Spécifications Techniques Détaillées

PROJET FASTEVAL

	Nom et Fonction	Date et Signature
Préparé par	DANEDE MARIE FONSECA NASCIMENTO MELANIE MOUHOUN YAMINA SOURY MAGDALEINE	09/03/2020
Validé par		

Résumé d'auteur :

Ce document présente les spécifications techniques détaillées du développement du projet TIC réalisé au sein du Master 1 MIAGE (Méthodes Informatiques Appliquées à la Gestion d'Entreprise) à l'Université de Bordeaux.

Le projet est appelé FastEval

Table des matières

1. Description générale

1.1. OBJECTIF

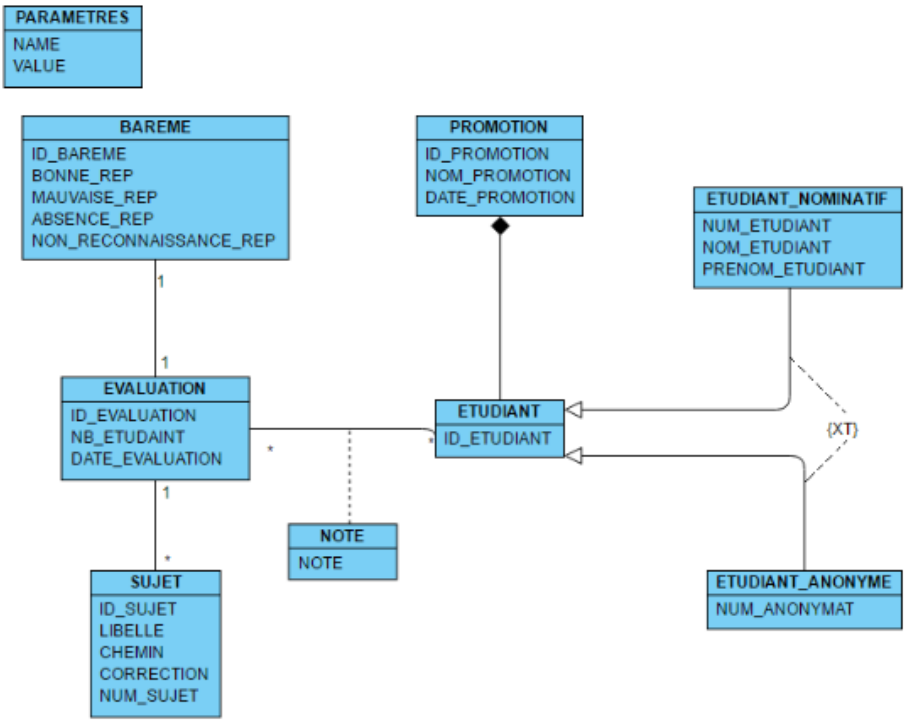
- **Que représente ce document ?**

  ⇨ Ce document représente les spécifications techniques détaillées du projet FASTEVAL proposé par M CATAPOULE au sein du Master 1 MIAGE à l'Université de Bordeaux.

- **Quels sont les documents que vous avez repris ?**

  ⇨ Nous nous sommes basées au cours de nos développements sur le cahier des charges rédigé par nos soins et validé par le client.

1.2. DIAGRAMME DE CLASSES



2. Description détaillée

2.1. INFORMATIONS TECHNIQUES

2.1.1. IDENTIFICATION DU PROGRAMME

Notre programme s'agit d'un site web ayant pour but de simplifier la gestion des évaluations, de la création à la correction. Il permet de :

- Générer un sujet au format .pdf à partir d'un fichier .txt, lui-même généré grâce à un script indépendant de l'application.
- Scanner / importer des copies.
- Corriger un sujet automatiquement, grâce à un barème modifiable.
- Consulter les statistiques et graphiques des notes de la classe.

2.1.2. ACCES

- **Comment peut-on y avoir accès ?**

  ⇨ Le projet est disponible sur GitHub : https://github.com/melaniefonseca/FastEval.git

