

Projet le club des Amis

I \ Présentation du contexte

Le laboratoire Galaxy Swiss Bourdin (GSB) possède un Comité d'Entreprise (CE) très important et très actif. Parmi ses nombreuses ramifications, il existe le club des AMIS (Aide Mutuelle Intérêt et Service). Ce club a la vocation d'entraide et d'amitié parmi les employés retraités et actuels. Les membres s'appellent donc des AMIS. Le but du club est d'organiser des actions afin de venir en aide à des personnes dans le besoin. Cela peut être des employés actuels ou des retraités, mais également le club se mobilise pour financer des œuvres diverses : recherche contre le SIDA, financement de voyages de jeunes en difficulté (issus de parents employés à GSB ou non), soutien à des AMIS en difficulté...

Outils Utilisés: Git Hub, MYSQL, PHP

Objectifs :

Objectif secondaire :

- Modélisation de la base de données (MCD, MLR)
- T3.3 Inscription à un dîner

Développeurs

- Stéphane Benatout
- Quentin Gimié D'Arnaud

II \ Conte-rendu hebdomadaire

Semaine 1 :

Début du PPE, on a commencé l'assemblée plénière et la mise en place ainsi que l'initiation à GIT. Découverte de Source Tree pour Quentin mais malheureusement quelques problèmes de compatibilité avec GIT ont fait qu'on a été contraint d'utiliser Git-Bash.

Semaine 2 :

L'assemblée plénière se clôture par la modélisation et la mise en place de la base de données du PPE. Puis la base de données nommée ppemamis a été créée.

Semaine 3 :

Nous avons commencé le PPE, avec notre première tâche qui a pour but de faire la modification et la suppression des repas. Nous avons créé la requête de modification et de suppression. Puis, nous avons vérifié les requêtes avec PhphMyAdmin. Enfin, nous avons rectifié quelques petites erreurs sur cette tâche.

Semaine 4 :

Dès le début, nous avons pris comme initiative de séparer les tâches demandées en deux parties. Stéphane a continué sur la première partie et Quentin a, de son côté, commencé à s'occuper de la deuxième tâche voir même à côté la troisième et dernière tâche.

Semaine 5 :

On a travaillé sur nos tâches respectives, mais nous avons rencontré de sérieux soucis de programmation. Stéphane a eu des problèmes avec le contrôleur et Quentin avec sa vue.

Semaine 6 :

Quentin a rencontré un souci de lecture de sa vue due à une erreur de code de chemin. Il a réussi à corriger les erreurs, grâce à une intense réflexion venant de Quentin, des professeurs, et de Stéphane. Pendant ce temps-là, Stéphane a essayé de terminer sa partie tout en comprenant ses erreurs et les erreurs de son coéquipier

Semaine 7 :

Pour la dernière Séance avant les vacances, Quentin et Stéphane ont échangé leurs tâches dans le but de corriger les erreurs, et essayer de terminer le PPE.

Vacances de Noël :

Durant ces vacances, Quentin a avancé sur son programme, malgré un problème qui lui a pris la moitié d'une semaine à résoudre. Stéphane a aussi travaillé le PPE tout en cherchant un très activement un stage.

Durant le stage :

Même si le délai est écoulé, Quentin a pris l'initiative de terminer et de pouvoir réussir à parvenir de rectifier ses erreurs de programmation après les journées de stage.

III\ Documentation Technique

Durant les deux premières heures de PPE, l'équipe de développeur s'est réunis et on a réalisé le Modèle Conceptuel de Données (MCD) et le Model Logique Relationnel (MLR) sur Windesign.

Détail des taches de la mission :

Pour information, nous partons du menu d'accueil, puis nous allons dans la rubrique "Gestion des dîners".



T3.2 Modification/Suppression d'un dîner

Pour cette tâche, on fait deux actions.

1. La modification on crée d'abord la fonction comportant la requête de modification avec des paramètres.

```
/*
 * Modification des données d'un repas
 */
* @param $idRepas
* @param $heure
* @param $date
* @param $places
* @param $lieu
*/
public function modifierRepas($idRepas, $heure, $date, $prix, $places, $lieu){
    $req = "UPDATE `repas` SET HEUREREPAS = '$heure', DATEREPAS = '$date', PRIXREPAS = '$prix', NBRPLACESREPAS = '$places', LIEUREPAS = '$lieu' WHERE
    IDREPAS = '$idRepas'";
    $rs = $pdo->query($req);
    return $rs;
}
```

