Institut de la Francophone pour l'Informatique
Promotion 7

Projet

Gestion d'emploi du temps et agenda personnel en ligne

Version 1.0

Implémentation

Tuteur :

Nguyen Hong Quang

Groupe :

Nguyen Thanh Bon

Pham Thi Ngoc Diem

Dang Thanh Ha

Hanoï 02.2003

Client du ET&A

Nom et prénom : Nguyen Hong Quang

Adresse :

Téléphone :

Courrier électronique : nhquang@ifi.edu.vn

Fournisseur de l'ET&A

Organisation : Institut de la Francophonie pour l'Informatique.

Groupe : Promotion 7

Tuteur : Nguyen Hong Quang

Membres : Groupe 7

Nguyen Thanh Bon email : **ntbon@ifi.edu.vn**

Pham Thi Ngoc Diem email : **ptndiem@ifi.edu.vn**

Dang Thanh Ha email : **dtha@ifi.edu.vn**

Révision

Auteur	Date	Version	Description
Pham Thi Ngoc Diem	10/02/03	0.8	Créer et compléter

TABLE DE MATIERE

1.Introduction

1.1.Objectifs du document

Ce document est pour but de :

- Procéder à une implémentation de l'ensemble des composants, des pages, des fonctionnalités requises pour le système de gestion des emplois du temps et agendas en ligne(ET & A).

- Documenter les interfaces entre les composants, les pages, les fonctions.

Ce document s'adresse principalement aux membres de l'équipe de développement du logiciel ET & A (développeur du système) ainsi qu'aux responsables techniques de côté du client. Il compose des implémentations des ensembles des composants.

1.2.Portée du produit

Le produit est un système qui sert à gérer les agendas personnels et les emplois du temps sur Web. Ce système est développé pour adapter d'abord à la situation actuelle de l'IFI. Pour les autres établissements ou pour l'IFI dans le futur, il faut éventuellement des modifications.

Il y a 4 parties principales dans le système. Ce sont :

- Partie administrative

- Partie de gestion des agendas personnels

- Partie de gestion des emplois du temps

- Parie d'aider à organiser une réunion.

La dernière partie est une partie optionnelle. L'équipe développera cette partie en fonction de temps disponible.

Voir le cahier des charges pour la description plus détaillé du système au point de vue du client.

Il y a trois types d'utilisateur du système :

1. Le public (y compris les étudiants de l'IFI) : consulter l'emploi du temps

2. Les personnels de l'IFI : login/logout au système, changer le mot de passe, gérer son agenda, consulter agenda des autres

3. Administrateur du système : login/logout au système, changer le mots de passe, gérer les utilisateurs, gérer les emplois du temps, gérer la liste de cours, ... Il faut que l'administrateur comprendre bien le processus de manipuler un emploi du temps ainsi que connaître bien les contraintes de l'IFI.

Comme ce système est développé sur le Web, alors il ne demande pas de condition particulière à la machine de l'utilisateur. La base de données et les Web page se trouvent sur le serveur.

1.3.Définition, acronymes et abréviations

- ET & A : emplois du temps et agendas en ligne
- BD : base de données

1.4. Documents de références

- Cahier des charges
- Dossier de spécification
- Dossier de conception

1.5. Aperçus du document

Ce document se compose de trois parties :

- La première partie rappeler l'introduction globale du système.
- Dans deuxième partie de ce document, on va présenter des décisions stratégiques pour l'implémentation de logiciel, ils sont basés sur le dossier de conception, des standards et les contraints du logiciel...
- Dans troisième partie, on va présenter la partie de l'implémentation de chaque composant du système.

2.Décisions stratégiques de l'implémentation

2.1.Environnement d'implémentation

EA&T est installé sur le système d'exploitation LINUX et la base de donnés choisi est mySQL.

2.2.Démarche d'implémentation

Le système d'implémentation se compose de trois paquets correspondants à trois types d'utilisateur avec les fonctionnalités principales du système. Tout ça sera réalisé selon l'ordre suivant :

Côté d'administrateur :

- Gestion de personnel : ajouter, modifier, supprimer
- Gestion de cours : ajouter, modifier, supprimer
- Gestion de salle : ajouter, modifier, supprimer
- Gestion d'ET : ajouter, modifier, supprimer

Côté de personnel:

- Gestion d'agenda : ajouter, modifier, supprimer
- Proposer des unions

Côté de public :

- Voir des Ets
- Voir des cours
- Voir des salles

2.3.Standards à suivre

- Modéliser : UML.
- Coder : << Coding convention standard. >>
- Ref : http://utvikler.start.no/code/php_coding_standard.html

2.4.Hypothèse et dépendances

Le but de ce dossier a pour fournir toutes les informations nécessaires pour que les développer puissent travailler bien même s'ils sont les débutants. Mais il est limité par le temps de travail (moins de 4 semaines) et la ressource d'humain (3 personnes) pour le concevoir. En outre, dans ce projet, le nombre de développer est fixé donc le niveau de détail de ce dossier sera fait en manière acceptable et compréhensible.