2.1.3. TECHNOLOGIES UTILISEES

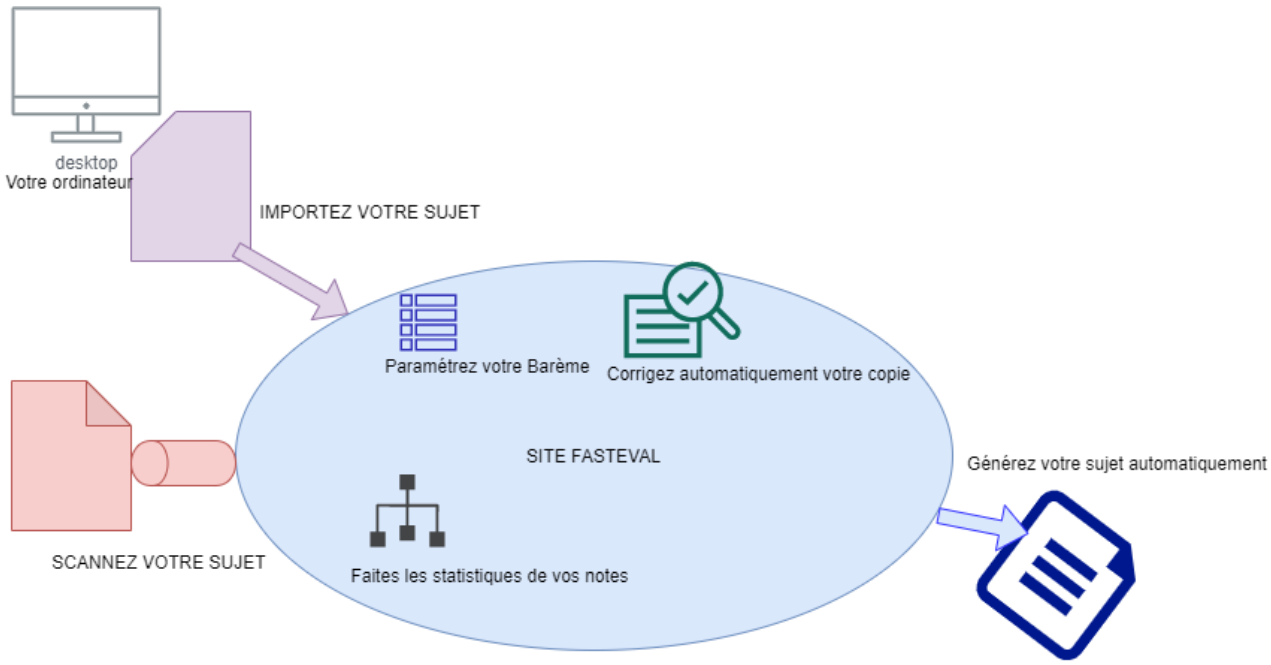
Nous avons utilisé Python pour les scripts de génération de sujet, de mise en page de sujet et de correction. Pour l'application web, les technologies présentent sont le PHP, le HTML et le CSS.

2.1.4. VUES CREES

Ce tableau recense les différentes vues au sein du projet FastEval ainsi que la description de chacune de celles-ci.

Nom de la vue	Description
Présentation	Dans cette page, vous aurez les descriptions détaillées des fonctionnalités du site.
Correction	Vous pouvez, grâce à cette page, scanner vos copies ou les importer depuis votre ordinateur et les corriger automatiquement.
Résultat	Les notes des étudiants sont disponibles sous forme de tableau, en fonction d'une évaluation.
Etudiants	Cette fenêtre permet de créer les étudiants via un formulaire ou via un fichier Excel. Les étudiants peuvent être anonymes (examen) ou nominatifs (contrôle continu). Il est aussi possible, dans cette même page, de créer une promotion à laquelle seront rattachés les étudiants.
FormEtudiantAnonyme	Vous permet de créer un étudiant anonyme (examen) grâce à un formulaire.
FormEtudiantNominatif	Vous permet créer un étudiant nominatif (contrôle continu) grâce à un formulaire.
Sujet	Vous pouvez enregistrer un nouveau sujet (.txt) et générer un fichier (.pdf) correspondant à un des sujets déjà enregistré. Il est également possible de créer une nouvelle évaluation dans cette fenêtre, à laquelle vous pourrez ajouter des sujets.
Statistiques	Une fois vos copies corrigées, vous pouvez voir les statistiques des notes.
Paramètres	Dans 'paramètres' vous pouvez attribuez le barème que vous désirez pour la correction de vos copies.
Plan du site	Vous avez le plan détaillé de l'ensemble du site et divers accès rapides.

2.2. CINEMATIQUE DU TRAITEMENT



2.3. DESCRIPTION DU TRAITEMENT

2.3.1. GENERATION DU SUJET

2.3.1.1. La page sujet

Sur cette fenêtre, est possible d'enregistrer un nouveau sujet. Pour cela, nous avons utilisés des balises <input> et un bouton « valider » qui fait le lien avec une autre page php (nécessaire pour faire des insert).

La case select Fichiers n'accepte que des .PDF :

```
<label > Fichier : </label> <br>
<input type="file" name="nv_sujet" id="nv_sujet" accept=".pdf" value="" multiple=""><br><br>
```

Nous avons utilisé la méthode POST car c'est une création de valeur. A chaque début de formulaire, dans la balise <form>, est inscrit l'action associée :

```
<form method="post" action="../model/EnregistreSujet.php" >
```

Voici ce que contient le model PHP :

```php
<?php

require_once ("../include/PdoFastEval.php");

$pdo = PdoFastEval::getPdoFastEval();
$maxSujet =$pdo->getMaxSujet();

if(!empty($_POST['Valider'])) {
    $idSujet = $maxSujet + 1;
    $libelleunique = $_POST["libelle_sujet"];
    $chemin = $_POST["nv_sujet"];
    $pdo-> insertSujet($idSujet, $libelleunique, $chemin);
}
include("../View/sujets.php");
?>
```

Les $_POST récupèrent les données inscrites dans les inputs. Il est également possible d'ajouter une évaluation avec le même mécanisme.

Ce tableau récapitule les fonctions du Pdo qui ont été utilisées pour cette partie de l'application (codées dans include/PdoFastEval.php).

