire :.....	2
Introduction.....	3
1. Éléments à garder.....	4
a) Dans l'utilisation, l'interface utilisateur (fonctionnalités).....	4
b) Dans le code (technique).....	5
2. Éléments à modifier.....	6
a) Dans l'utilisation, l'interface utilisateur (fonctionnalités).....	6
b) Dans le code (technique).....	7
3. Schéma entité-association.....	8
4. Schéma relationnel.....	9
5. Programmation future (optionnel).....	10
6. Répartition des tâches.....	11
Conclusion.....	12

Introduction

Dans le cadre de cette SAE, nous allons présenter une analyse critique de l'application existante afin de proposer une trajectoire de développement cohérente.

Notre étude identifie les parties logiciels à conserver, les failles et techniques à corriger, ainsi que la stratégie de refonte adoptée par l'équipe pour transformer cette application en une solution robuste et sécurisée.

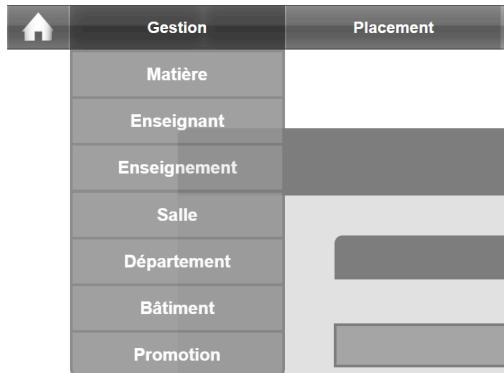
1. Éléments à garder

a) Dans l'utilisation, l'interface utilisateur (fonctionnalités) - FEISTHAUER Simon

Pour la partie Interface, il faut faire une refonte majeure du site que ce soient les couleurs du site au global dans chaque page mais aussi les boutons qui sont peu intuitifs à cause de leurs interface graphique donc il n'y a pas grand-chose à garder à ce niveau-là.

Pour la partie Utilisation :

Ses catégories sont efficaces car elles permettent de séparer la configuration (Gestion) de l'action (Placement).



Dans la partie « Gestion » :

Gestion
Matière
Enseignant
Enseignement
Salle
Département
Bâtiment
Promotion

Pour chaque page (chaque page de Gestion a exactement la même structure) de la partie Gestion il faut garder cette lisibilité pour voir les informations par exemple dans la page Matière on voit directement quelle matière et assignée à quelle promotion.

Il faudrait également garder le fait qu'on puisse directement dans la page créer un nouvel élément associé à la page.

Un autre point fort vient du fait qu'on voit directement les boutons à droite de la ligne pour soit supprimer ou modifier une ligne existante. (Attention cependant on peut aussi trouver une autre solution pour éviter la répétition de bouton en ne mettant qu'une seule fois modifier et supprimer pour toutes les matière/promotion.)

Matière	Promotion		
Éco-Gestion-Droit	BUT1 INFO 2025		
Informatique	BUT1 INFO 2025		
Mathématiques	BUT1 INFO 2025		
Éco-Gestion-Droit	BUT2 INFO 2025		
Informatique	BUT2 INFO 2025		
Mathématiques	BUT2 INFO 2025		
Éco-Gestion-Droit	BUT2 Passerelle INFO 2025		
Informatique	BUT2 Passerelle INFO 2025		
Mathématiques	BUT2 Passerelle INFO 2025		
Éco-Gestion-Droit	BUT3 INFO 2025		
Informatique	BUT3 INFO 2025		
Mathématiques	BUT3 INFO 2025		

Dans la partie Placement.

En ce qui concerne l'interface mettre les différentes étapes est une bonne idée.

Pour ce qui est de l'utilisation il faut garder tous les champs des différentes étapes car elles sont toutes essentielles.

Étape 1 : Informations obligatoires

Date 11/02/2026

Début du devoir Heure Minute Durée Heure Minute

Promotion INFO BUT1 Tous les groupes Matière Salle

Enfin, dans la partie placement des étudiant il faut garder cette gestion des place qui est vraiment bien faite l'interface est claire la seul choses à vraiment modifier c'est la partie séparation des rangs qui n'est pas très intuitif mais sinon le reste est à garder.

b) Dans le code (technique) - TOK Mikail

Concernant la partie technique, il y a plusieurs bons points à garder tout d'abord bien qu'il y ait beaucoup de répétitions et beaucoup de fichier, l'application reste globalement fonctionnelle. Ce qui signifie que la logique du code est bonne il suffirait donc plus d'optimiser ses fichiers existant.

Analyse de la base de données

L'analyse de la base de données révèle que l'ensemble des tables respecte la deuxième forme normale (2FN): les valeurs sont atomiques et chaque colonne dépend de l'intégralité de sa clé primaire. Cette conformité nous permet de réutiliser la structure actuelle comme une base solide (nous évitant ainsi de devoir tout reprendre de zéro).

Analyse du code

Malgré la nécessité d'une refonte, plusieurs briques de l'application actuelle seront conservés pour la suite du projet.

