

## DOSSIER DE PROJET

Nom de naissance	✕ Lesage
Nom d'usage	✕
Prénom	✕ Pierre
Adresse	✕ 14 Place de l'espérance 60400 Noyon

### Titre professionnel visé

Développeur-se logiciel – Niveau III

## **Sommaire**

<b>1. <u>Remerciements</u></b>	Page 3
<b>2. <u>Résumé du projet en anglais</u></b>	Page 4
<b>3. <u>Liste des compétences du référentiel</u></b>	Page 5
<b>4. <u>Cahier des charges ou expressions des besoins de l'application à développer</u></b>	Page 6
a)Gestion des utilisateurs	Page 6
b)Gestion du temps	Page 7
c)Gestion des mails et de l'historique	Page 8
d)Base documentaire	Page 9
<b>5. <u>Spécifications fonctionnelles</u></b>	Page 10
<b>6. <u>Spécifications techniques</u></b>	Page 11
<b>7. <u>Réalisations</u></b>	Page 14
<b>8. <u>Conclusions</u></b>	Page 20

## **1- Remerciements**

Je tiens à remercier la société Simplon.co qui a créé cette formule de formation de 6 mois et sans qui nous n'aurions pas été en formation.

Merci à Sandrine Oriol et Alexandre Denurra de Novei pour leurs soutiens pendant les 6 mois intensifs.

Merci également aux 2 formateurs, Johanne Badin et Mickael Noel pour nous avoir donné les ficelles pour réussir dans nos futurs métiers.

Les Autres apprenants avec lesquels j'ai collaboré pendant la formation sur les différents projets durant la formation.

Les membres de la dernière équipe avec qui j'ai collaboré pour réaliser le projet présenter lors du titre professionnel.

## **2- Résumé du projet en anglais**

The project I worked on as part of my professional title « web developer » is a web application concerning candidates who will become learners, be registered on a waiting list or be refused. It also allows the teaching team to deal with the groups, presence time, working plannings, visits. The objective of my team are :

- the creation of different user groups, able or enable to use the connection,
- get information on candidates according to their status,
- deal with the users : identification, password, group they belong to, and access to different pages,
- during the selection, the teaching team will be able to vote for the candidates they want to get selected. The first vote is based on the application form, the second one takes place after the selection tests to choose the future learners.

For this project, we used HTML, CSS, Javascript and PHP languages and frameworks : Bootstrap, JQuery and Angular JS. The database we used is MySQL.

### **3- Liste des compétences du référentiel**

Pour l'activité les compétences du référentiel utilisé dans le projet présenté pour titre professionnel « Développeur-se Logiciel »:

- Développer une application simple de mobilité numérique a été utiliser car notre projet est responsive et peut s'afficher correctement sur tout les supports (téléphone, tablette, écran d'ordinateur) ;
- Développer des composants d'accès aux données, car dans le projet, pour accéder à l'intranet il faut rentrer ses identifiants et mots de passe, et pour certaines informations il faut faire partie des groupes autorisés.
- Développer des pages web en lien avec une base de donnée, notre travail a été effectuer avec la base de donnée créer. Ces données sont parfois afficher, modifier, supprimer dans le projet présenter.

## **4- Cahier des charges ou expressions des besoins de l'application à développer**

Le projet de d'intranet qui est présenter pour le titre professionnel a été partager en 5 parties dans les différents groupes de la formation :

- Gestion des utilisateurs ;
- Gestion du temps ;
- Gestion des mails et de l'historique ;
- Base documentaire ;
- Importation ;

### **a) Gestion des utilisateurs**

L'objectif de cette partie est la gestion de chaque candidature pour la formation ainsi que les différentes informations liées aux promotions et aux équipes pédagogiques.

Les candidats qui seront enregistrées dans la base de donnée, pourront changer de statut (candidats, accepter en promo, mis sur liste d'attente, ou refusé).

Pour les candidats, il sera possible de modifier différentes partie du profil de l'utilisateur (photo, profil github...).

Dans cette partie, il sera gérer l'authentification des personnes qui souhaitent accéder à l'intranet grâce à un identifiant et un mot de passe.

