



ECOLE DES MINES D'ALBI
C A R M A U X

GUIDE D'ADMINISTRATION DE L'APPLICATION DE GESTION DES STAGIAIRES EMAC

— *Projet GSI* —

Chef de projet utilisateur
Correspondant technique
Réalisation

Danielle DOLMIERE
Emmanuel OTTON
Sébastien MANNHEIM
Thomas VAN OUDENHOVE

— Janvier 2003 —

Table des matières

| | |
|---------------------------------------------------------------|-----------|
| Avant-propos | 4 |
| 1 Modélisation | 5 |
| 1.1 Modèle Conceptuel des Données | 5 |
| 1.2 Modèle Logique des Données | 6 |
| 2 Structure de l'application : enchaînement des écrans | 7 |
| 3 Le script fiche.cgi | 8 |
| 3.1 Fonctionnement | 8 |
| 3.2 Onglet <i>Données personnelles</i> | 8 |
| 3.3 Onglet <i>Données sur le stage</i> | 9 |
| 3.4 Onglet <i>Données financières</i> | 9 |
| 3.5 Onglet <i>Suivi du stagiaire</i> | 9 |
| 3.6 Onglet <i>Historique</i> | 10 |
| 4 Le script liste.cgi | 11 |
| 4.1 Affichage de la liste | 11 |
| 4.2 Recherche multi-critères | 11 |
| 5 Le script mandatement.cgi | 13 |
| 5.1 Demande du mois | 13 |
| 5.2 Tableau des mandatements | 13 |
| 5.3 Tableau d'engagement | 13 |
| 6 Le programme BdsVidange.pl | 14 |
| 6.1 Description | 14 |
| 6.2 En-tête du package | 14 |
| 6.3 Subroutines | 15 |
| 6.3.1 Vidange | 15 |
| 6.3.2 ConfirmerDates | 15 |
| 6.3.3 InfoNotifSignee | 15 |
| 7 Package BdsMep.pm | 17 |
| 7.1 Description | 17 |
| 7.2 Subroutines | 17 |
| 7.2.1 Entete | 17 |

| | | |
|-----------|-------------------------------|-----------|
| 7.2.2 | PiedPage | 17 |
| 7.2.3 | ConvertirDateEntree | 18 |
| 7.2.4 | ConvertirDateSortie | 18 |
| 7.2.5 | ComposerTextField | 18 |
| 7.2.6 | ComposerPopup | 19 |
| 7.2.7 | ComposerRadioGroup | 19 |
| 7.2.8 | ObtenirDroit | 20 |
| 7.2.9 | ListePays | 20 |
| 8 | Package BdsRecup.pm | 21 |
| 8.1 | Description | 21 |
| 8.2 | Subroutines | 21 |
| 9 | Package BdsLDAP.pm | 22 |
| 9.1 | Description | 22 |
| 9.2 | Subroutines | 22 |
| 9.2.1 | ListeTuteur | 22 |
| 9.2.2 | ListeService | 23 |
| 9.2.3 | Inverser | 23 |
| 9.2.4 | MailPersonne | 23 |
| 9.2.5 | Service | 23 |
| 9.2.6 | LoginPersonne | 24 |
| 10 | Package BdsMailAuto.pm | 25 |
| 10.1 | Description | 25 |
| 10.2 | En-tête du package | 25 |
| 10.2.1 | Packages utilisés | 25 |
| 10.2.2 | Variables communes | 25 |
| 10.3 | Subroutines | 26 |
| 10.3.1 | EnvoiMail | 26 |
| 10.3.2 | MailMoisAvant | 26 |
| 10.3.3 | MailNotifSignee | 26 |
| 10.3.4 | ChoixTextMail | 27 |
| 10.3.5 | Remplace | 27 |
| 11 | Package BdsGenpack.pm | 29 |
| 11.1 | Description | 29 |
| 11.2 | Subroutines | 29 |
| 11.2.1 | TableauGratif | 29 |
| 11.2.2 | TraiteTab | 30 |
| 11.2.3 | traitement_fichier | 30 |
| 11.2.4 | traite | 31 |
| 11.2.5 | Ouvrir | 31 |

Avant-propos

L'application de gestion des stagiaires EMAC est un projet proposé par Danielle DOLMIERE, qui en était le chef de projet utilisateur. Nous avons réalisé une application remplissant la plupart des fonctions demandées dans le cahier des charges. Ce document regroupe une petite documentation sur les fonctions et packages développés pour que l'application fonctionne.