- Accès aux données : L'utilisation de l'interface PDO est maintenu tout comme la centralisation de la connexion dans un fichier unique ([connexion.php](#)).
- Sécurité SQL : Le recours aux requêtes préparées (notamment dans [gest_mat](#))



```
5 v try {
6   // Ajout d'un élément à la liste d'ajout
7   $stmt = $pdo->prepare('SELECT COUNT(*) as a FROM matiere WHERE nom_matiere = :nom_matiere AND id_promo = :id_promo');
8   $stmt->execute(['nom_matiere' => $_POST['nom_matiere'], 'id_promo' => $_POST['id_promo']]);
9   $countex = $stmt->fetch(PDO::FETCH_ASSOC);
10  // Toute ici la developpement système doit
```

- Fiabilité du code : La gestion des erreurs via des blocs [try-catch](#)



```
5 v try {
6   // Connexion à la base de données avec PDO
7   $dsn = 'mysql:host=localhost;dbname=testsae;charset=utf8';
8   $username = 'root';
9   $password = '';
10  // Options pour gérer les erreurs et s'assurer de la compatibilité UTF-8
11  $options = [
12    PDO::ATTR_ERRMODE => PDO::ERRMODE_EXCEPTION, // Pour lever une exception en cas d'erreur
13    PDO::ATTR_DEFAULT_FETCH_MODE => PDO::FETCH_ASSOC, // Pour obtenir des résultats sous forme de tableau associatif
14    PDO::MYSQL_ATTR_INIT_COMMAND => "SET NAMES utf8" // Pour s'assurer que les requêtes sont en UTF-8
15  ];
16
17  // Création de la connexion PDO
18  $pdo = new PDO($dsn, $username, $password, $options);
19
20 //echo "Connexion réussie !";
21 v } catch (PDOException $e) {
```

L'architecture actuelle possède une logique métier claire que nous souhaitons conserver ([import.php](#), [gest_promo.php](#), [gest_grp.php](#), etc)

La structure modulaire (un fichier par gestion, ex: [gest_mat](#), [gest_ens](#)) est pertinente et facilite la navigation dans le code.

Nous conserverons également l'intégralité de la logique liée à l'export PDF (`export_pdf.php`, `x_fctPdfSalle.php`) et les bibliothèques comme FPDF et EzPDF.

Tous les assets frontend (`css`, `javascript`, `images`) ainsi que les utilitaires (`util_placement`) seront réintégrés.

2. Éléments à modifier

a) Dans l'utilisation, l'interface utilisateur (fonctionnalités) - SCHOU Lilian

- Page de connexion et barre de navigation :

Le premier contact avec l'application révèle des lacunes très importantes en termes d'autonomie et de clarté. Tout d'abord, l'application démarre sur une page de connexion. Cependant, il n'y a pas de bouton pour pouvoir se créer un compte. La possibilité de se créer un compte directement depuis l'application (en vérifiant bien entendu pendant cette phase que l'utilisateur fait bien partie des enseignants et de l'université de Lorraine) permettrait de rendre l'application plus autonome et indépendante.

Service d'authentification



Ensuite, dans la barre de navigation, il y a un bouton de déconnexion. Cependant, il n'y a aucune indication sur l'utilisateur actuellement connecté. Il serait donc judicieux d'ajouter le nom d'utilisateur à l'intérieur de ce bouton (exemple : afficher "Se déconnecter de ..." au lieu de juste "Déconnexion") ou bien de l'afficher dans la barre de navigation à gauche du bouton "Déconnexion".



Un système de déconnexion automatique au bout d'un certain temps serait très important au niveau de l'ergonomie (et de la sécurité, voir plus loin). En effet, l'absence de déconnexion automatique représente une faille ergonomique et sécuritaire majeure. Sur des postes partagés, cela expose les données du précédent utilisateur au suivant sans que ce dernier n'en soit conscient.

- Étape 1 :

Le flux de travail souffre de comportements incohérents et d'un manque de retour d'information (feedback). Pour commencer, une erreur avec un message incompréhensible pour l'utilisateur survient la première fois qu'un groupe d'étudiants est ajouté à un devoir :

localhost:8888 indique

```
erreur : <br />
<b>Warning</b>: Undefined array key "nbPU1" in <b>
Applications/MAMP/htdocs/WebBUT2/sae_4.01/Placement/
util_placement/fct_placement.php</b> on line <b>486</b><br /
>
<br />
<b>Warning</b>: Undefined array key "nbPU1" in <b>
Applications/MAMP/htdocs/WebBUT2/sae_4.01/Placement/
util_placement/fct_placement.php</b> on line <b>487</b><br />
```

OK

Lorsque l'on clique sur le bouton « Plus », des données semblent être ajoutées. En effet, le message d'erreur « Nombre maximal de matière atteint » apparaît alors à chaque fois que l'utilisateur essaye d'ajouter le même groupe d'étudiant, confirmant bien que malgré le message d'erreur incompréhensible vu précédemment, les données ont tout de même bien été ajoutées. Cependant, il n'y a aucune indication que ces données ont été saisies :

The screenshot shows a web-based form titled "Placement". At the top, it says "Étape 1 : Informations devoir". Below this, there are several input fields: "Date" set to "18/02/2026", "Début du devoir" set to "10:00", "Durée" set to "6:00", "Promotion" dropdown set to "INFO BUT2", "Groupe" dropdown set to "Groupe 1", "Informatique" dropdown, "Amphi A" dropdown, and a "+" button. At the bottom right of the form area, there is a "Suivant" button.