Nom de la fonction	Description
getLibelle()	Récupération des noms des sujets contenus dans le champ libelle de la table sujet.
getIdSujet()	Récupération des id des sujets contenus dans le champ id_sujet de la table sujet.
getNombresIdSujet()	Récupération du nombre de sujets contenus la table sujet. Utilisation d'un COUNT.
getIdpromotion()	Récupération des id des promotions contenus dans le champ id_promotion de la table promotion.
getIdevaluation()	Récupération des id des évaluations contenues dans le champ id_evaluation de la table evaluation.
getLibellePromotionById($id)	Récupération d'un libelle de promotion contenu dans le champ libelle de la table promotion, en fonction d'un id_promotion.
getLibelleById($id)	Récupération d'un libellé de sujet contenu dans le champ id_sujet de la table sujet, en fonction d'un id_sujet
getDate($evaluation)	Récupération de la date d'une évaluation contenue dans le champ date_evaluation de la table evaluation, en fonction d'un id_evaluation.

2.3.1.2. Le script de génération de sujet

A l'appel du script, plusieurs informations sont demandées à l'utilisateur depuis la console :

- le fichier contenant les questions : il s'agit de la « base de données » contenant toutes les questions, avec leur niveau de difficulté, leur(s) thématique(s)…
- le nom du nouveau fichier qui sera créé,
- les thématiques désirées ; il peut y en avoir plusieurs, une seule ou aucune,
- les difficultés souhaitées ; il peut également y en avoir plusieurs, une seule ou aucune,
- le nombre de questions à générer ; par défaut, il y en a 20.

Le script parcourt alors le fichier contenant les questions pour y trouver les paramètres généraux et construire une liste de toutes les questions. On récupère le paramètre « shuffle » ; s'il est présent, on mélange les questions.

On choisit alors une question au hasard et, si elle répond aux attentes définies à l'appel du script, on l'ajoute à notre liste de nouvelles questions, sinon on la supprime de la liste (pour ne pas avoir à la revérifier). On reproduit ce même procédé jusqu'à ce qu'on ait atteint le nombre de questions demandé. Pour chaque question, on récupère le paramètre « shuffle » ; s'il est présent, on mélange les réponses.

Une fois la liste des nouvelles questions finalisée, on les écrit dans un fichier texte (.txt) portant le nom saisi par l'utilisateur.

Ce fichier est alors utilisable seul (on peut réaliser la mise en forme que l'on souhaite) ou en complément du script de mise en forme.

2.3.1.3. Le script de mise en page du sujet

Tout comme le script précédent, ce dernier demande quelques informations lors de son appel, mais celles-ci sont fournies par le php :

- le fichier qui contient le sujet, généré par le script ci-dessus,
- le nom du nouveau fichier qui sera généré,
- s'il s'agit, ou non, d'une évaluation anonyme,
- le numéro du sujet (si une évaluation possède différents sujets).

Le script parcourt alors le fichier contenant toutes les questions du sujet et y recherche différents paramètres :

- le titre,
- la présentation / consignes,
- le paramètre « columns » pour savoir sur combien de colonnes le sujet est présenté,
- les questions pour les ajouter dans une liste.

On crée ensuite les feuilles de question. Elles contiennent, pour chaque question, l'intitulé et les réponses possibles. En fonction des paramètres définis pour chaque question, les réponses peuvent être affichées de différentes façons :

- les unes en dessous des autres,
- les unes à côté des autres,
- sur un nombre de colonnes défini.

On y retrouve également le nom et le prénom de l'étudiant, ou son numéro d'anonymat.

On génère ensuite les feuilles de réponses. Celles-ci reprennent le numéro de chaque question ayant que l'intitulé de chaque réponse possible. A chaque réponse est associée une case.

Sur chacune de ces feuilles de réponses, on retrouve un encart où l'étudiant doit saisir son numéro étudiant ou son numéro d'anonymat.

2.3.2. SCANNE ET CORRECTION DES SUJETS

2.3.2.1. La page correction

Cette page comporte une liste déroulante. La liste comportera autant de sujet qu'il y a dans la base. On récupère ces valeurs avec la variable identificationSujet. Le Formulaire a une méthode GET. Il récupère donc des données.

```
$identificationSujet= isset($_GET['idsujet']) ? $_GET['idsujet']:false;
```

```
<form method="get"  style="text-align: center;">
```

Pour afficher toutes les évaluations dans la base de données, on les récupère dans une variable $id. Voici sa fonction associée.

```php
public function getIdSujet(){
 $sql="select distinct sujet.id_sujet from fasteval.sujet";
 $res=PdoFastEval::$monPdo->query($sql);
 $laLigne=$res->fetchAll();
 return $laLigne;
}
```

Toutes les valeurs sont retournées grâce au FetchAll. Ici, la valeur affichée sera le libellé et non l'id. L'id sert à les identifier dans la liste.

```php
<select name='idsujet'>
    <option value="-1">Choisissez votre sujet</option>
    <?php
    for ($i = 0 ; $i < sizeof($id) ; $i++) {
    $libellesujet=$pdo->getLibelleById($id[$i][0]);
        ?>
        $idsujet=$id[$i][0];
        <option value = '<?php print_r($id[$i][0]) ?>'  label='<?php echo $libellesujet ?>'> </option>
        <?php
    }
   ?>
</select>
```

Voici la fonction qui affiche les libellés :

```php
public function getLibelleById($id){
    $sql="select sujet.libelle as a from fasteval.sujet where '$id'=id_sujet ";
    $res=PdoFastEval::$monPdo->query($sql);
    $laLigne=$res->fetch();
    $valeur = $laLigne['a'];
    return $valeur;
}
```

Une autre liste déroulante est présente pour afficher les promotions.