Les accès seront également gérer selon le groupe auxquels ils appartiendront car toutes les personnes ne pourront pas se connecter et/ou accéder à toutes les pages du site qu'ils soient :

- administrateur ;
- équipe pédagogique ;
- simplon ;
- candidats ;
- apprenants.

L'équipe pédagogique et les personnes faisant partie de Simplon pourront, à partir de cette partie, de visualiser les candidats et de les filtrer selon les résultats.

Pour l'équipe pédagogique il sera également possible de modifier les informations concernant les utilisateurs, ce qui permettra de changer la photo qui a été pris lors de l'entretien.

Au moment de l'inscription et lors des entretiens, le jury pourra voter s'il le candidat est accepter, refuser. Ce qui permettra de changer le statut de ces derniers.

b) Gestion du temps

L'objectif de cette partie du projet est de gérer le temps de présence, les absences et les retards des apprenants dans la formation.

Chaque apprenant, pendant la formation pointera sa présence chaque demi journée durant la formation.

L'équipe pédagogique pourra renseigner les différentes absences des apprenants et renseigner en leur donnant le statut de justifiée ou non.

En cas de retard, les apprenants pourront faire des commentaires sur les raisons du retard.

c) Gestion des mails et de l'historique

L'objectif de cette partie est de pouvoir envoyer des mails automatiquement en fonction du changement de statut ou tout autre action possible. Il est également possible de créer ces mails et de les envoyer automatiquement.

Il est possible dans l'application, pour l'équipe pédagogique de créer des mails type, de les visualiser, de les modifier et de les enregistrer dans la base.

Il y a aussi un historique qui permet de voir les différentes actions liées à un utilisateur et il peut être également ajouter un commentaire pour mieux comprendre les traces laissées automatiquement.



#### d) Base documentaire

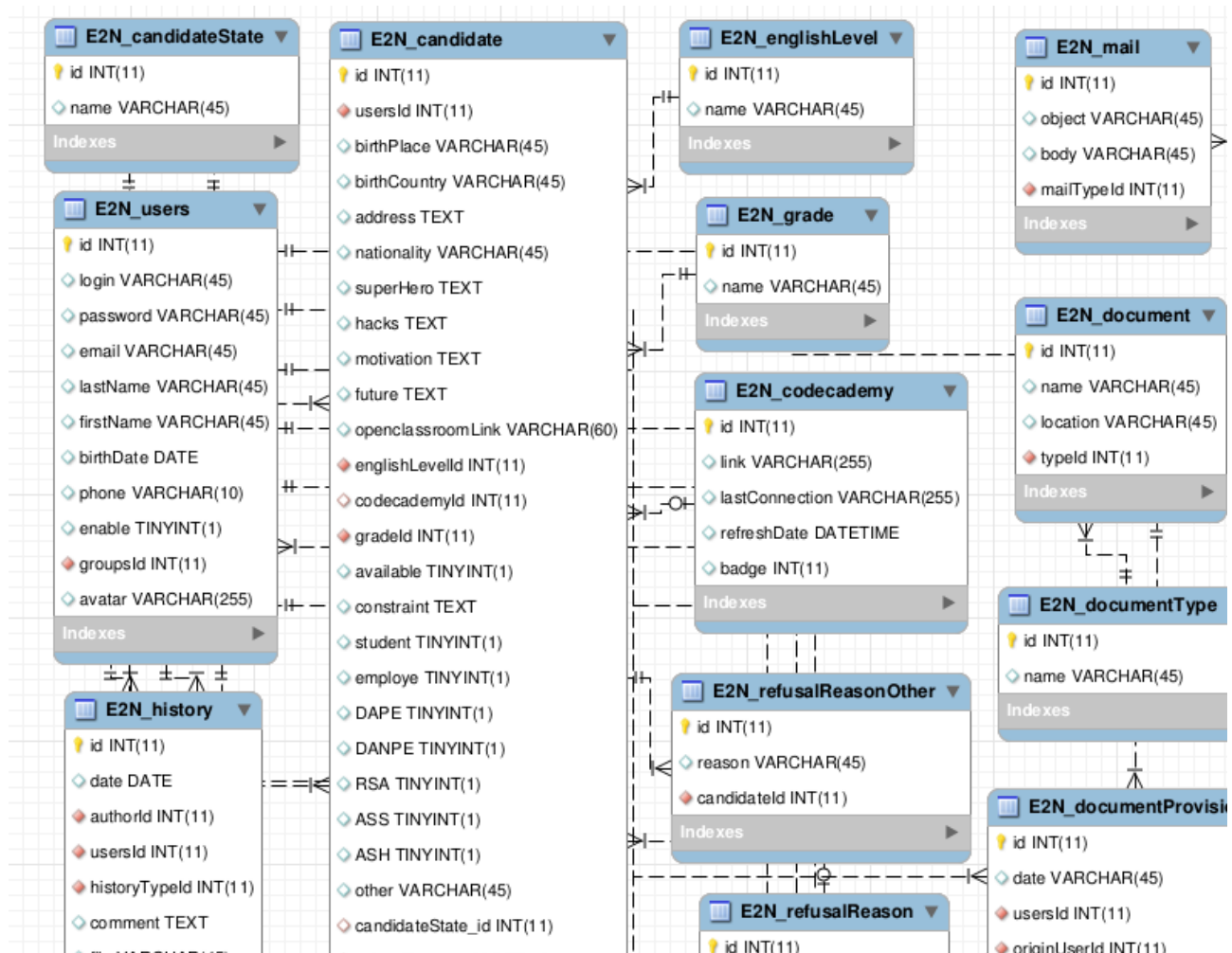
Pendant la formation, il est possible de transmettre de mettre à disposition des documents aux utilisateurs de manière individuel ou collective.