De plus, un bouton pour réinitialiser ces données serait alors manquant. L'utilisateur ne dispose d'aucun moyen simple pour vider une saisie erronée, rendant le processus rigide.

Il m'a également été possible de créer des devoirs à des dates déjà passées, ce qui est ne devrait pas être réalisable.

Une fois la première erreur incompréhensible corrigée, le groupe d'étudiant a pu être ajouté et j'ai pu voir un tableau avec des colonnes. Cependant, il est assez difficile de savoir au début à quoi correspondent réellement ces colonnes. Il serait alors une bonne idée d'ajouter plus d'informations en en-tête de ces colonnes :

E36	20 / 40
INFO BUT1	2
Informatique	23
X	

Le bouton de suppression des informations (en forme de croix) pourrait également être coloré en rouge pour se démarquer des autres et accentuer l'importance de son action.

- Étape 2 :

Après avoir cliqué sur le bouton « Suivant » à l'étape 1 et juste avant que le tableau des places ne s'affiche à l'étape 2, il arrive parfois (pas toujours) qu'il y ait un court instant de quelques secondes pendant lequel rien ne sera affiché, ou alors des données seront partiellement affichées. L'application pourra sembler alors figée pour l'utilisateur, ce qui pourrait être résolu simplement en ajoutant un petit logo de chargement pendant que toutes les données n'auront pas été chargées.

Le principe de sélectionner deux places puis d'appuyer sur un bouton pour les intervertir pourrait être simplifié en une méthode de glisser-déposer des cases pour les réorganiser, ce qui ferait gagner en temps, en ergonomie et en modernité.

J'ai également remarqué que le bouton « Suivant » pouvait parfois ne pas apparaître, rendant ainsi l'interface assez instable.

- Étape 3 :

La phase finale de l'application, censée délivrer le résultat du travail, est défaillante. Il y a des erreurs à l'étape 3, bloquant complètement son utilisation et rendant l'application totalement inutile :

Placement

Étape 3 : Exportation

```
Fatal error: Uncaught PDOException: SQLSTATE[22007]: Invalid datetime format: 1292 Incorrect date value: '2026-2/-10' for column 'e27503u_infoplacement`.`devoir`.`date_devoir` at row 1 in
/Applications/MAMP/htdocs/WebBUT2/sae_4.01/Code modifié par
Fatih/placement/Placement/util_placement/up_stage3.php:15 Stack trace: #0
/Applications/MAMP/htdocs/WebBUT2/sae_4.01/Code modifié par
Fatih/placement/Placement/util_placement/up_stage3.php(15): PDOStatement->execute(Array) #1
/Applications/MAMP/htdocs/WebBUT2/sae_4.01/Code modifié par
Fatih/placement/Placement/util_placement/up_stage3.php(69): saveDevoir() #2 {main} thrown in
/Applications/MAMP/htdocs/WebBUT2/sae_4.01/Code modifié par
Fatih/placement/Placement/util_placement/up_stage3.php on line 15
```

[Précédent](#)

Ce problème vient très probablement d'un problème de configuration de la base de données présente sur le serveur DevWeb car en utilisant une base de données sur un serveur local, le problème n'apparaît plus :

Placement

Étape 3 : Exportation

	Liste	Feuille d'émarginement	Plan
Amphi 3			

[Précédent](#)

Le système de conversion en fichier PDF ne fonctionne malgré tout pas (malgré les indications de modifications de paramètres de navigateur présentes sur la page d'accueil de l'application). En effet, peu importe lequel des trois boutons l'on presse, des erreurs (différentes) s'afficheront :

Une fois les erreurs corrigées et le fichier téléchargé, le lecteur de PDF de l'appareil indique que le fichier ne peut pas être ouvert.

En complément, des erreurs avec des balises
 peuvent apparaître fréquemment et certains boutons ne s'affichent pas toujours correctement.