Ce tableau récapitule les fonctions du Pdo qui ont été utilisées pour cette partie de l'application (codées dans include/PdoFastEval.php).

Nom de la fonction	Description
getIdSujet()	Récupération des id des sujets contenus dans le champ id_sujet de la table sujet.
getIdpromotion()	Récupération des id des promotions contenus dans le champ id_promotion de la table promotion.
getLibelleById($id)	Récupération des libellés des sujets contenus dans le champ id_sujet de la table sujet, en fonction d'un id.
getLibellePromotionById($idpromotion)	Récupération des libellés des promotions contenus dans le champ libelle de la table sujet, en fonction d'un id.

2.3.2.2. Le script de correction

Afin de réaliser la correction des copies, trois scripts python sont exécutés :

- Un premier script est exécuté afin de convertir chaque page du PDF contenant les copies des étudiants en images afin qu'elles puissent être traitées par les autres scripts.
- Le deuxième script permet de détecter le numéro étudiant ou le numéro d'anonymat en fonction du type de l'évaluation. Il prend en entrée le chemin de l'image de la copie dont on souhaite connaître le numéro d'étudiant ou d'anonymat.
- Le dernier script, quant à lui, permet de réaliser la correction des copies. Ce script a besoin de quelques informations afin de corriger et d'enregistrer les notes de chaque étudiant en base de données, mais ces informations sont fournies par le PHP :
  - L'image de la copie à corriger,
  - La correction du sujet,
  - Le barème.

La correction des copies s'effectue en différentes étapes :

- La première étape consiste en la détection de la grille de réponse.
- La deuxième étape est la transformation du sujet en niveaux de gris, permettent de détecter au mieux les zones grisées dans la grille.
- L'étape trois permet d'extraire l'ensemble des bulles de réponses de l'examen.
- L'étape quatre comporte le tri des cercles de réponses détectés.
- L'étape cinq permet de transformer la chaîne de caractère correspondant aux réponses en dictionnaire afin de réaliser la correction.
- Enfin, lors de l'étape 5, on détecte la ou les réponses grisées, on vérifie si celles-ci sont les réponses qu'il fallait grisées, on applique le barème et on dessine les contours des bonnes et mauvaises réponses sur la copie.

2.3.3. STATISTIQUES

Pour l'onglet Statistique, la vue fait le lien avec le controller PdoFastEval pour l'affichage des données.

2.3.3.1. Liste déroulante

Dans la Vue Statistique a été mise en place une liste déroulante afin de sélectionner les données d'une évaluation en particulier. Le paramètre est l'ID_EVALUATION de la table EVALUATION de la base FASEVAL.

La liste comportera autant de sujet qu'il y a dans la base. On récupère ces valeurs avec la variable ideval. Le formulaire a une méthode GET. Il récupère donc des données.

```
$idevaluation = isset($_GET['ideval']) ? $_GET['ideval']:false;
```

```
<form method="get"  style="text-align: center;">
```

Pour afficher toutes les évaluations dans la base de données, on les récupère dans une variable $ideval ; voici sa fonction associée.

```
public function getiD(){
    $sql="select distinct id_evaluation from fasteval.note";
    $res=PdoFastEval::$monPdo->query($sql);
    $laLigne=$res->fetchAll();
    return $laLigne;
}
```

Toutes les valeurs sont retournées grâce au FetchAll.

2.3.3.2. Les fonctions Select

Chaque fonction select a pour but d'extraire des données et de les lire dans l'onglet STATISTIQUES. Elle porte référence au champ ID-EVALUATION de la table EVALUATION.

```
public function getNoteHaute($evaluation){
```

Une jointure est donc réalisée dans chacune de ces dernières pour faire le lien avec ce champ :

```
$sql="select max(p.note) AS a from fasteval.note p where '$evaluation' = p.id_evaluation";
```

Chaque fonction a la même architecture : on commence par sélectionner les données, on fait le lien avec la base de données, on récupère la ligne avec un Fetch puis on l'affiche.

```
public function getNoteHaute($evaluation){
    $sql="select max(p.note) AS a from fasteval.note p where '$evaluation' = p.id_evaluation";
    $res=PdoFastEval::$monPdo->query($sql);
    $laLigne=$res->fetch();
    $valeur = $laLigne['a'];
    return $valeur;
}
```

Ce tableau récapitule les fonctions du Pdo qui ont été utilisées pour cette partie de l'application (codées dans include/PdoFastEval.php).