Cette application contient deux scripts principaux : `fiche.cgi` et `liste.cgi`. Ils sont accessibles depuis le menu principal de l'application. Un troisième script (`mandatement.cgi`) sert à générer le tableau des gratifications à mettre en paiement pour un mois donné.

Nous avons choisi de donner le préfixe `Bds` – ou `bds` – (Base de Données Stagiaires) à certains fichiers utiles pour l'application (packages, fichiers \LaTeX à générer, ...).

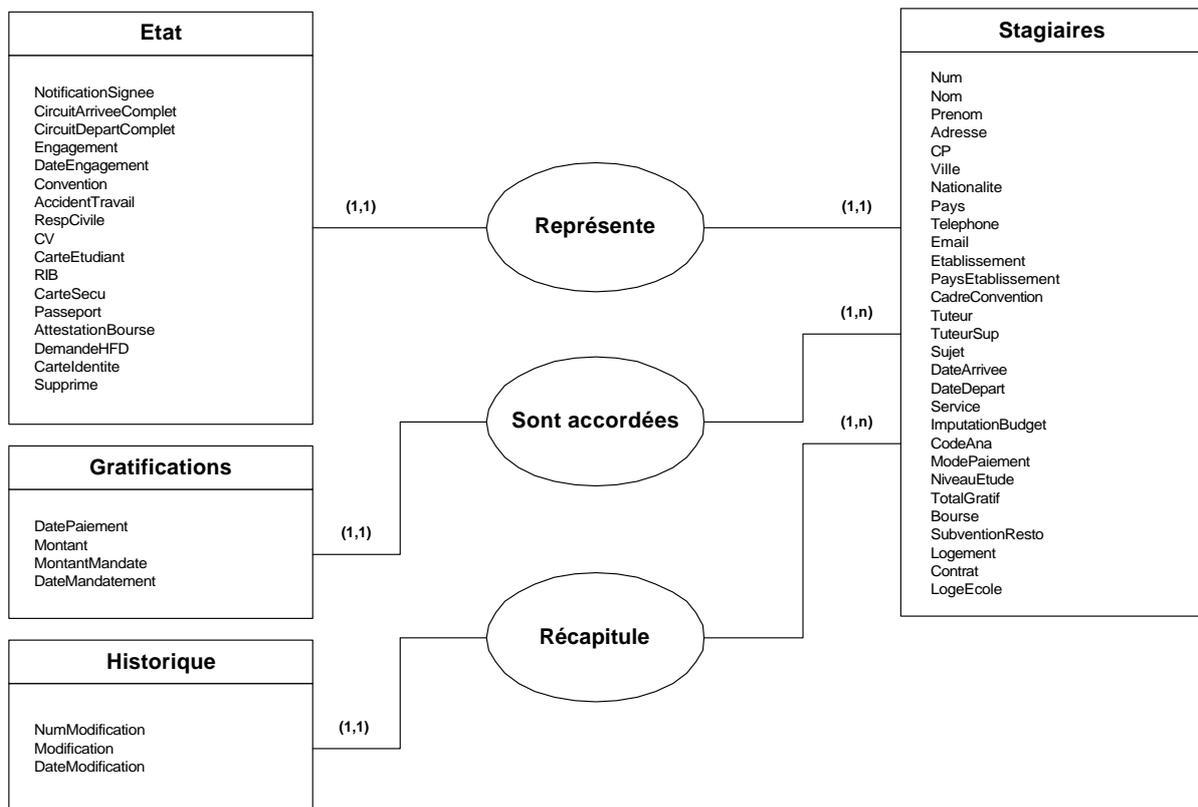
Le script `index.cgi` permet d'afficher le menu en fonction des droits de l'utilisateur. Les fonctions suivantes sont disponibles (si l'utilisateur possède tous les droits d'accès nécessaires) :

- Création d'une nouvelle fiche Stagiaire
- Liste des Stagiaires dont la notification est signée
- Liste des Stagiaires avant notification
- Recherche de stagiaires
- Tableau de mandatement