À l'instar de l'analyse, il est clair que l'interface souffre de problèmes de conception profonds. Le manque de validation des entrées (dates), l'absence de gestion d'état de chargement et l'instabilité des fonctions d'exportation confirment que l'application nécessitera une refonte globale sur le plan de l'expérience utilisateur.

b) Dans le code (technique) - TOPAL Fatih

Tout d'abord, le premier élément dans le code qui nous interpelle est l'organisation des fichiers :

```
✓ PLACEMENT
  > css
  > ezpdf
  > fpdf186
  > images
  > javascript
  > util_placement
  ↗ connexion.php
  ↗ content.php
  ↗ crea_salle.php
  ↗ cs_stage1.php
  ↗ cs_stage2.php
  ↗ cs_stage3.php
  ↗ cs_stage4.php
  ↗ deconnexion.php
  ↗ etudiant.php
  ↗ export_pdf_ens.php
  ↗ export_pdf.php
  ↗ fct_pdf.php
  ↗ gest_bat.php
  ↗ gest_dpt.php
  ↗ gest_ens.php
  ↗ gest_ensmat.php
  ↗ gest_grp.php
  ↗ gest_mat.php
  ↗ gest_mat2.php
  ↗ gest_promo.php
  ↗ gest_salle_old.php
  ↗ gest_salle.php
  ↗ home.php
  ↗ import.php
  ↗ index.php
  ↗ login.php
  ↗ modif_salle.php
  ↗ ms_stage1.php
  ↗ ms_stage2.php
  ↗ ms_stage3.php
  ↗ ms_stage4.php
  ↗ ms_stage5.php
  ↗ ms_stage6.php
  ↗ ptrim.php
```

Cette organisation n'est pas du tout intuitive.

→ Un fichier css, qui est un des seuls éléments dont on comprend l'utilité

→ Un fichier nommé par le langage utilisé, qui n'est pas du tout une "bonne" idée car ne donne aucune indication autre.

→ Des fichiers PHP, en désordre, avec certains comme "connexion.php" dont on comprend l'utilité.

→ D'autres comme ces fichiers "cs_*" qui ne nous donne aucune information sur leur contenu ou leur fonction. Et ils sont numérotés, ce qui est très étrange.

→ Les fichiers sont tous nommés en camel_case, ce qui ne pose pas de soucis pour les fichiers php mais devrait plutôt être remplacé par du kebab-case ou du lowercase pour les fichiers javascript.

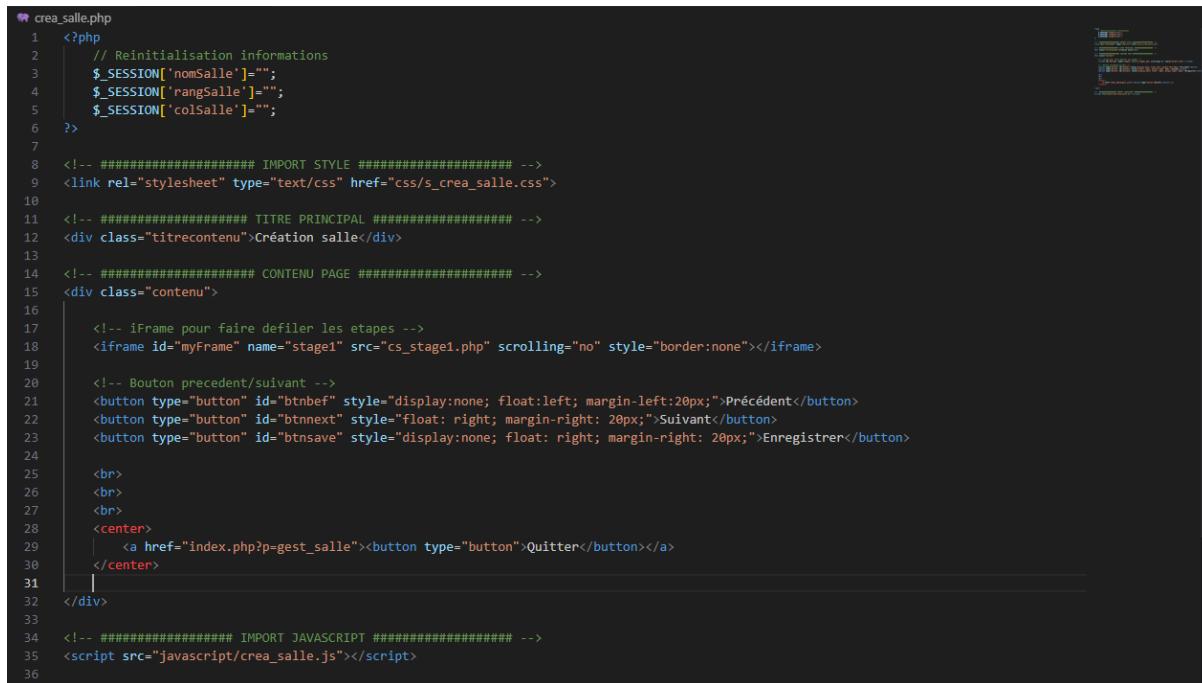
→ on voit un fichier _old, ce qui laisse supposer que ce fichier n'est plus utile, et qui est tout de même resté dans le code, ce qui pollue l'application.