Nom de la fonction	Description
getNomSujet($evaluation)	Récupération du nom du sujet contenus dans table sujet, à partir d'un id d'évaluation.
getDate($evaluation)	Récupération de la date de l'évaluation contenue dans le champ date_evaluation de la table évaluation, à partir d'un id d'évaluation.
getType($evaluation)	Récupération du type d'évaluation (EX ou CC) contenu dans le champ type de la table évaluation, à partir d'un id d'évaluation.
getPromo($evaluation)	Récupération de la promotion contenue dans le champ nom_promotion de la table promotion, à partir d'un id d'évaluation.
getNombresdeNoteInferieureA ($evaluation,$note)	Récupération du nombre de notes inférieures, par rapport à une note de référence, contenu dans le champ note de la table note, à partir d'un id d'évaluation. Utilisation d'un count.

getNombresdeNoteSuperieurouegaleA ($evaluation,$note)	Récupération du nombre de notes supérieures ou égales, par rapport à une note de référence, contenu dans le champ note de la table note, à partir d'un id d'évaluation. Utilisation d'un count.
getNombresdeNoteEntre ($evaluation,$notebasse,$notehaute)	Récupération du nombre de notes, entre 2 notes de référence, contenu dans le champ note de la table note à partir d'un id d'évaluation. Utilisation d'un count. La note haute à une limite de 20. La note basse est inférieure ou égale.
getNombres($evaluation)	Récupération du nombre de note contenu dans le champ note de la table note, à partir d'un id d'évaluation. Utilisation d'un count.
getMoyenne($evaluation)	Récupération de la moyenne d'une évaluation, contenu dans le champ note de la table note, à partir d'un id d'évaluation. Utilisation d'un avg.
getMediane($evaluation)	Il n'existe pas de fonction médiane en SQL. 4 étapes : • Classer les notes par ordre croissant avec un select, • Récupération du nombre de note, • Identification de la nature de ce nombre : PAIR ou IMPAIR avec un if, • Détermination de la médiane.
getNoteHaute($evaluation)	Récupération de la plus grande note d'une évaluation contenu dans le champ note de la table note, à partir d'un id d'évaluation. Utilisation d'un MAX.
getNoteBasse($evaluation)	Récupération de la plus petite note d'une évaluation contenu dans le champ note de la table note, à partir d'un id d'évaluation. Utilisation d'un MIN.

Les résultats sont ensuite affichés dans l'onglet STATISTIQUES.

Récupération des valeurs dans une variable locale :

```
$note_haute=$pdo-> getNoteHaute($idevaluation);
```

Utilisation du mot clé PRINT-R OU ECHO pour l'affichage :

```
<br><label style="font-weight: bold;"> Note la plus haute :  <?php print_r($note_haute)  ?> </label><br>
```

2.3.3.3. Les Graphs

Deux graphes sont présents :

- Barchart pour la proportion de notes inférieures et supérieures à 10,
- ColumnChart pour la répartition des notes.

Les deux graphs ont été réalisées avec la librairie Google Chart et utilisent des fonctions de PDOFastEval. Leurs développements sont faits directement dans la view STATISTIQUES.

Graphe	Description
BARCHART	Récupération des données avec les fonctions : getNombresdeNoteInferieureA($evaluation,$note) et getNombresdeNoteInferieureEtSuperieurOuEgal A ($evaluation,$note). Dans les options : choix des couleurs, du titre, du nombre de parts.
COLUMNCHART	Récupération des données avec la fonction : getNombresdeNoteEntre($evaluation,$notebasse,$notehaute). Dans les options : choix des couleurs, du titre, du nombre de colonnes.

Voici la balise en charge de l'affichage du BARCHART :

```
<div id="myPieChart" style='height:200px;'></div>
```

Voici la balise en charge de l'affichage du COLUMNCHART :

```
<div id="columnchart_values" style="width: 1000px; height: 350px;"></div>
```

2.3.3.4. Responsivité

Afin que cette fenêtre soit responsive, nous avons utilisé une bibliothèque bootstrap (notamment pour les graphes).

2.3.4. PRESENTATION

Cette fenêtre ne comporte que du HTML. Dans le texte, certains mots sont cliquables et renvoient vers une autre page. Le mot Paramètres renvoie vers la page paramètre de FastEval. La balise suivante en est responsable :

```
<a href='/FastEval/View/parametre'>Paramètres</a>
```

Un bouton cliquable a également été mis en place Il renvoie vers la fenêtre correction grâce au onclick qui comporte un href vers la page correction.

```
<input class="bouton_Correction" type=button onclick=window.location.href="/FastEval/View/correction";
VALUE ="Je veux corriger mes sujets"  />
```

2.3.5. RESULTATS

2.3.5.1. Liste déroulante

Dans la Vue Résultat a été mise en place une liste déroulante afin de sélectionner les données d'une évaluation en particulier. Le paramètre est l'ID_EVALUATION de la table EVALUATION. La liste comportera autant de sujets qu'il y en a dans la base. On récupère ces valeurs avec la variable ideval. Le formulaire a une méthode GET. Il récupère donc des données.

```
$idevaluation = isset($_GET['ideval']) ? $_GET['ideval']:false;
```

```
<form method="get"  style="text-align: center;">
```

Pour afficher toutes les évaluations dans la base de données, on les récupère dans une variable $ideval. Voici sa fonction associée.

```
public function getiD(){
    $sql="select distinct id_evaluation from fasteval.note";
    $res=PdoFastEval::$monPdo->query($sql);
    $laLigne=$res->fetchAll();
    return $laLigne;
}
```

Toutes les valeurs sont retournées grâce au FetchAll.