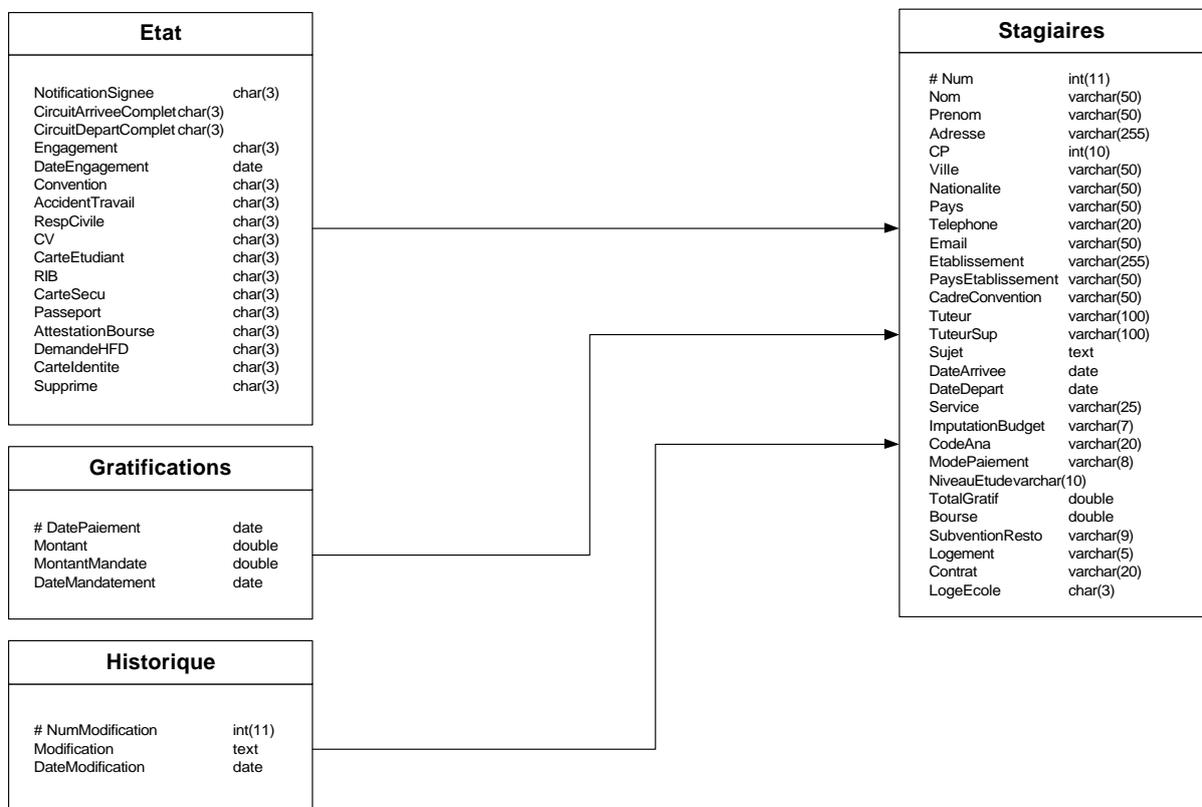
Chapitre 1

Modélisation

1.1 Modèle Conceptuel des Données

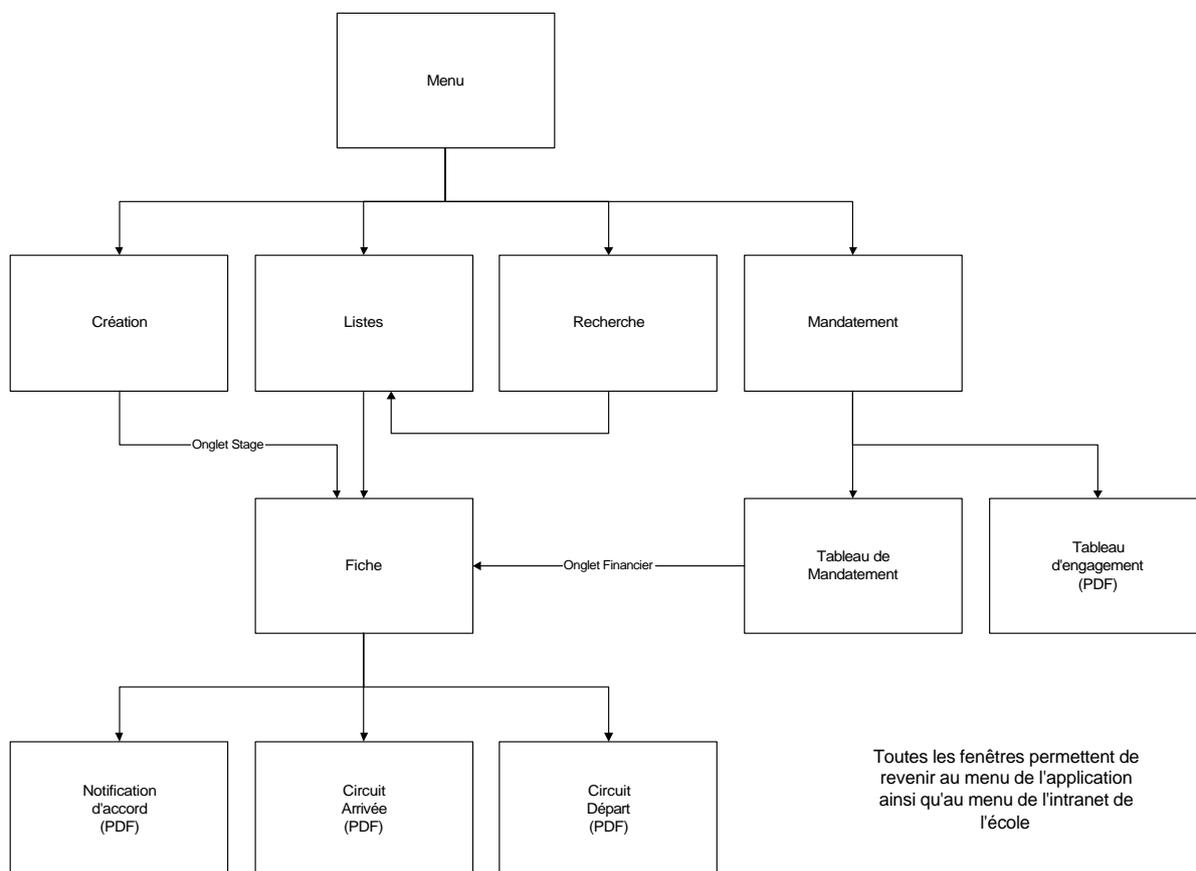


1.2 Modèle Logique des Données



Chapitre 2

Structure de l'application : enchaînement des écrans



Chapitre 3

Le script `fiche.cgi`

3.1 Fonctionnement

Ce script est la base de l'application. Toutes ses fonctions permettent d'afficher les informations sur un stagiaire, sous forme d'onglets. Les informations contenues dans les onglets sont détaillées dans les sections suivantes.

Les mêmes onglets permettent de saisir ou d'afficher les informations, selon que le script `fiche.cgi` est appelé avec un paramètre donnant l'identifiant (`'Num'` ou `'NumStagiaire'`, `int(11)`, selon les tables) ou sans paramètre. Lors d'un appel sans paramètres, tous les champs sont vides ; si on envoie le numéro d'un stagiaire, les champs non vides de la base de données sont déjà remplis, et parfois modifiables.

3.2 Onglet *Données personnelles*

On retrouve ici toutes les informations personnelles sur le stagiaire¹ :

- Nom du stagiaire (`'Nom'`, `varchar(50)`)
- Prénom (`'Prenom'`, `varchar(50)`)
- Adresse (`'Adresse'`, `varchar(255)`)
- Code Postal (`'CP'`, `int(10)`)
- Ville (`'Ville'`, `varchar(50)`)
- Nationalité (`'Nationalite'`, `varchar(50)`)
- Pays d'origine (`'Pays'`, `varchar(50)`)
- Numéro de téléphone (`'Telephone'`, `varchar(20)`)
- Adresse électronique (`'Email'`, `varchar(50)`)
- Établissement d'origine (`'Etablissement'`, `varchar(255)`)
- Pays de l'établissement d'origine (`'PaysEtablissement'`, `varchar(50)`)

Les champs obligatoires pour l'édition de la notification d'accord sont : le nom, le prénom, l'établissement d'origine et son pays.

¹les noms entre parenthèses sont ceux correspondants aux noms de champs de la base de données

3.3 Onglet *Données sur le stage*

Les informations entrées ici concernent le service d'accueil, le tuteur, ainsi que le sujet et les dates de stage, ...