Enfin, des fichiers commençant par x_, ce qui est totalement incompréhensible.

```
↗ request_s3.php
↗ util_placement.php
↗ visu_salle.php
↗ x_fctPdfSalle.php
↗ x_salle.php
```

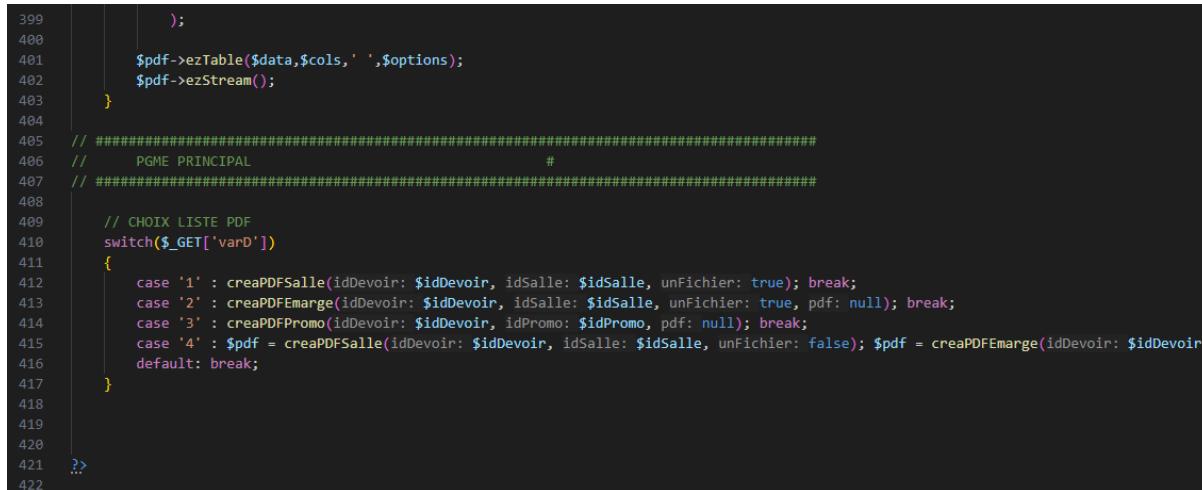
Si on commence à ouvrir les fichiers, on va tomber sur plusieurs types de fichiers :

- ceux qui font une trentaine de lignes comme crea_salle.php :



```
 crea_salle.php
1 <?php
2 // Reinitialisation informations
3 $_SESSION['nomSalle']=""; 
4 $_SESSION['rangSalle']=""; 
5 $_SESSION['colSalle']=""; 
6 ?>
7
8 <!-- ##### IMPORT STYLE ##### -->
9 <link rel="stylesheet" type="text/css" href="css/s_crea_salle.css">
10
11 <!-- ##### TITRE PRINCIPAL ##### -->
12 <div class="titrecontenu">Création salle</div>
13
14 <!-- ##### CONTENU PAGE ##### -->
15 <div class="contenu">
16
17 <!-- iFrame pour faire defiler les etapes -->
18 <iframe id="myFrame" name="stage1" src="cs_stage1.php" scrolling="no" style="border:none"></iframe>
19
20 <!-- Bouton precedent/suivant -->
21 <button type="button" id="bttnbef" style="display:none; float:left; margin-left:20px;">Précédent</button>
22 <button type="button" id="bttnnext" style="float: right; margin-right: 20px;">Suivant</button>
23 <button type="button" id="bttnsave" style="display:none; float: right; margin-right: 20px;">Enregistrer</button>
24
25 <br>
26 <br>
27 <br>
28 <center>
29 | <a href="index.php?p=gest_salle"><button type="button">Quitter</button></a>
30 </center>
31
32 </div>
33
34 <!-- ##### IMPORT JAVASCRIPT ##### -->
35 <script src="javascript/crea_salle.js"></script>
36
```

- et ceux qui ont font 422 comme export_pdf



```
399
400
401     );
402
403     $pdf->ezTable($data,$cols, ' ', $options);
404     $pdf->ezStream();
405 }
406
407 // ##### PGME PRINCIPAL #####
408 // #####
409
410 // CHOIX LISTE PDF
411 switch($_GET['varD']){
412 {
413     case '1' : creaPDFSalle(idDevoir: $idDevoir, idSalle: $idSalle, unFichier: true); break;
414     case '2' : creaPDFEmarge(idDevoir: $idDevoir, idSalle: $idSalle, unFichier: true, pdf: null); break;
415     case '3' : creaPDFPromo(idDevoir: $idDevoir, idPromo: $idPromo, pdf: null); break;
416     case '4' : $pdf = creaPDFSalle(idDevoir: $idDevoir, idSalle: $idSalle, unFichier: false); $pdf = creaPDFEmarge(idDevoir: $idDevoir,
417     default: break;
418 }
419
420 ?>
421
```

Le nom de ce deuxième fichier laisse supposer qu'il fait à lui seul une partie des fonctionnalités d'exportation pdf, cependant il aurait été judicieux de soit le mettre dans un dossier et indiqué que ce n'est pas un élément qui est fait partie de l'application en soit, ou le diviser en plusieurs sous-fichiers pour permettre une meilleure lecture à une personne qui ne connaît pas l'application ou du moins cette fonctionnalité.

Il existe un autre problème de "structure" dans les fichiers : aucun modèle n'est utilisé dans la structure des fichiers. Il aurait peut être été judicieux de passer sur un modèle Model-View-Controller et ne pas tout mélanger dans chaque fichier, car sinon il est très difficile d'apporter des modifications ou de faire des tests. Cela est valable pour tous les fichiers.