2.3.5.2. Les tableaux d'affichage de note

Il y a deux cas possibles ; soit c'est une évaluation dans le cadre d'un examen, donc le tableau affichera le numéro d'anonymat et la note, soit c'est du contrôle continu et le tableau retourna le numéro étudiant, le nom, prénom et note de l'étudiant.

```
<table>

<tr>
   <th> Numero Anonymat </th>
   <th> Note </th>
</tr>
```

```
<table >

<tr>
   <th> Numero Etudiant</th>
   <th> Nom </th>
   <th> Prénom</th>
   <th> Note </th>
</tr>
```

Les tableaux ont été créés aves des lignes et des colonnes. Les numéros des colonnes restent fixes car ils sont déjà déterminés. Par exemple, la colonne 1 recense les numéros d'anonymat ou d'étudiants. Des variables ont été déclarées pour l'affichage de données (Voir partie 2.3.5.3).

```
<p><?php $tabl= array();

for ($l = 0 ; $l < sizeof($numeroetudiant) ; $l++) {
    $tabl[$l][1]=$numeroetudiant[$l][0];
    $tabl[$l][2]=$nom[$l][0];
    $tabl[$l][3]=$prenom[$l][0];
    $tabl[$l][4]=$note[$l][0];
}

    $tab2= array();

for ($l = 0 ; $l < sizeof($numeroanonyme) ; $l++) {
    $tab2[$l][1]=$numeroanonyme[$l][0];
    $tab2[$l][2]=$note[$l][0];
}
```

Voici le code de l'affichage du tableau des notes pour les évaluations de contrôle continu. Il y a 4 colonnes et autant de lignes que d'étudiants présents à cette évaluation.

```
            <?php
                for ($i = 0; $i <  sizeof($numeroetudiant); $i++) {
            ?>

    <tr>
       <?php
            for ($j = 1; $j <= 4; $j++){
       ?>
       <td> <?php echo $tabl[$i][$j]; ?> </td >

       <?php
            }
       ?>

    </tr>

       <?php
            }
       ?>
   </table>
```

2.3.5.3. Les fonctions Select

Chaque fonction est un select, et a pour but d'extraire des données et de les lire dans l'onglet Résultats. Elles sont développées dans PdoFastEval.

Toutes ces fonctions retournent plusieurs valeurs grâce à un FETCHALL ; Voici un exemple de sa structure.

```php
public function getNoteEtudiantByIdEvaluation($evaluation){
    $sql="select  note.note from note where '$evaluation' = note.id_evaluation";
    $res=PdoFastEval::$monPdo->query($sql);
    $laLigne=$res->fetchAll();
    return $laLigne;
}
```

Ce tableau récapitule les fonctions du Pdo qui ont été utilisées pour cette partie de l'application (codées dans include/PdoFastEval.php).

Nom de la fonction	Description
getNumeroAnonymatByIdEvaluation ($evaluation)	Récupération des numéros d'anonymat contenus dans le champ num_anonymat de la table etudiant_anonyme, à partir d'un id d'évaluation.
getNumeroEtudiantByIdEvaluation ($evaluation)	Récupération des numéros étudiant contenus dans le champ num_etudiant de la table etudiant_nominatif, à partir d'un id d'évaluation.
getNomEtudiantByIdEvaluation ($evaluation)	Récupération des noms d'étudiants contenus dans le champ nom_etudiant de la table etudiant_nominatif, à partir d'un id d'évaluation
getPrenomEtudiantByIdEvaluation ($evaluation)	Récupération des prénoms d'étudiants contenus dans le champ prenom_etudiant de la table etudiant_nominatif à partir d'un id d'évaluation
getNoteEtudiantByIdEvaluation ($evaluation)	Récupération des notes d'étudiants contenu dans le champ note de la table note à partir d'un id d'évaluation.

2.3.6. ETUDIANTS

2.3.6.1. Les listes déroulantes.

Cette fenêtre comporte deux listes déroulantes. La première pour sélectionner un étudiant : anonyme ou nominatif.

```
<select name='Anonyme'>
    <option value="-1">Sélectionnez un type d'étudiants</option>
    <option value="1">Anonyme</option>
    <option value="0">Nominatif</option>
     </select><br><br>
```

Sa validation passe par le bouton Valider, qui renvoie soit vers la fenêtre étudiants nominatifs soit vers la fenêtre étudiants anonymes. Elles contiennent le nécessaire pour enregistrer un étudiant (voir partie sur les inputs).