L'équipe pédagogique pourra d'ajouter, supprimer ou modifier un document.

Elle pourra également mettre ses documents à disposition à des équipes et/ou à des utilisateurs qui seront averti par mail dès qu'un document sera à leur disposition ou qu'une mise à jour a été faites.

## 5- Spécifications fonctionnelles

Pour créer les différentes fonctions nous avons travaillé diagramme de merise pour trouver les liens entre les tables et les réunir pour faire fonctionner les différentes requêtes.



## 6- Spécifications techniques

Pour la réalisation du projet nous avons utilisé comme langage HTML/CSS, javascript et PHP.

Nous avons pu utiliser des framework bootstrap, JQuery et AngularJs.

Pour la gestion de la base de donnée nous avons utilisé MySQL et pour le schéma des tables nous avons utilisé comme outil workbench.

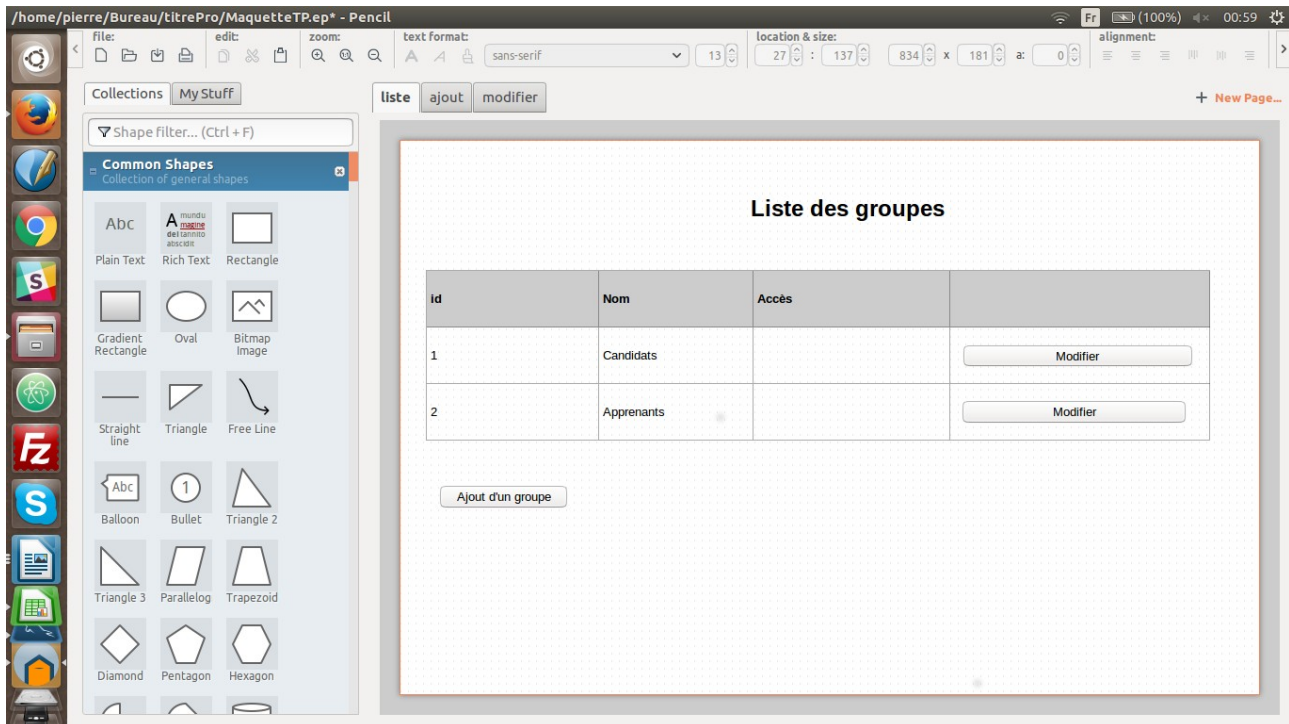
Pour l'organisation à l'intérieur des groupes et la gestion des taches à effectuer, nous avons pu utiliser trello.