Puis on va créer le contrôleur de cette tâche.

```
case 'modifierDiner':{
    if(isset($_POST['dateDiner']) && ($_POST['heureDiner']) && ($_POST['prixDiner']) && ($_POST['
    nbplaceDiner']) && ($_POST['lieuDiner'])){

        $idRepas=$_POST['idRepas'];
        $date=$_POST['dateDiner'];
        $heure=$_POST['heureDiner'];
        $prix=$_POST['prixDiner'];
        $places=$_POST['nbplaceDiner'];
        $lieu=$_POST['lieuDiner'];
        $pdo->modifierDiner($idRepas,$heure,$date,$prix,$places,$lieu);
    }
    break;
}
```

Cette tâche va permettre de rappeler les variables du formulaire et de vérifier si elles sont définies et si elles ont une valeur autre que Null. Ensuite, on affecte les valeurs aux variables qui servent de paramètres à la requête. Enfin on rappelle la vue du formulaire d'inscription à un dîner.

1. La Suppression

On va créer la fonction comportant la requête de suppression d'un repas donné.

```

/*
 *Suppression des données d'un repas
 */
@param $idRepas
*supprime le diner par l'identifiant
*/
public function supprimerRepas($idRepas){
    $req = " DELETE * FROM `repas` WHERE NUMREPAS='$idRepas'";
    $rs =PdoGsb::$monPdo->query($req);
    return $rs;
}

```

Puis, on va créer le contrôleur de suppression.

```

case 'supprimerDiner':{
    $idRepas= $_REQUEST['idRepas'];
    $pdo->supprimerRepas($idRepas);
    break;
}

```

Le contrôleur vérifie l'identifiant du repas pour ensuite le supprimer grâce à la fonction de suppression. Nous pouvons même rajouter un message de confirmation dans le but de signaler à l'utilisateur que la suppression a été bien exécutée.

T3.3 Inscription d'un ami à un dîner

1. La vue

Cette vue va permettre de pouvoir inscrire un amis à un repas.

```

<?php
    $listediner=$pdo->getRepas();

    $repas=array();

    $cmbdiner="cmbdiner";

    $cmbamis="cmbamis";
    $cnbrepas="NombreRepas";

?>

```

Dans la première partie, on déclare trois variables qui reçoivent les données des deux formulaires et du nombre de places par repas. La liste des repas s'affichera dans un tableau grâce à la fonction getRepas().

```

/*
 *
 *retourne tous les repas
 */
public function getRepas(){
    $req="SELECT * FROM REPAS";
    $rs =PdoGsb::$monPdo->query($req);
    $ligne=array();
    if($rs == true){
        $ligne = $rs->fetchAll();
    }
    return $ligne;
}

```

Le formulaire va utiliser une méthode POST pour passer les données à travers l'URL de manière non visible. Le formulaire affiche la valeur choisi dans la liste déroulante. Le selected permet de stocker la ou les valeurs sélectionnées.

Le fetch permet de "chercher" dans la table toutes les valeurs qui correspondent, afin de placer ces données dans une liste roulante.

```
if (isset($_POST['cmbdiner'])) {  
  
    $repas=$pdo->getRepasById($_POST['cmbdiner']);  
  
    $cmbdiner=$_POST['cmbdiner'];  
    ?>
```

On va tester les variables avec la condition isset() et ainsi permettre donner en paramètre de la fonction getRepasId() qui récupere tous les repas par les identifications.

```
/*récupération des repas par l'identifiant  
=  
*@paramètre $idRepas  
=  
*/  
public function getRepasById($idRepas){  
    $req= "SELECT * FROM REPAS WHERE NUMREPAS=$idRepas";  
    $rs = PdoGsb::$monPdo->query($req);  
    $rec=$rs->fetch();  
    return $rec;  
}
```

Ensuite, on attribut, tout en testant la condition, la valeur du premier foreach à la variable \$cmbdiner.


```

<script type="text/javascript">
    var cmbdiner = document.getElementById('cmbdiner');

    if (cmbdiner == null){
        alert('Attention champs vide');
        cmbdiner.addClass('error'); //les bordures de la zone de textes passent en rouge
        erreur = true;
    }

    var Nbrdiner= document.getElementById('Nbrdiner');

    if(Nbrdiner ==null){
        alert('Attention champs vide');
        Nbrdiner.addClass('error'); //les bordures de la zone de textes passent en rouge
        erreur = true;
    }

}
</script>

```

Les tests en javascript ont pour but de vérifier la condition d'un champ Null ou vide et de les marquer comme erreur.