On décrit dans cette partie des hypothèses et dépendances que les utilisateurs doivent respecter :

- A côté d'administrateur : Il y a seulement l'administrateur du système qui a le droit de mettre à jour des ETs, cours et salles.
- A côté de personnel: chaque personnel n'a que le droit de mettre à jour des ses agendas que il a créés.

- A côté de public : Tous les utilisateurs ont le droit de voir des ETs, cours, salles.

2.5.Contraints généraux

Environnement matériel et logiciel

+ Une machine de côte serveur :

- OS: LINUX ou Windows
- Serveur Web Apache.
- Base de Donnée mySQL
- 256 M RAM
- 100 M Disque Dur.
- PHP 4.3 langage supporté.

+ Les machines de côte client:

- Navigateur de Web sous Linux ou Windows.
- HTML 1.1, javascripts langages supportés.

Vérification et validation de la conception :

- La conception doit décrire toutes les fonctions principales du système ET&A.
- Ces opérations nécessaires sont effectuées par tous les membres dans le groupe et le responsable.

Disponibilité de ressource :

On doit faire l'attention sur des ressources du système et du réseau comme :

- La mémoire : 64 Mo RAM
- L'espace de disque dur : >= 500 Mo
- La vitesse du processeur : >= 300 Mhz
- Le débit du réseau : >= 56 Kbit/s

Communication de réseau

- Le réseau marche bien
- La disponibilité et la fiabilité sont assurées.
- Permettre plusieurs utilisateurs à accéder en même temps.
- L'application est toujours stable.

Exigence de performance

- Temps de réponse : cela dépend de la complexité de requête de l'utilisateur, la vitesse du réseau et la capacité de traitement des applications liées, donc on ne peut donner des contraints claires. Mais, le temps total de réponse ne dépasse pas 1 minute.
- Précision de réponse : avec chaque requête de l'utilisateur, cette application doit rendre une seule réponse exacte ou un ensemble de réponses semblables. Cela dépend de chaque situation.

- Communication de réseau : Internet ou Intranet connecté.

Exigence d'inter-operabilité :

- Des informations et des données inter - communiquées doivent claires, simples et connues.
- Il existe toujours le mécanisme d'authentification et de vérification.

Base de données :

- Des types de données doivent être convenables avec le système de gestion de base de données MySQL
- Des structures de données entrées et sorties doivent être utilisables et visualisantes

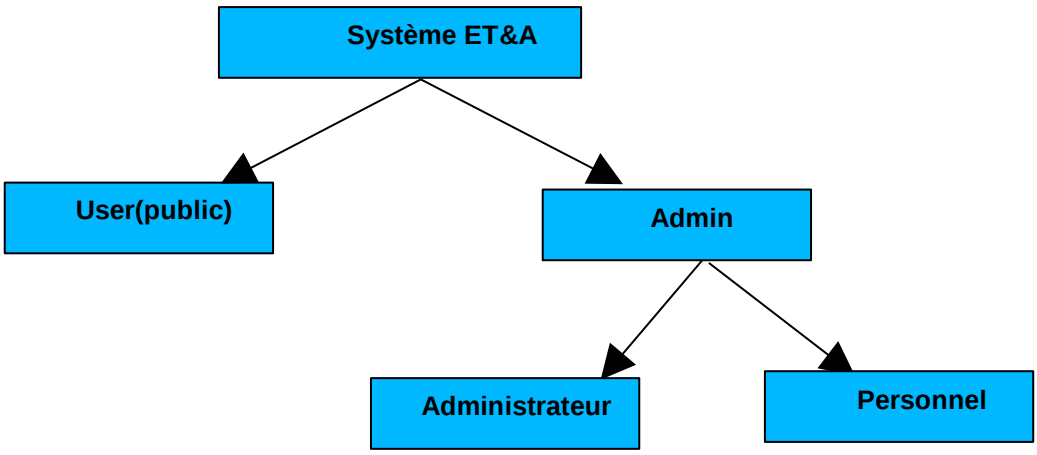
Tracabilité par rapport à la spécification

Pour la simplicité, Il y a quelques changement par rapport à la spécification et à la conception. Dans la partie de création d'emploi du temps: il n'y a pas d'opérations:

1. insérer une semaine
2. supprimer la semaine courrante
3. Mettre à jour la liste d'hoiraire
4. Quelques changements d'interface.