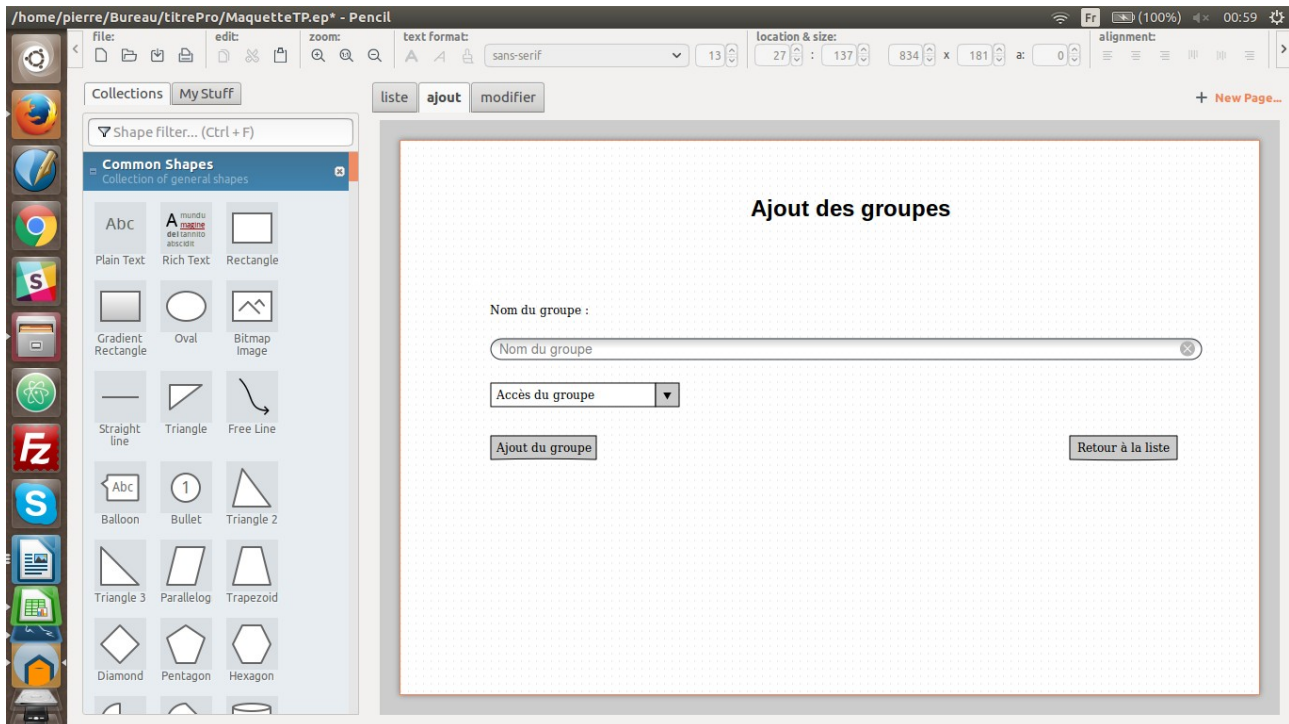
Pour avoir une vision des vues que nous voulions nos parties nous avons également pu utiliser Pencil.