Il n'y a également aucune séparation front/back end dans le code en général.

De plus, on voit à certains endroits que `$_SESSION` est utilisé comme variable global , comme dans ce fichiers cs_stage4.php :

```
for($i=0; $i<$_SESSION['rangSalle']; $i++)
{
    for($j=0; $j<$_SESSION['colSalle']; $j++)
    {
        if($_SESSION['structSalle'][$i][$j]=='1')
        {
            $placeOk++;
        }
        else if($_SESSION['structSalle'][$i][$j]=='2')
        {
            $placeHandi++;
        }
    }
}
```

Ce qui nous pose un problème notamment dans la traçabilité des données.

Du côté de la sécurité, on peut voir plusieurs failles, notamment dans le fichier export_pdf.php ou des variables GET sont directement dans les requêtes :

```
function creaPDFEmarge($idDevoir, $idSalle, $unFichier, $pdf): mixed
{
    $querySalle=mysql_query(query: "SELECT nom_salle FROM salle WHERE id_salle=$idSalle");
    $nomSalle=mysql_result(result: $querySalle, row: 0);      Function 'mysql_result' has be
```

C'est une porte ouverte aux injections SQL.

Pour ce qui est des variables GET et POST, il y a souvent aucune vérification dans le code, ce qui rend l'application très vulnérable (XSS, ...).

Et enfin, évidemment l'utilisation de MD5 est une sécurité faible, considéré comme non sûr depuis 2013.

Pour ce qui concerne de l'utilisation des id (très frappant notamment sur la base de donnée) on voit un énorme problème : ils sont utilisés partout.

C'est l'élément principal faisant que le code est vraiment incompréhensible, d'autant plus qu'il n'y a aucune documentation ou commentaire dans le code nous permettant de comprendre ne serait-ce qu'un peu ce à quoi ils correspondent.

<code>id_salle</code>	<code>id_devoir</code>	<code>id_groupe</code>	<code>id_mat</code>	Extrait de la table <code>devoir_groupe</code>
63	1335	6716	230	
63	1360	6717	230	
63	1361	6718	230	

Mais ce n'est pas le seul problème, il y a également les multiples fichiers, ou morceaux de codes abandonnées dans le code, comme les fichiers _old ou _copy évoqués en première partie de l'analyse, ou ici dans etudiant.php :

```
<div id="bloctitreetud"><div class="nomtitreetud">Nom</div><div class="nomtitreetud">Prénom</div><div class="nomtitreetud">Tiers-temps</div>
<?php
    $stmt1 = $pdo->prepare(query: 'SELECT * FROM groupe WHERE id_promo = :promo ORDER BY nom_groupe');
    $stmt1->execute(params: ['promo' => $promo]);
    while ($tab1 = $stmt1->fetch(mode: PDO::FETCH_ASSOC)) {
        $stmt2 = $pdo->prepare(query: 'SELECT * FROM etudiant WHERE id_groupe = :id_groupe ORDER BY nom_etudiant, prenom_etudiant');
        $stmt2->execute(params: ['id_groupe' => $tab1['id_groupe']]);
        while ($tab = $stmt2->fetch(mode: PDO::FETCH_ASSOC)) {
            /*$query1=mysql_query('SELECT * FROM groupe WHERE id_promo='.$promo.' ORDER BY nom_groupe');
            while ($tab1=mysql_fetch_array($query1)){
                $query=mysql_query('SELECT * FROM etudiant WHERE id_groupe='.$tab1[id_groupe].'' ORDER BY nom_etudiant, prenom_etudiant');
                while($tab=mysql_fetch_array($query)){
                    */
        }
    }
?>           <div class="contenutabetud">
```

15% de ce fichier est constitué de code abandonné.

Un autre élément frappant dans ce code est le fait qu'il n'y a pas d'encapsulation, tout est global. Utiliser des classes nous aurait aidé à mieux comprendre le code, et de pouvoir suivre le flux d'exécution.

Enfin, nous pouvons remarquer qu'il n'y a pas de fichier test, mais dans ce cas il y a deux choses à considérer :