3.Implémentation

La structure d'implémentation du système est décrit comme suit:



Toutes les pages sont les fichiers de php.

3.1.Composants communs:

C'est le fichier db_mysql.inc et common.php. Ces composants sont utilisés en toutes pages.

3.1.1 db_mysql.inc

Ce fichier déclare une classe qui contient des opérations de base de données.

3.1.1.1Connexion de base de données

```
function connect($Database = "",$Host = "",$User = "",$Password = ""){
    /* Handle defaults */
    if ("" == $Database)
      $Database = $this->Database;

if ("" == $Host)
      $Host     = $this->Host;
    if ("" == $User)
      $User     = $this->User;
    if ("" == $Password)
      $Password = $this->Password;
    /* establish connection, select database */
    if ( 0 == $this->Link_ID ) {
      $this->Link_ID=mysql_pconnect($Host, $User, $Password);
      if (!$this->Link_ID) {
        $this->halt("connect($Host, $User, \$Password) failed.");
        return 0;
      }
      if (!@mysql_select_db($Database,$this->Link_ID)) {
        $this->halt("cannot use database ".$this->Database);
        return 0;
      }
    }
    return $this->Link_ID;
  }
```

3.1.1.2 Libérer la résultat d'une requête

```
function free() {
      @mysql_free_result($this->Query_ID);
      $this->Query_ID = 0;
  }
```

3.1.1.3 Exécuter une requête

```
function query($Query_String) {
    if ($Query_String == "")
      return 0;
    if (!$this->connect()) {
      return 0;
    };
    if ($this->Query_ID) {
      $this->free();
    }
```

```
    if ($this->Debug)
      printf("Debug: query = %s<br>\n", $Query_String);
    $this->Query_ID = @mysql_query($Query_String,$this->Link_ID);
    $this->Row   = 0;
    $this->Errno = mysql_errno();
    $this->Error = mysql_error();
    if (!$this->Query_ID) {
      $this->halt("Invalid SQL: ".$Query_String);
    }
    return $this->Query_ID;
  }
```

3.1.1.4 Obtenir l'ensemble de résultat d'une requête

```
  function next_record() {
    if (!$this->Query_ID) {
      $this->halt("next_record called with no query pending.");
      return 0;
    }
    $this->Record = @mysql_fetch_array($this->Query_ID);
    $this->Row   += 1;
    $this->Errno  = mysql_errno();
    $this->Error  = mysql_error();
    $stat = is_array($this->Record);
    if (!$stat && $this->Auto_Free) {
      $this->free();
    }
    return $stat;
  }
```

3.1.2 Common.php:

Ce fichier utilise db_mysql.inc. Il se compose des opérations concernant au établissement de session de travail, à la vérification de données et de droit de login de l'utilisateur au système...

3.1.2.1 Etablissement de session de travail

```
function set_session($param_name, $param_value)
{
  global ${$param_name};
  if(session_is_registered($param_name))
    session_unregister($param_name);
  ${$param_name} = $param_value;
  session_register($param_name);
}
```

3.1.2.2 Vérification de données

```
// Convert value for use with SQL statament
 function tosql($value, $type)
 {
  if(!strlen($value))
   return "NULL";
  else
   if($type == "Number")
    return str_replace (",", ".", doubleval($value));
   else
   {
    if(get_magic_quotes_gpc() == 0)
    {
     $value = str_replace("'","''",$value);
     $value = str_replace("\\","\\\\",$value);
    }
    else
    {
     $value = str_replace("\\'","''",$value);
     $value = str_replace("\\\"","\"",$value);
    }

    return "'" . $value . "'";
   }
```

3.1.2.3 Vérification de droit de login de l'utilisateur au système

```
// Verify user's security level and redirect to login page if needed
function check_security($security_level, $path_file)
{
 global $UserRights;
 if(!session_is_registered("UserID"))
    header ("Location: "+$path_file+"?querystring=" . urlencode(getenv("QUERY_STRING")) .
"&ret_page=" .  urlencode(getenv("REQUEST_URI")));
 else
          if (!session_is_registered("UserRights") || $UserRights < $security_level)
    header ("Location: "+$path_file+"?querystring=" . urlencode(getenv("QUERY_STRING")) .
"&ret_page=" .  urlencode(getenv("REQUEST_URI")));
}
```

3.2. Composant d'administrateur

Il se compose les pages de php suivantes :