Ci-dessous, voici ce qui a été imagine à la base pour la liste des groupes.

Et comment était imaginer le formulaire de création des groupes qui ressemble également au formulaire pour modifier les groupes.



Et comment était imaginer le formulaire de création des groupes qui ressemble également au formulaire pour modifier les groupes.



Pour redimensionner les images, photos nous avons pu utiliser GIMP.  
Pour créer notre code nous avons utilisé netbeans comme IDE.

## 7- Réalisations

Notre groupe de 4 personnes, était chargé la partie gestion des utilisateurs. Les taches étaient réparties en 4 parties :

- la gestion des groupes ;
- la gestion des utilisateurs ;
- la gestion des candidats ;
- la gestion des votes.

J'étais chargés de la partie groupe. Elle devait :

- afficher les différents groupes qui existent dans la base de donnée ;
- avoir un formulaire de création des groupes ;
- pouvoir modifier le nom des groupes ou le statut dans un formulaire qui est déjà rempli par les informations selon le groupe sélectionner ;
- pouvoir activer et désactiver les groupes qui s'affichent avec une croix verte ou une croix rouge selon son statut.

Organisation :

Nous avons organisé notre code selon le modèle MVC (Modèle Vue Controlleur) pour que ça soit plus facile à organiser notre code et plus clair.

Pour notre partie nous avons d'abord créer dans le Modèle un fichier database pour pouvoir se connecter à la base de donnée.

```
<?php

class database {

    public function __construct() {

    }

    public function connectDb() {
        try {
            return new PDO('mysql:host=192.168.1.129;dbname=tp_user;charset=utf8', 'usr_user', 'usr_user', array(PDO::ATTR_ERRMODE =>
PDO::ERRMODE_EXCEPTION));
        } catch (Exception $e) {
            die('Erreur : ' . $e->getMessage());
        }
    }
}
```

Dans ce fichier nous avons créer une class database dans laquelle on a créer une fonction connectDb qui essaye de se connecter à la base de donnée tp\_user dans le cas contraire il affiche un message d'erreur.

Ce fichier doit se retrouver dans le haut du fichier de la vue avec les autres fichiers qui sont en lien avec la vue comme ci-dessous.

```
<?php
include_once $_SERVER['DOCUMENT_ROOT'] . 'model/database.php';
include_once $_SERVER['DOCUMENT_ROOT'] . 'model/group.php';
include_once $_SERVER['DOCUMENT_ROOT'] . 'controller/groupController.php';
include_once $_SERVER['DOCUMENT_ROOT'] . 'view/header.php';
?>
```

La partie gestion des utilisateurs a été découpé en plusieurs partie :

- Les groupes ;
- Les utilisateurs ;
- Les candidats.

### \* Le model

J'ai créer le fichier « model » qui s'appelle group.php dans lequel nous avons créer une class « usrGroup ».

```
class usrGroup {  
  
    public $id;  
    public $name = '';  
    public $enable = 0;  
    private $pdo;
```

Dans cette classe nous avons mis les attributs qui correspondent à ce qu'il y a dans table groupe dans la base de donnée, c'est à dire l'id, le name, enable.

Puis j'ai créer le constructeur :

```
public function __construct($pdo) {  
    $this->pdo = $pdo;  
}
```

Ensuite j'ai crée mes fonctions qui vont permettre de faire mes différentes vues.

- getGroupList

```
public function getGroupList() {  
    $req = $this->pdo->query('SELECT * FROM `E2N_groups`');  
    $request = $req->fetchAll(PDO::FETCH_ASSOC);  
    return $request;  
}
```

Cette fonction a été faite pour que s'affiche dans la vue la liste des groupes de la base de donnée dans la vue.