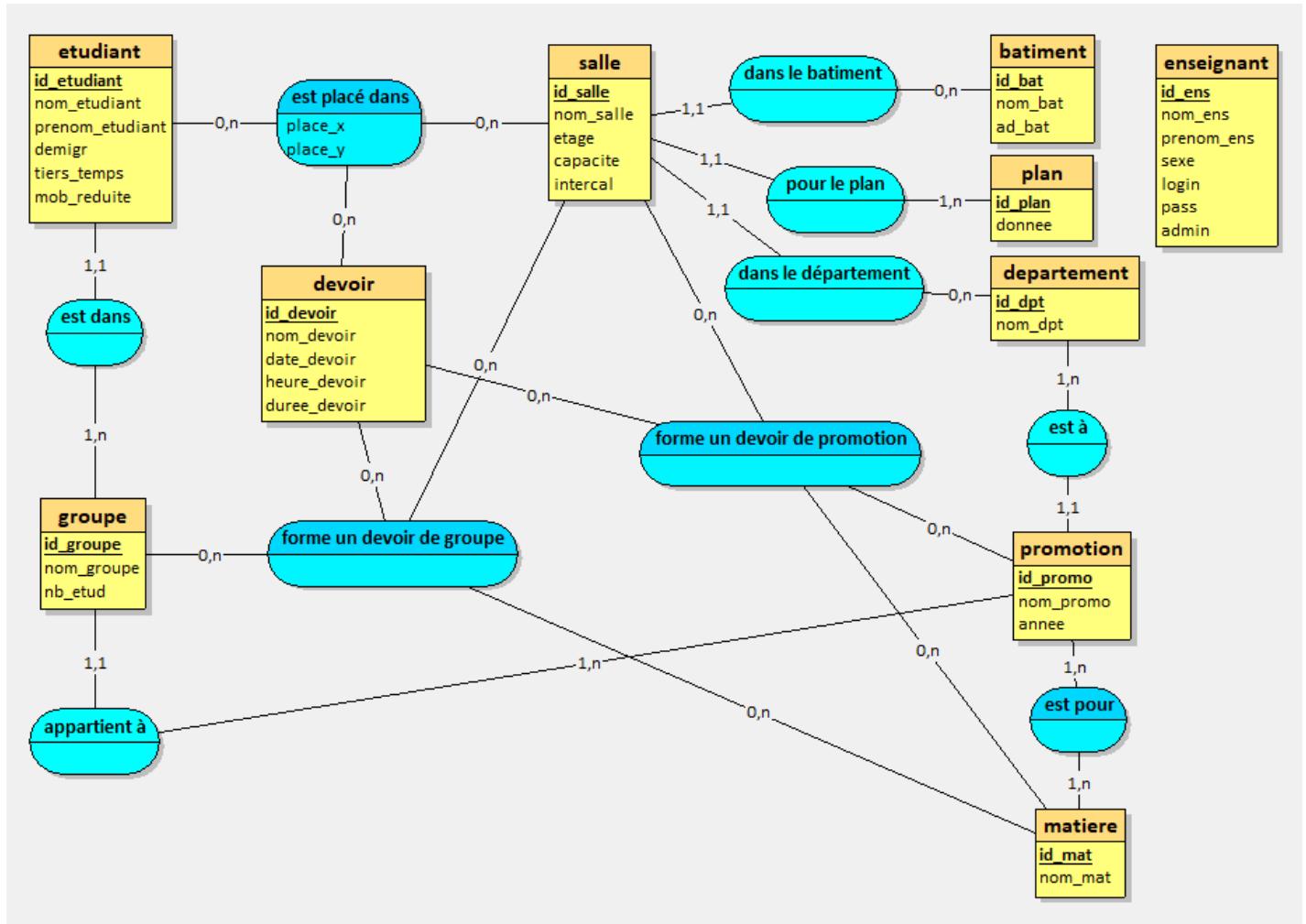
- Soit ils ont volontairement été retirés car il s'agit d'une version finale de l'application, dans ce cas le fichier connexion.php ne devrait pas non plus être là car dans la version finale on ne met pas les identifiants de la base de données.
- Soit il n'y a pas de test unitaire, ce qui rend les bugs impossibles à être détectés et nous laisse douter de la qualité du code (car aucune preuve).

En conclusion, sur les principes SOLID, le code ne respecte presque aucun principe et parfois même très gravement (principe d'une seule responsabilité avec les fichiers à responsabilités multiples, le principe de la dépendance inversible avec les dépendances directes, ...).

Tous ces éléments dans le code devront être pris en compte dans le nouveau code et seront à modifier.

3. Schéma entité-association

Note : les tables surveille et enseigne ne sont pas représentées car elles ne sont strictement pas utilisées.