L'action qui fait le changement de fenêtre se trouve dans le fichier formEtudiant, appelé par la balise <form>.

```
<form action="../model/formEtudiant.php" method="post">
<input class="bouton_valider" type="submit" name="form" value="Créer les étudiants via un formulaire" ><br><br>
```

La seconde liste déroulante permet de sélectionner une promotion. La liste comportera autant de promotions qu'il y en a dans la base. On récupère ces valeurs avec la variable identificationPromotion. Le formulaire a une méthode GET. Il récupère donc des données.

```
$identificationPromotion= isset($_GET['idpromo']) ? $_GET['idpromo']:false;
<form method="get"  style="text-align: center;">
```

Pour afficher toutes les évaluations dans la base de données, on les récupère dans une variable $id. Voici sa fonction associée.

```
public function getIdPromotion(){
    $sql="select distinct promotion.id_promotion from fasteval.promotion";
    $res=PdoFastEval::$monPdo->query($sql);
    $laLigne=$res->fetchAll();
    return $laLigne;
}
```

Toutes les valeurs sont retournées grâce au FetchAll. Ici, la valeur affichée sera le libellé et non l'id. L'id sert à les identifier dans la liste.

```
      <select name='idpromo'>
<option value="-1">Choisissez la promotion </option>
<?php
for ($i = 0 ; $i < sizeof($id) ; $i++) {
$libellePromotion=$pdo->getLibellePromotionById($id[$i][0]);
    ?>
    $idpromo=$id[$i][0];
    <option value = '<?php print_r($id[$i][0]) ?>' label='<?php echo $libellePromotion ?>'> </option>
    <?php
}
?>
```

Voici la fonction qui affiche les libellés :

```
public function getLibellePromotionById($idPromotion){
    $sql="select promotion.nom_promotion as a from fasteval.promotion where id_promotion='$idPromotion' ";
    $res=PdoFastEval::$monPdo->query($sql);
    $laLigne=$res->fetch();
    $valeur = $laLigne['a'];
    return $valeur;
}
```

2.3.6.2. Les inputs pour la création en base

Ici, il est possible de créer des promotions ou des étudiants. Pour cela, nous avons utilisés des balises <input> et un bouton valider qui fait le lien avec une autre fenêtre php (nécessaire pour faire des insert).

Nous avons utilisé la méthode POST car c'est une création de valeur. A chaque début de formulaire, dans la balise est inscrit l'action associé ;

```
<form action="../model/EnregistrePromotion.php" method="post">
```

Voici ce que contient le model PHP :

```
require_once ("../include/PdoFastEval.php");

$pdo = PdoFastEval::getPdoFastEval();

if(!empty($_POST['Valider'])) {
    $nom_promotion = $_POST["nom_promotion"];
    $date_promotion = $_POST["date_promotion"];
    $pdo-> insertPromotion($nom_promotion, $date_promotion);
}

include("../View/etudiants.php");
```

Les $_POST récupèrent les données inscrites dans les inputs. Quand on clique sur le bouton Valider, l'insertion se réalise. C'est le même principe pour insérer des étudiants nominatifs ou anonymes.

2.3.6.3. Les fonctions Insert

Nom de la fonction	Description
insertPromotion ($nom_promotion, $date_promotion)	Une promotion est ajoutée dans la table Promotion, avec un nom et une date.
insertEtudiantNominatif ($id_promotion, $numEtudiant, $nom, $prenom)	Un étudiant est ajouté dans la table Etudiat_nominatif, avec un id_prmotion, un nom, un prénom et un numéro d'étudiant.
insertEtudiantAnonyme ($id_promotion, $numAnonymat)	Un étudiant est ajouté dans la table Etudiat_anonyme, avec un id_promotion et un numéro d'anonymat.

2.3.7. PARAMETRES

L'onglet paramètre réunit des inputs et un bouton valider qui met à jour le barème en base. La méthode POST est utilisée.

La mise à jour se fait dans un fichier à part : EnregistreParam. Il est appelé par la balise <form>.

```
<form method="post" action="../model/enregistreParam.php" style="text-align: center;" >
```

```
<?php
$bonne_rep = $_POST['bonne_reponse'];
$mauvaise_rep = $_POST['mauvaise_reponse'];
$absence_rep = $_POST['absence_reponse'];
$non_reco_rep = $_POST['non_reconnaissance_reponse'];
$stockage_sujet = $_POST['stockage_sujet'];
$stockage_copies = $_POST['stockage_copies_corrige'];
require_once ("../include/PdoFastEval.php");

$pdo = PdoFastEval::getPdoFastEval();
$pdo->majParametre('bonne_reponse',$bonne_rep);
$pdo->majParametre('mauvaise_reponse',$mauvaise_rep);
$pdo->majParametre('absence_reponse',$absence_rep);
$pdo->majParametre('non_reconnaissance_reponse',$non_reco_rep);
$pdo->majParametre('stockage_sujet',$stockage_sujet);
$pdo->majParametre('stockage_copies_corrige',$stockage_copies);
include("../View/parametre.php");
?>
```

Voici le détail des fonctions ; elles sont codées dans le fichier PdoFastEval.php.

Nom de la fonction	Description
getBareme($LeLibelle)	Récupération des barèmes contenus dans le champ value de la table paramètre, à partir d'un libelle.
majParametre($leLibelle, $laValeur)	Un barème est mis à jour dans la table Paramètre en fonction du libellé (ex : bonne réponse).