Nous faisons donc une requête ou on va chercher toute les données dans la table E2N\_groups. On utilise fetchAll car on souhaite tous les groupes dans le tableau et ensuite on lui demande de retourner le tableau avec les groupes.

- addGroup

```
public function addGroup() {  
    $queryResult = $this->pdo->prepare('INSERT INTO `E2N_groups`(`name`, `enable`) VALUES(:name, :enable)');  
    $queryResult->bindValue(':name', $this->name, PDO::PARAM_STR);  
    $queryResult->bindValue(':enable', $this->enable, PDO::PARAM_STR);  
    $queryResult->execute();  
    return $queryResult;  
}
```

Cette fonction permet, lorsque l'on crée un groupe dans le formulaire, ce dernier soit insérer dans la base de donnée. On indique également que la valeur des deux champs sont des String.

- changeGroup

```

public function changeGroup() {
    $queryResult = $this->pdo->prepare('UPDATE `E2N_groups` SET `name` = :name, `enable`=:enable WHERE id = :id');
    $queryResult->bindValue(':id', $this->id, PDO::PARAM_INT);
    $queryResult->bindValue(':name', $this->name, PDO::PARAM_STR);
    $queryResult->bindValue(':enable', $this->enable, PDO::PARAM_INT);
    $queryResult->execute();
    return $queryResult;
}

```

La fonction permet quand on sélectionne un groupe pour modifier le nom ou le statut, que dans la base de donnée, les modifications soient faites.

#### - selectGroup

```

public function selectGroup() {
    $queryResult = $this->pdo->prepare('SELECT `name`, `enable` FROM `E2N_groups` WHERE id = :id');
    $queryResult->bindValue(':id', $this->id, PDO::PARAM_INT);
    $queryResult->execute();
    $request = $queryResult->fetch(PDO::FETCH_ASSOC);
    if (is_array($request)) {
        $this->name = $request['name'];
        $this->enable = $request['enable'];
        return true;
    } else {
        return false;
    }
}

```

La fonction selectGroup est complémentaire à la fonction changeGroup car cette fonction va permettre quand on va sélectionner le group, il va récupérer les informations du name et du enable et dans le formulaire de modification, vont apparaître ces informations selon l'id qui sélectionner.

#### \* Le controleur

Il a été créer pour toute les parties un fichier controleur par vue.

#### - groupController

```

<?php
include_once $_SERVER['DOCUMENT_ROOT'] . 'model/database.php';
include_once $_SERVER['DOCUMENT_ROOT'] . 'model/group.php';
?>
<?php

$pdo = new database();
$connectPdo = $pdo->connectdb();
$usrGroup = new usrGroup($connectPdo);
//Appel de la méthode getGroupList de la classe groupe.
//La fonction retourne un tableau que l'on met dans $groupList
$groupList = $usrGroup->getGroupList($usrGroup);
?>

```

Le contrôleur permet de vérifier qu'on est connecté à la base de donnée et ensuite on appelle la méthode getGroupList qui va retourner un tableau que l'on met dans la variable \$groupList.

#### - groupAdd

Pour ajouter un groupe, le contrôle que l'on effectue :



```

if (isset($_POST['addGroup']) && !empty($_POST['name'])) {
    $usrGroup->id = $id;
    $usrGroup->name = $_POST['name'];
    if (isset($_POST['enable']) && $_POST['enable'] == 1) {
        $usrGroup->enable = $_POST['enable'];
    } else {
        $usrGroup->enable = 0;
    }
    $usrGroup->addGroup();
    header('location: manageGroup.php');
}

```

On vérifie que la fonction addGroup existe et que le champ name est rempli, alors on stock la valeur du name et pour mettre une valeur aux enable, on vérifie que nous avons dit que le groupe est activé et s'il l'a, on donne la valeur 1 sinon c'est 0. Une fois que le bouton ajouter est appuyé on retourne sur la liste des groupes où apparaît le groupe ajouter.