- Le prénom et le nom du tuteur ('*Tuteur*', `varchar(100)`)
- Le prénom et le nom du tuteur suppléant ('*TuteurSup*', `varchar(100)`)
- Le sujet de stage ('*Sujet*', `text`)
- La date d'arrivée à l'école ('*DateArrivee*', `date`)
- La date de fin du stage ('*DateDepart*', `date`)
- Le service dans lequel le stagiaire va travailler ('*Service*', `varchar(25)`)
- Le cadre de la convention² ('*CadreConvention*', `varchar(50)`)
- Niveau d'étude du stagiaire ('*NiveauEtude*', `varchar(10)`)

Pour la notification d'accord, tous les champs de cet onglet doivent être renseignés.

3.4 Onglet *Données financières*

Dans cet onglet sont saisies toutes les informations concernant le cadencement des gratifications éventuelles et leur montant total, ainsi que le mode de paiement, ...

- Mode de paiement du stagiaire ('*ModePaiement*', `varchar(8)`)
- Montant total de sa gratification ('*TotalGratif*', `double`)
- Montant de bourse(s) éventuellement perçue(s) par ailleurs ('*Bourse*', `double`)
- Niveau de subvention au restaurant ('*SubventionResto*', `varchar(9)`)
- Imputation budgétaire ('*ImputationBudget*', `varchar(7)`)
- Code analysé ou marge ('*CodeAna*', `varchar(20)`)
- Contrat ('*Contrat*', `varchar(20)`)
- Le stagiaire est-il logé par l'école? ('*LogeEcole*', `char(3)`)
- Numéro du logement ('*Logement*', `varchar(5)`)

Pour éditer la notification, les montants des gratifications et leur échelonnement, ainsi que le niveau de subvention du restaurant doivent être saisis.

3.5 Onglet *Suivi du stagiaire*

Dans cet onglet, on retrouve des données booléennes permettant de savoir si le stagiaire a fait parvenir à l'école tous les documents nécessaires, si la notification d'accord a été signée, et, une fois le stagiaire arrivé, s'il a fini ses circuits départ et arrivée. Des boutons permettent aussi d'éditer les documents suivants, au format PDF : notification d'accord, circuit arrivée et circuit départ.

- Notification d'accord signée? ('*NotificationSignee*', `char(3)`)
- Circuit d'arrivée complet? ('*CircuitArriveeComple*', `char(3)`)
- Circuit de départ complet? ('*CircuitDepartComple*', `char(3)`)
- Date d'engagement³ ('*DateEngagement*', `date`)
- Convention signée? ('*Convention*', `char(3)`)

²DEA, ERASMUS, ...

³Correspond à la date de saisie de signature de la notification

- Attestation de couverture accident du travail? ('*AccidentTravail*', char(3))
- Attestation de couverture des risques "responsabilité civile" et "individuelle accident"? ('*RespCivile*', char(3))
- CV apporté? ('*CV*', char(3))
- Copie de la carte d'étudiant? ('*CarteEtudiant*', char(3))
- Copie de la carte de sécurité sociale ou attestation de couverture prestation maladie? ('*CarteSecu*', char(3))
- 2 RIB? ('*RIB*', char(3))
- Copie de la carte d'identité (pour les ressortissants de l'UE)? ('*CarteIdentite*', char(3))
- Copie passeport avec visa ou carte de séjour (ressortissants hors UE)? ('*Passeport*', char(3))
- Attestation de bourse ou de revenu pendant le séjour? ('*AttestationBourse*', char(3))
- Demande au HFD? ('*DemandeHFD*', char(3))
- Un bouton permet d'activer le champ '*Supprime*', si le stagiaire ne vient pas (quelle qu'en soit la raison)

3.6 Onglet *Historique*

Cet onglet recense toutes les étapes de la création et/ou de la modification des données relatives au stagiaire. Les données viennent de la table '*historique*' et sont reliées à la table par le numéro du stagiaire (voir le Modèle Conceptuel des Données, page 5).