1. adminAccueil.php
2. adminChangerMotDePasse.php
3. adminCoursInfo.php
4. adminCoursProfesseur.php
5. adminEmploiChangerEtat.php
6. adminEmploiCopier.php
7. adminEmploiCours.php
8. adminEmploiCours_backup.php
9. adminEmploiCoursProfesseur.php
10. adminEmploiCreation.php
11. adminEmploiCreationJour.php
12. adminEmploiListeHoiraire.php
13. adminEmploiParametre.php
14. adminEmploiSupprimer.php
15. adminHeader.php
16. adminIndex.php
17. adminListeCours.php
18. adminListeEmploi.php
19. adminListeProfesseur.php
20. adminListeSalle.php
21. adminLogout.php
22. adminMenu.php
23. adminProfesseurInfo.php
24. adminSalleInfo.php

Puisqu'il y a beaucoup de pages en cette partie, on ne s'intéresse qu'aux pages principales.

3.2.1 Page adminListeProfesseur.php

Définition	Cette page affiche les personnels de l'IFI qui sont pris de **Trombi**. Elle permet d'ajouter, de supprimer ou de modifier les informations concernant aux personnels.
Méthodes	- Targeted link. - Submit.
Associations	Cette page associe à la page: *adminProfesseurInfo*

Réalisation:

– **Prendre des donnés de trombi pour afficher:**

```
	global $db;
	global $position;
	global $trombi;
	global $sFileName;
	global $checklist;
	$iRecordsPerPage = 10;
	$iCounter = 0;
	$iPage = 0;
	$iNumPage;
	$bEof = false;
	$transit_params = "";
	$form_params = "";
	$sSQL = "SELECT Personne.PersonneID, " .
                                        "Personne.Nom, " .
                                        "Personne.Prenom, " .
                                        "Personne.Courriel, " .
                                        "Position.Libelle " .
                                        "FROM Personne LEFT JOIN Position USING (PositionCode) WHERE Position.PositionCode='" . tosql($position, "Number") . "'";
	$trombi->query($sSQL);
	$nextRecord = $trombi->next_record();
	$numRecord = $trombi->nf();
	$iNumPage = round(($numRecord-(($numRecord-1)%$iRecordsPerPage))/$iRecordsPerPage)+1;
	if (!$nextRecord) return;
	$iPage = get_param("PageNumber");
	if(!strlen($iPage)) $iPage = 1;
	else $iPage = intval($iPage);
	if(($iPage - 1) * $iRecordsPerPage != 0) {
		do {
		  $iCounter++;
		} while ($iCounter < ($iPage - 1) * $iRecordsPerPage && $trombi->next_record());
		$nextRecord = $trombi->next_record();
	}
	$iCounter = 0;
	while ($nextRecord && $iCounter < $iRecordsPerPage) {
		$fldPersonneID = $trombi->f("PersonneID");
		$fldNom = $trombi->f("Nom") . " ";
		$fldNom .= $trombi->f("Prenom");
		$fldCourrier = $trombi->f("Courriel");
		$fldLibelle = $trombi->f("Libelle");
		$nextRecord = $trombi->next_record();
```

```
$sSQL = "SELECT * FROM Personnel WHERE personnel_id=" . $fldPersonneID;
$db->query($sSQL);
$dbNextRecord = $db->next_record();
$fldNomDeLogin = $db->f("nom_de_login");
```

- **Supprimer des personnels du système:**

```
if ($submit) // do action
        switch ($submit) {
                case    "Supprimer":
                        if (!isset($skill_set)) break;
                        reset($skill_set); // Set the internal pointer of an array to its first element
                        while (list($id, $checked)=each($skill_set)) {
                                //if ($id == get_session("UserID")) continue;
                                $sWhere = "personnel_id=" . tosql($id, "Number");
                                $sSQL = "DELETE FROM Personnel WHERE " . $sWhere;
                                $db->query($sSQL);
                        }
                        break;
        }
```

3.2.2 Page adminProfesseurInfo.php

Définition	Cette page permet d'ajouter ou de modifier les informations concernant aux personnels.
Méthodes	- Submit.
Associations	Non

Réalisation:

- **Ajouter ou modifier les informations concernant aux personnels:**

```
switch (strtolower($action)) {
        case "enregistrer":
                test_error();
                if(strlen($error)) return;
                if ($formType == "ajouter") {
                        $sSQL = "INSERT INTO Personnel (" .
                                        "personnel_id, nom_de_login, " .
                                        "mot_de_passe, niveau_de_securite) " .
                                        "VALUES (" .
                                        tosql($personnelID, "Number") . ", " .
                                        tosql($fldNomDeLogin, "Text") . ", " .
                                        tosql($fldMotDePasse, "Text") . ", " .
                                        tosql(0, "Number") . ")";
                }
```

```
                else {
                        $sWhere = "personnel_id=" . tosql($personnelID, "Number");
                        $sSQL = "UPDATE Personnel SET " .
                                "nom_de_login =" . tosql($fldNomDeLogin, "Text") .
                                ", mot_de_passe =" . tosql($fldMotDePasse, "Text") .
                                ", niveau_de_securite =" . tosql(0, "Number");
                        $sSQL .= " WHERE " . $sWhere;
                }
                $db->query($sSQL);
                header("Location: adminListeProfesseur.php?PositionSelected=" . $position);
                break;
        case "annuler":
                header("Location: adminListeProfesseur.php?PositionSelected=" . $position);
                return;
}
```