3. LES JEUX DE TESTS

3.1. PLAN DE TEST « INDEX.PHP »

n°	Jeux d'essai	Résultats attendus	Résultats
0	Tentative d'accès à la page presentation.php,	Affichage de la page présentation	√
1	Tentative d'accès à la page correction.php,	Affichage de la page correction	√
2	Tentative d'accès à la page resultats.php,	Affichage de la page Résultat	√
3	Tentative d'accès à la page etudiant.php,	Affichage de la page Etudiants	√
4	Tentative d'accès à la page sujet.php	Affichage de la page Sujet	√
5	Tentative d'accès à la page statistiques.php	Affichage de la page statistiques	√
6	Tentative d'accès à la page parametres.php	Affichage de la page Paramètres	√
7	Tentative d'accès à la page Plan site	Affichage de la page plan site	√

3.2. PLAN DE TEST «PRESENTATION.PHP»

n°	Jeux d'essai	Résultats attendus	Résultats
0	Bouton statistiques	Affichage de la page statistiques	√
1	Bouton paramètres	Affichage de la page paramètres	√
2	Bouton “ je veux corriger mes copies “	Affichage de la page correction	√
3	Texte explicatif visible ?	Texte expliquant l’objectif du site	√

3.3. PLAN DE TEST «CORRECTION.PHP»

n°	Jeux d'essai	Résultats attendus	Résultats
0	Bouton de sélection d’un sujet parmi ceux existant sur le site	Liste déroulante avec les sujets existants / choix d’un sujet	√
1	Importation des copies	Importer un sujet à partir de l’ordinateur	√
2	Bouton choisir une promotion	Afficher les promotions existantes et en choisir une	√
3	Bouton Valider	Valider les informations données et avoir la correction	√

3.4. PLAN DE TEST «RESULTAT.PHP»

n°	Jeux d'essai	Résultats attendus	Résultats
0	Sélectionner une évaluation déjà effectués	Avoir accès aux évaluations déjà effectuées	√

3.5. PLAN DE TEST «ETUDIANTS.PHP»

n°	Jeux d'essai	Résultats attendus	Résultats
0	Sélectionner le type d'étudiants	Affichage d'une liste déroulante avec comme choix : nominatif / anonyme	√
1	Bouton " créer les étudiants via un formulaire"	Accès à la page création formulaire	√
2	Création des étudiants grace à un fichier XLS Bouton Parcourir	Récupérer un fichier depuis l'ordinateur	√
3	Création d'une promotion Bouton valider	Renseigner le nom de la promotion et l'année et valider	√

3.6. PLAN DE TEST «SUJET.PHP»

n°	Jeux d'essai	Résultats attendus	Résultats
0	Ajout d'une évaluation / date	Avoir accès à un calendrier et choisir une date	√
1	Choisir le type du test (CC / EX)	Avoir une liste déroulante et choisir le type de l'examen	√
2	Sélection de promotions	Avoir une liste déroulante et choisir une promotion parmi celles existantes	√
3	Ecrire le nombre d'étudiants	Renseigner le champs (informations obligatoires)	√
4	Bouton Valider	Valider les informations renseignées	√
5	Ajouter un sujet / renseigner un nom	Donner un libellé au sujet	√
6	Sélectionner une évaluation	Liste déroulante avec les évaluations déjà créées	√
7	Téléchargement du fichier text	Parcourir le fichier texte et le télécharger	√
8	Valider	Valider et générer le sujet	√
9	Choisir un sujet	Liste déroulante avec les sujets déjà créés / choix du sujet	√

3.7. PLAN DE TEST «STATISTIQUES.PHP»

n°	Jeux d'essai	Résultats attendus	Résultats
0	Sélectionner un sujet déjà corrigé	Liste déroulante pour sélectionner un sujet corrigé parmi ceux existants Avoir toutes les statistiques des notes	√

3.8. PLAN DE TEST «STATISTIQUES.PHP»

n°	Jeux d'essai	Résultats attendus	Résultats
0	Remplir les champs barème (bonne/mauvaise/ absence/non reconnaissance de la réponse)	Avoir un barème pré rempli applicable dans la correction des sujets.	√
1	Stockage des copies et des sujets	Choisir le chemin ou enregistrer les copies déjà corrigées et les sujets générés	√

3.9. PLAN DE TEST «PLANSITE.PHP»

n°	Jeux d'essai	Résultats attendus	Résultats
0	Tentative d'accès à la page presentation.php,	Affichage de la page presentation	√
1	Tentative d'accès à la page correction.php,	Affichage de la page correction	√
2	Tentative d'accès à la page resultats.php,	Affichage de la page Résultat	√
3	Tentative d'accès à la page etudiant.php,	Affichage de la page Etudiants	√
4	Tentative d'accès à la page sujet.php	Affichage de la page Sujet	√
5	Tentative d'accès à la page statistiques.php	Affichage de la page statistiques	√
6	Tentative d'accès à la page parametres.php	Affichage de la page Paramètres	√