Pour modifier un groupe, les contrôles qu'on effectue :

```

if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $usrGroup->id = $id;
    $groupRecov = $usrGroup->selectGroup();
    $nameRecov = $usrGroup->name;
}

```

On vérifie qu'il y a bien un id et on récupère le name qui correspond à l'Id et qu'on stocke dans la variable \$nameRecov.

```

if (count($_POST) > 0) {
    if (empty($_POST['name'])) {
        $nameError = true;
    } else {
        $name = htmlspecialchars($_POST['name']);
    }
    if (isset($_POST['selectGroup'])) {
        $usrGroup->id = $_POST['idGroup'];
        $usrGroup->name = $_POST['name'];
        $usrGroup->enable = $_POST['enable'];
        $usrGroup->selectGroup();
    }
}

```

On vérifie que le bouton a été appuyé et que le name est rempli et on stocke les valeurs de l'Id le name et enable pour pouvoir le retrouver dans la liste des groupes.

## \* La vue

## - manageGroup

```

<div class="container-fluid">
  <div class="row">
    <div class="col-lg-12 usrTitleGroup">Liste des groupes</div>
    <table class="table table-striped grouplist">
      <thead>
        <tr>
          <th>N°</th>
          <th>Nom du groupe</th>
          <th>Activer</th>
        </tr>
      </thead>
      <tbody>
        <?php foreach ($grouplist as $groupDetail) { ?>
          <tr>
            <td><?php echo $groupDetail['id'];
            ?></td>
            <td><?php echo $groupDetail['name'];
            ?></td>
            <td><?php if ($groupDetail['enable'] == 1) { ?>
              <td></td>
              <?php } else {
              ?>
              <td></td>
              <?php } ?>
            <td>
              <div class="btn-group" role="group">
                <a class="btn btn-warning usrModifGroup" href="groupAdd.php?id=<?php echo $groupDetail['id']; ?>"><span
class="glyphicon glyphicon-user"></span> Modifier</a>
                </div></td>
          </tr>
        <?php } ?>
      </tbody>
    </table>
    <a class="btn btn-default usrAddGroupBtn" href="groupAdd.php">Création Groupe </a>
  </div>
</div>

```

Nous avons créer un tableau avec le numéro d'id, le nom, savoir s'il est actif ou non et un bouton en lien avec le formulaire pour modifier le groupe.

Nous avons fait une boucle pour que tant qu'il trouve un groupe il stocke les informations et les inscribes dans notre tableau.

Pour l'activation et la désactivation, j'ai rajouté une condition, si le enable = 1 alors au lieu que dans le tableau n'apparaisse un, j'ai voulu faire apparaître une image de validation.

## - groupAdd

```

<a href="manageGroup.php" name="manageGroup.php">Retour à la liste</a>
<div class="col-lg-12 titleGroup">
  <div class="row">
    <?php if (empty($_GET['id'])) { ?>
      <h1>Créations groupes</h1>
    <?php } else {
    ?>
      <h1>Modifications groupes</h1>
    <?php } ?>
  </div>
</div>
<div class="form-group">
  <form action="groupAdd.php" method="post" class="cadre">
    <p><?php ?></p>
    <div class="form-group idk <?php echo $nameError ? 'has-error' : 'has-success' ?>>
      <label for="nom">Nom du groupe :</label>
      <input type="text" value="<?php echo $nameRecov ?>" name="name" class="form-control" />
      <label for="enable">Activation du groupe ? : </label><input type="checkbox" value=1 checked="" name="enable"/><br />
    </div>
    <?php if (!$id) { ?>
      <input type="submit" value="Ajouter un Groupe" name="addGroup"/><br/>
    <?php } else { ?>
      <input type="hidden" value="<?php echo $id ?>" name="idGroup"/>
      <input type="submit" value="Modifier un Groupe" name="changeGroup"/><?php } ?><br/>
    </form>
  </div>
<?php
include_once $_SERVER['DOCUMENT_ROOT'] . 'view/footer.php';
?>

```

Pour la modification et l'ajout d'un groupe nous avons utilisé le même formulaire.