3.2.3 Page adminListeCours.php

Définition	Cette page affiche les cours.Elle permet d'ajouter, de supprimer ou de modifier les cours.
Méthodes	- Targeted link. - Submit.
Associations	Cette page associe à la page: *adminCoursInfo*

Réalisation:

- **Supprimer**

```
if ($submit) // do action
        switch ($submit) {
                case    "Ajouter":
                        header("Location: adminCoursInfo.php?FormType=ajouter");
                        break;
                case    "Supprimer":
                        if (!isset($skill_set)) break;
                        reset($skill_set); // Set the internal pointer of an array to its first element
                        while (list($id, $checked)=each($skill_set)) {
                                //if ($id == get_session("UserID")) continue;
                                $sWhere = "cours_id=" . tosql($id, "Text");
                                $sSQL = "DELETE FROM Cours_professeurs WHERE " . $sWhere;
                                $db->query($sSQL);
                                $sSQL = "DELETE FROM Cours WHERE " . $sWhere;
                                $db->query($sSQL);
                        }
                        break;
}
```

- **Ajouter des cours et des professeurs qui sont responsables**

```
$sSQLadd = "INSERT INTO Cours (" .
                    "cours_id, " .
                    "nom_de_cours, " .
                    "nombre_dheures, " .
                    "plan_du_cours, " .
                    "description_de_cours) " .
                "VALUES (" .
                    tosql($fldCoursID, "Text") . "," .
                    tosql($fldNomDeCours, "Text") . "," .
                    tosql($fldNombreDHeures, "Number") . "," .
                    tosql($fldPlanDeCours, "Text") . "," .
                    tosql($fldDescriptionDeCours, "Text") . ")";
        $sWhere = "cours_id=" . tosql($coursID, "Text");
        $sSQLupdate = "update Cours set " .
            "cours_id=" . tosql($fldCoursID, "Text") . "," .
            "nom_de_cours=" . tosql($fldNomDeCours, "Text") . "," .
            "nombre_dheures=" . tosql($fldNombreDHeures, "Number") . "," .
            "plan_du_cours=" . tosql($fldPlanDeCours, "Text") . ", " . "description_de_cours=" .
tosql($fldDescriptionDeCours, "Text");
        $sSQLupdate = $sSQLupdate .     " WHERE " . $sWhere;

        switch (strtolower($action)) {
            case "ajouter":
                test_error();
                if(strlen($error)) return;
                $db->query($sSQLadd);
                header("Location: adminCoursProfesseur.php?CoursID=" . $fldCoursID);
                break;
            case "enregistrer":
                test_error();
                if(strlen($error)) return;
                $db->query($sSQLupdate);
                $sSQL = "update Cours_professeurs set cours_id=" .
                        tosql($fldCoursID, "Text") . " WHERE cours_id=" .
            tosql($coursID,"Text");
                $db->query($sSQL);
                $sSQL = "SELECT * FROM Cours_professeurs WHERE cours_id=" .
                tosql($fldCoursID,"Text");
                $db->query($sSQL);
                if (!$db->next_record())
                        header("Location:       adminCoursProfesseur.php?CoursID="      .
$fldCoursID);
                else
```

```
                                        header("Location: adminListeCours.php");
                                break;
                        case "annuler":
                                header("Location: adminListeCours.php");
                                return;
                        case "modifier_profs":
                                test_error();
                                if(strlen($error)) return;
                                if ($formType=="ajouter")
                                        $db->query($sSQLadd);
                                else
                                        $db->query($sSQLupdate);

                                header("Location: adminCoursProfesseur.php?CoursID=" . $fldCoursID);
                                break;
        }
```

3.2.4 Page adminListeSalle.php

L'implémentation est analoguement à la page adminListeCours.php

3.2.5 Page adminListeEmploi.php

Définition	Cette page affiche les noms des emplois du temps (ET). Elle permet de créer, de supprimer ou de modifier ... des emplois du temps.
Méthodes	- Targeted link. - Submit.
Associations	Cette page associe à la page : *adminEmploiParametre, adminEmploiCopier, adminEmploiChangerEtat.*