1. Le contrôleur

```

    case 'inscription':{
        if (isset($_POST['cmbdiner']) && isset($_POST['']) &&
            isset($_POST['NombreRepas'])) {
            $idRepas = $_POST['cmbdiner'];
            $idAmis = $_POST['cmbamis'];
            $cnbrepas = $_POST['NombreRepas'];

            $pdo->insertAmisRepas($idRepas,$idAmis,$
                cnbrepas);
            include("Vue/v_inscriptionDinerAmis.php");
        }
    }

```

Nous refaisons une condition pour vérifier les valeurs récupérées de la vue et nous les affectons aux paramètres de la fonction insertAmisRepas(), ce qui permet d'ajouter un amis au repas.

```

/*
 *Insérer une nouvelle invitation d'un amis à un repas
 *@paramètre
 *
 */
public function insertAmisRepas($idRepas,$idAmis,$
    cnbrepas){
    $req="INSERT INTO ($cmbdiner,$cmbamis,$cnbrepas)VALUES
        (?, ?, ?)";
    PdoGsb::$monPdo->exec($req);
}

```

T3.4 Edition de la liste des participants à un dîner

- La vue

La vue v_imprimerDinerPDF.php permet d'imprimer une liste comportant les données d'inscription d'un ami à un repas.

```

<?php

$content='

<page>
    <h1 style="align-text:center">Edition de la liste
    des Amis invité à un Diner</h1>

    <table border=1>

        <tr>
            <td >DATEREPAS</td>
            <td>HEUREREPAS</td>
            <td>LIEUREPAS</td>
            <td>PRIXREPAS</td>
            <td>NombreRepas</td>
            <td>NomAmis</td>
            <td>PrénomAmis</td>

        </tr>';

$listeparticipant = $pdo->getParticipantRepas($idRepas);

while ($ligne=$listeparticipant->fetch(PDO::
    FETCH_OBJECT)) {
    $content.='<tr>';
    $content.='<td>'. $ligne->date_repas. '</td>';
    $content.='<td>'. $ligne->heure_repas. '</td>';
    $content.='<td>'. $ligne->lieu_repas. '</td>';
    $content.='<td>'. $ligne->nombre_repas. '</td>'
        ;
    $content.='<td>'. $ligne->nom_amis. '</td>';
    $content.='<td>'. $ligne->prénom_amis. '</td>';
    $content.='</tr>';
}

$content.="</table></page>";

```

Nous créons la variable content qui va avoir pour valeur toute la syntaxe de la page (présentation, titre, tableaux, valeur du tableaux, etc...). La deuxième variable nommé \$listeparticipant contient toutes les participations grâce à la fonction getParticipantRepas().

```

case 'edition':{
    $repas=1;
    $listediner=$pdo->getRepas();
    echo("erreur");
    $listeparticipant = $pdo->getParticipantRepas($
        listediner);
    include("Vue/v_imprimerDinerPDF.php");
    break;
}

```

Permet de sélectionner en faisant une jointure entre toute les participations à un repas avec les amis qui y participent. Après on fait une boucle while permettant de sélectionner les convives désirés à un repas précis.

- Le premier contrôleur d'édition:

```

case 'edition':{
    $repas=1;
    $listediner=$pdo->getRepas();
    echo("erreur");
    $listeparticipant =$pdo->getParticipantRepas($
        listediner);
    include("Vue/v_imprimerDinerPDF.php");
    break;
}

```

- Le deuxième contrôleur : est un test pour afficher l'impression par lien

```

case 'edition2':{
    echo "<script>document.location.href=\"".$URL."
        Vue/v_imprimerDinerPDF.php\";</script>";
    break;
}

```

IV\ Documentation Utilisateur

Cette partie montre la partie utilisateur de la mission 9. Il faut savoir que chaque partie est reliée dans le menu et dans la rubrique Gestion des dîners. Puis allez directement à l'endroit qui porte le même nom que celle des tâches.

T3.3 Inscription d'un ami à un diner

Le test de la vue :

La vue *v_inscriptionDinerAmis.php* permet d'inscrire un ami à un diner

Partie 1 : Sélection d'un repas

On va choisir un dîner proposer dans la liste déroulant et ainsi afficher les données correspondantes.

Inscription d'un ou de plusieurs amis aux dîners

Choisir un repas

DATEREPAS	HEUREREPAS	PRIXREPAS	NombreRepas	LIEUREPAS
2017-11-13	00:12:45	450	6	Maxime

Fatal error: Call to a member function fetch() on a non-object in C:\Program Files (x86)\EasyPHP-DevServer-14.1VC9\data\localweb\app33w2017\Include\class.pdo.php on line 258

NomAmis PrenomAmis

Grâce à l'auto complétion on va pouvoir récupérer un ami.

Inscription d'un ou de plusieurs amis aux dîners

13/11/2017 Maxime

Liste des amis :

Stephane Benatout

Nombre Places: 45

Valider

Puis on valide en cliquant sur le bouton.

La gestion des erreurs

Les tests en javascript informe l'utilisateur lors d'un champ vide



La zone de nombres de place est vide.