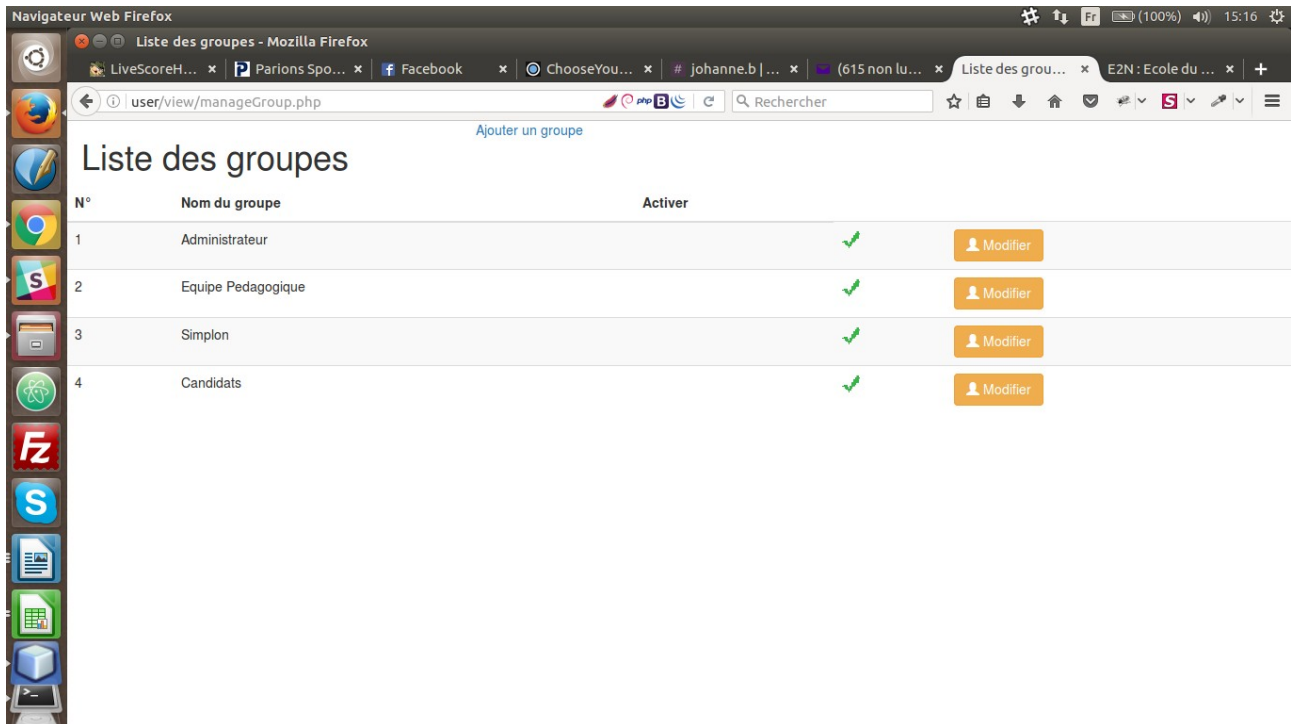
D'abord pour le titre, j'ai mis une condition disant que s'il n'y avait d'Id dans l'URL c'était le titre

création du groupe car c'est le cas quand on clique sur le bout « créations groupes » et dans le cas contraire le titre « Modifications groupes » apparaît.

On récupère la valeur de l'id, s'il n'a pas de valeur alors le bouton « Ajouter un Groupe » apparaît pour valider le formulaire et dans le cas contraire on cache le ce bouton et c'est le bouton « Modifier un Groupe » qui apparaît.

Pour le name, on demande de récupérer la valeur dans la variable \$nameRecov et alors apparaît le nom du groupe quand on le modifie.

- Aperçu de la première page



## **8- Conclusions**

Ce projet m'a permis de comprendre plein choses que je pensais avoir compris pendant la formation et qui finalement n'étaient pas clair.

J'ai aussi compris que tout seul, et quand on est en difficulté, il faut savoir trouver de l'aide auprès des membres de son équipe et de n'avoir pas peur d'aller les déranger.

C'est un projet difficile à faire en peu de temps, avec la pression du titre professionnel en bout de course. Nous savons que toute les fonctionnalités qui était prévu ne sont pas au rendez-vous, mais nous savons tous que nous avons fait de notre mieux et que nous nous sommes dépasser pour le réaliser. C'est une bonne expérience pour notre futur métier.