Réalisation:

```
if ($submit) // do action
        switch ($submit) {
                case "Creer":
                        header("Location: adminEmploiParametre.php?FormType=ajouter");
                        break;
                case "Copier":
                        header("Location: adminEmploiCopier.php");
                        break;
                case "Changer Etat":
                        header("Location: adminEmploiChangerEtat.php");
                        break;
                case "Supprimer":
                        if (!isset($skill_set)) break;
                        reset($skill_set); // Set the internal pointer of an array to its first element
                        while (list($id, $checked)=each($skill_set)) {
```

```
                                        //if ($id == get_session("UserID")) continue;
                                        $sWhere = "emploi_id=" . tosql($id, "Text");
                                        $sSQL = "DELETE FROM Emploi WHERE " . $sWhere;
                                        $db->query($sSQL);
                                        $sSQL = "DELETE FROM Emploi_cours WHERE " . $sWhere;
                                        $db->query($sSQL);
                                        $sSQL = "DELETE FROM Emploi_jour WHERE " . $sWhere;
                                        $db->query($sSQL);
                                        $sSQL = "DELETE FROM Emploi_horaire WHERE " . $sWhere;
                                        $db->query($sSQL);
                                }
                                break;
        }
```

3.2.6 Page adminEmploiCreation.php

Définition	Cette page affiche les emplois du temps en détaillé en semaine. Elle permet de mettre à jour les jours de la semaine.
Méthodes	- Targeted link. - Submit.
Associations	Cette page associe à la page : *adminEmploiSeanceJour.php*

Réalisation:

- **Permet de ajouter une période en l'emploi du temps .**

```
function form1_Action($action) {
        global $db;
        global $error;
        global $formType;
        global $fldCoursID;
        global $fldSalleID;
        global $emploiID;
        global $horaireID;
        global $seanceDate;
        global $semaineselected;

        $sSQLadd = "INSERT INTO Emploi_jour (" .
                        "emploi_id, " .
                        "cours_id, " .
                        "salle_id, " .
                        "horaire_id, " .
                        "date) " .
                "VALUES (" .
                        tosql($emploiID, "Text") . "," .
```

```
                tosql($fldCoursID, "Text") . "," .
                tosql($fldSalleID, "Number") . "," .
                tosql($horaireID, "Number") . "," .
                tosql($seanceDate, "Date") .")";
        $sWhere = "emploi_id=" . tosql($emploiID, "Text");
        $sWhere .= "AND date=" . tosql($seanceDate, "Date");
        $sWhere .= "AND horaire_id=" . tosql($horaireID, "Number");
        $sSQLupdate = "UPDATE Emploi_jour SET " .
                "cours_id=" . tosql($fldCoursID, "Text") . "," .
                "salle_id=" . tosql($fldSalleID, "Number");
        $sSQLupdate = $sSQLupdate .     " WHERE " . $sWhere;

        switch (strtolower($action)) {
                case "enregistrer":
                        test_error();
                        if(strlen($error)) return;
                        if ($formType=="ajouter")
                                $db->query($sSQLadd);
                        else
                                $db->query($sSQLupdate);
                        header("Location: adminEmploiCreation.php?EmploiID=" . $emploiID .
"&SemaineSelected=" . $semaineselected);
                        break;
                case "annuler":
                        header("Location: adminEmploiCreation.php?EmploiID=" .
                        $emploiID . "&SemaineSelected=" . $semaineselected);
                        break;
        }
}
```

3.2.7 adminEmploiCopier.php

Définition	Cette page permet de copier un ET déjà existe à un nouvel ET.
Méthodes	- Submit.
Associatio	Non

Réalisation:

- **Copier des parammètres**

```
function copy_emploi($emploiID)
{       global $db;
        global $nouveauEmploiID;
        global $nouveauNomDeLEmploi;
```

```
$sWhere = " WHERE emploi_id=" . tosql($emploiID, "Text");
$sSQL = "SELECT * FROM Emploi " . sWhere;
$db->query($sSQL);
$db->next_record();
$fldNomDeLEmploi = $db->f("nom_de_lemploi");
$fldNombreDeSemaine = $db->f("nombre_de_semaine");
$fldPremierJour  = $db->f("premier_jour");
$fldDernierJour = $db->f("dernier_jour");
$fldNombreSeance = $db->f("nombre_seance");
$fldDuree = $db->f("duree");
$fldDebutMatin = $db->f("debut_matin");
$fldNombreHeureMatin = $db->f("nombre_heure_matin");
$fldDebutMidi = $db->f("debut_midi");
$fldNombreHeureMidi = $db->f("nombre_heure_midi");
$fldEtat = $db->f("etat");
$fldSalleID = $db->f("salle_id");
$sSQL = "INSERT INTO Emploi (" .
                    "emploi_id, " .
                    "nom_de_lemploi, " .
                    "nombre_de_semaine, " .
                    "premier_jour, " .
                    "dernier_jour, " .
                    "nombre_seance, " .
                    "duree, " .
                    "debut_matin, " .
                    "nombre_heure_matin, " .
                    "debut_midi, " .
                    "nombre_heure_midi, " .
                    "etat, " .
                    "salle_id) " .
            "VALUES (" .
                    tosql($nouveauEmploiID, "Text") . "," .
                    tosql($nouveauNomDeLEmploi, "Text") . "," .
                    tosql($fldNombreDeSemaine, "Number") . "," .
                    tosql($fldPremierJour, "Date") . "," .
                    tosql($fldDernierJour, "Date") . "," .
                    tosql($fldNombreSeance, "Number") . "," .
                    tosql($fldDuree, "Number") . "," .
                    tosql($fldDebutMatin, "Time") . "," .
```

```
                        tosql($fldNombreHeureMatin, "Number") . "," .
                        tosql($fldDebutMidi, "Time") . "," .
                        tosql($fldNombreHeureMidi, "Number") . "," .
                        tosql("creer", "Text") . "," .
                        tosql($fldSalleID, "Number") .
                        ")";
        $db->query($sSQL);
}
```

- **Copier des hoiraires**

```
function copy_emploi_horaire($emploiID)
{
        global $db;
        global $db1;
        global $nouveauEmploiID;
        global $nouveauNomDeLEmploi;
        $sed = "SELECT eh.horaire_id, eh.debut, eh.fin, eh.seance FROM Emploi_horaire AS eh WHERE emploi_id=" . tosql($emploiID, "Text");
        $db1->query($sed);
        while ($db1->next_record()) {
                $fldHoraireID = $db1->f("horaire_id");
                $fldDebut = $db1->f("debut");
                $fldFin = $db1->f("fin");
                $fldSeance = $db1->f("seance");
                $sSQL = "INSERT INTO Emploi_horaire (horaire_id, emploi_id, debut, fin, seance) VALUES (" .
                        tosql($fldHoraireID, "Number") . ", " .
                        tosql($nouveauEmploiID, "Text") . ", " .
                        tosql($fldDebut, "Time") . ", " .
                        tosql($fldFin, "Time") . ", " .
                        tosql($fldSeance, "Number") . ")";
                $db->query($sSQL);
        }
}
```

- **Copier des cours**

```
function copy_emploi_cours($emploiID)
{
        global $db;
        global $db1;
        global $nouveauEmploiID;
```

```
	global $nouveauNomDeLEmploi;
	$sed = "SELECT cours_id FROM Emploi_cours WHERE emploi_id=" . tosql($emploiID, "Text");
	$db1->query($sed);
	while ($db1->next_record()) {
		$fldCoursID = $db1->f("cours_id");
		$sSQL = "INSERT INTO Emploi_cours (emploi_id, cours_id) VALUES (" .
			tosql($nouveauEmploiID, "Text") . ", " .
			tosql($fldCoursID, "Text") . ")";
		$db->query($sSQL);
	}
}
```

- **Copier des cours-professeur**

```
function copy_emploi_cours_professuer($emploiID)
{
	global $db;
	global $db1;
	global $nouveauEmploiID;
	global $nouveauNomDeLEmploi;
	$sed = "SELECT cours_id, personnel_id FROM Emploi_cours_professeur WHERE emploi_id=" . tosql($emploiID, "Text");
	$db1->query($sed);
	while ($db1->next_record()) {
		$fldCoursID = $db1->f("cours_id");
		$fldPersonnelID = $db1->f("personnel_id");
		$sSQL = "INSERT INTO Emploi_cours_professeur (emploi_id, cours_id, personnel_id) VALUES (" .
			tosql($nouveauEmploiID, "Text") . ", " .
			tosql($fldCoursID, "Text") . ", " .
			tosql($fldPersonnelID, "Text") . ")";
		$db->query($sSQL);
	}
```

3.3.Composant de personnel

Cette partie ne peux pas encore être effectuée parce qu'on n'a pas de temps.