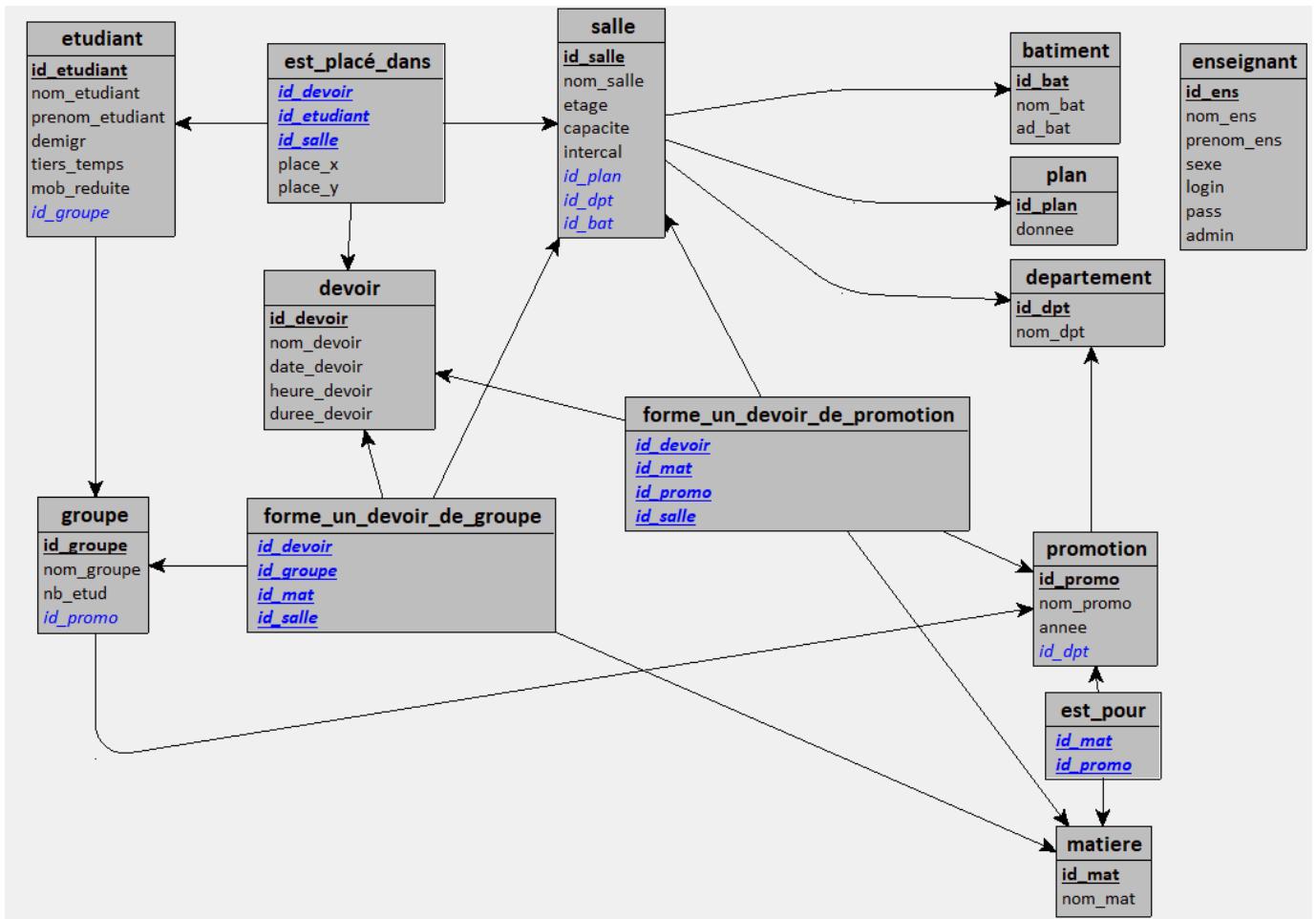


La table enseignant n'est pas reliée, comme dans la base de données.

On voit bien que certains éléments ne sont pas bien réfléchis, notamment les devoirs qui pourraient simplement faire absorber des clés comme id_mat ou id_salle au devoir directement, car on risque d'avoir des problèmes d'id des devoirs.

De plus, la table enseignant pourrait simplement être lié à un devoir.

4. Schéma relationnel



5. Répartition des tâches

Membre	Tâches
TOPAL Fatih	Refonte BDD & Qualité ⇒ Nettoyer le code mort & optimiser ⇒ Mettre en place une documentation technique ⇒ Remplacer MD5 ⇒ Planifier l'implémentation de nouvelles fonctionnalités suite aux interviews
SCHOU Lilian	UX/UI ⇒ Refonte de la barre de navigation ⇒ Ajout de la fonctionnalité de création de compte ⇒ Ajouts d'indicateurs de chargement et de feedbacks d'erreurs ⇒ Implémentation d'idées au cours du redéveloppement
FEISTHAUER Simon	Frontend & Log. Métier ⇒ Refonte graphique ⇒ Correction des fonctionnalités posant problème ⇒ Refonte répartition des boutons

TOK Mikail

Backend et sécurité

- ⇒ Rendre le code maintenable et réutilisable.
- ⇒ Mettre en place un modèle MVC.
- ⇒ Protection des injections SQL via la généralisation des requêtes préparées.
- ⇒ Éliminer les vulnérabilités critiques (XSS, ...).
- ⇒ Corriger le système de conversion en pdf.

Conclusion

Pour conclure, l'analyse de cette application révèle un contraste entre une logique métier qui fonctionne et une réalisation technique fragile.

La base du projet est clairement réutilisable (base de données cohérente, interface pertinente et gestion du placement des étudiants en salle bien réfléchis).

Cependant, il y a aussi beaucoup de problèmes, notamment sur le plan de l'expérience utilisateur (pas d'autonomie, instabilité de certaines fonctionnalités, manque de retour visuels) et sur le plan technique (pas d'architecture structurée, code mort et méthodes de hachage obsolètes (MD5) représentant des problèmes sécuritaires).

L'absence des tests unitaires et les vulnérabilités aux injections SQL dans certaines parties du code sont également à noter.

Notre objectif sera donc de transformer ce prototype en un outil professionnel, sécurisé et surtout intuitif.