3.4.Composant de public

Il se compose les pages de php suivantes:

1. Index.php
2. userAccueil.php
3. userEmploi.php

4. userEmploi2.php
5. userEmploiCours.php
6. userHeader.php
7. userListsCours.php
8. userCoursInfo.php
9. userListeSalle.php
10. userMenu.php

3.4.1 la page userEmploi.php

Définition	Cette page affiche des composantes pour aider utilisateur de choisir l'emploi du temps à voir.Elle permet de choisir un emploi du temps à voir et les informations concernant l'emploi du temps choisi: ET, cours, professeur.
Méthodes	- Targeted link. - Submit.
Associations	Cette page associe à la page: userEmploiCours, userEmploiProfesseur, userEmploi2

Réalisation

- **Prendre les noms d'ETs dans base de données et afficher dans le combobox**

```
<select name="ListeEmploi">
<?
                $sSQL = "SELECT * FROM Emploi";
                $db->query($sSQL);
                while ($db->next_record()) {
                        $fldEmploiID = $db->f("emploi_id");
                        $fldNomDeLEmploi = $db->f("nom_de_lemploi");
                        echo "<OPTION value=\"" . $fldEmploiID . "\"";
                        if ($position==$fldEmploiID) echo " SELECTED";
                        echo ">" . $fldNomDeLEmploi .  "</OPTION> <br>";
                }
?>
</select>
```

- **Les opérations pour afficher l'ET, Liste de cours**

```
 switch (strtolower($action)) {
        case "emploi du temps":
                header("Location: userEmploi2.php?EmploiID=" . $emploiselected);
                break;
        case "liste des cours":
                header("Location: userEmploiCours.php?EmploiID=" . $emploiselected);
                break;
```

```
                case "liste des professeurs":
                        header("Location: userEmploiProfesseur.php?EmploiID=" . $emploiselected);
                        break;
        }
```

3.4.2 la page userListeCours.php

Définition	Cette page affiche la liste de cours actuels de l'IFI.
Méthodes	- Targeted link.
Associations	Cette page associe à la page: userCoursInfo.php

Réalisation:

- **Prendre des données et afficher la liste de cours actuels de l'IFI.**

```
                                        $sSQL = "SELECT  m.cours_id as m_cours_id, " .
                                        "m.nom_de_cours as m_nom_de_cours, " .
                                        "m.nombre_dheures as m_nombre_dheures, " .
                                        "m.plan_du_cours as  m_plan_du_cours, " .
                                        "m.description_de_cours as m_description_de_cours " .
                                        "FROM Cours m ";
        $db->query($sSQL);
        $nextRecord = $db->next_record();
        $numRecord = $db->nf();
        $iNumPage = round(($numRecord-(($numRecord-1)%$iRecordsPerPage))/
$iRecordsPerPage)+1;
        if (!$nextRecord) return;
        $iPage = get_param("PageNumber");
        if(!strlen($iPage)) $iPage = 1;
        else $iPage = intval($iPage);
        if(($iPage - 1) * $iRecordsPerPage != 0) {
                do {
                  $iCounter++;
                } while ($iCounter < ($iPage - 1) * $iRecordsPerPage && $db->next_record());
                $nextRecord = $db->next_record();
        }
        $iCounter = 0;
        while ($nextRecord && $iCounter < $iRecordsPerPage) {
                $fldCoursID = $db->f("m_cours_id");
                $fldNomDeCours = $db->f("m_nom_de_cours");
                $fldNombreDHeures = $db->f("m_nombre_dheures");
                $fldPlanDeCours = $db->f("m_plan_du_cours");
```

```
$nextRecord = $db->next_record();
```

3.4.3 la page userListeSalle.php

Définition	Cette page affiche la liste de salles de l'IFI.
Méthodes	Non
Associations	Non

Réalisation:

- **Prendre des données et afficher la liste de salles de l'IFI.**

```
$sSQL = "SELECT m.salle_id as m_salle_id, " .
                                        "m.nom_de_salle as m_nom_de_salle, " .
                                        "m.description_de_salle as m_description_de_salle " .
                                        "FROM Salle m ";
        $db->query($sSQL);
        $nextRecord = $db->next_record();
        $numRecord = $db->nf();
        $iNumPage = round(($numRecord-(($numRecord-1)%$iRecordsPerPage))/
$iRecordsPerPage)+1;
        if (!$nextRecord) return;
        $iPage = get_param("PageNumber");
        if(!strlen($iPage)) $iPage = 1;
        else $iPage = intval($iPage);
        if(($iPage - 1) * $iRecordsPerPage != 0) {
                do {
                  $iCounter++;
                } while ($iCounter < ($iPage - 1) * $iRecordsPerPage && $db->next_record());
                $nextRecord = $db->next_record();
        }
        $iCounter = 0;
        while ($nextRecord && $iCounter < $iRecordsPerPage) {
                $fldSalleID = $db->f("m_salle_id");
                $fldSalleNom = $db->f("m_nom_de_salle");
                $fldSalleDescription = $db->f("m_description_de_salle");
                $nextRecord = $db->next_record();
```