Dans cette table, on stocke en particulier les modifications effectuées et la date à laquelle ces modifications ont eu lieu. Le résultat apparaît sous cette forme :

27/01/2003

Modification des données par otton

Tuteur = Danielle DOLMIERE

TuteurSup = Jacques LAMOTHE

Sujet = eytjeytkezh zgfhs dh qseth qa gfdh q dfh q fhq dfhq dfhq dfh
qdfh q dfhq dfhqdfh qdh qdhfqd fh qdfh q ds hf qdgh q dhf q f hqdf QTR

DateArrivee = 1980-09-09

DateDepart = 2003-01-17

Service = /DR/CROMEPE

CadreConvention = Convention avec société tiers

NiveauEtude = < BAC

Chapitre 4

Le script `liste.cgi`

4.1 Affichage de la liste

La liste des stagiaires affiche une ligne pour chaque stagiaire comportant :

- le nom,
- le prénom,
- la date de début de stage,
- la date de fin de stage,
- le nom du tuteur,
- le service,
- un lien vers sa fiche.

Chaque nom de champ est cliquable et permet d'effectuer un tri (croissant ou décroissant) sur la colonne.

4.2 Recherche multi-critères

Le formulaire de recherche permet de saisir des critères de recherche pour les champs suivants :

- le nom du stagiaire,
- son numéro de logement,
- sa ville d'origine,
- son pays d'origine,
- le nom du tuteur,
- le nom du tuteur suppléant,
- le service,
- le cadre de convention,
- l'établissement d'origine,
- le pays de l'établissement,
- le niveau d'étude,
- l'imputation du budget,
- le mode de paiement,
- la notification d'accord (signée ou non),
- une date (recherche tous les stagiaires présents à l'école à cette date).

Par défaut, tous les champs sont réglés sur 'tout' et on retrouve la liste de tous les stagiaires. Une fois la recherche lancée, le programme affiche la liste des stagiaires pour lesquels les critères saisis sont vérifiés.

Chapitre 5

Le script `mandatement.cgi`

5.1 Demande du mois

Depuis le menu de départ, on accède à un formulaire de saisie d'un mois et d'une année. Ensuite, il est possible d'éditer le tableau de mise en paiement des gratifications, à signer par le secrétaire général, ou de saisir les mandatements effectifs de tous les stagiaires pour le mois saisi.

Pour pouvoir récupérer les stagiaires dont on veut générer le tableau ou saisir les mandatements, il faut générer les dates du premier et du dernier jour du mois pour envoyer ces paramètres aux fonctions générant le tableau des mandatements et le tableau des engagements.

5.2 Tableau des mandatements

Ce tableau sert à faciliter la saisie commune de mandatements pour plusieurs stagiaires pour un même mois. Il contient le nom des stagiaires bénéficiant d'une gratification pour le mois demandé, le montant de cette gratification et la date prévue de paiement de cette gratification. Il est alors possible de saisir le montant du mandatement (toujours inférieur à la gratification) et la date du mandatement (réglée par défaut sur la date du jour).

5.3 Tableau d'engagement

Le tableau d'engagement sort au format PDF, pour pouvoir l'imprimer. Il doit ensuite être signé par le secrétaire général de l'école pour que les stagiaires puissent bénéficier de leurs gratifications.

Chapitre 6

Le programme BdsVidange.pl

6.1 Description

Ce programme sert à :

1. Vidanger le répertoire temporaire de tous les fichiers générés par PDFLATEX
2. Scanner la base de données pour vérifier si une notification a été signée la veille ; si oui, le programme envoie des mails d'information à toutes les personnes concernées
3. Scanner la base de données pour savoir si un stagiaire part dans un mois ; si oui, un mail est envoyé au tuteur du stagiaire pour lui demander de confirmer la date de départ, et, le cas échéant, de demander la prolongation du stage (ou l'avancement des dates)

Ce programme est à lancer toutes les nuits, après minuit (configuration des dates).

6.2 En-tête du package

Packages utilisés

- `File` : `::Spec` : pour obtenir le chemin du répertoire temporaire (fonction `tmpdir`)
- `Date` : `::Calc` : pour obtenir la date du jour (fonction `Today`)
- `BdsMep` : pour utiliser les fonctions de conversion de dates
- `DBI` : pour l'accès aux bases de données
- `BdsMailauto` : pour l'envoi de mails
- `BdsRecup` : pour récupérer le tableau de hachage des correspondances entre noms des champs et valeurs de la base de données

Variables communes

- `$dbh` : le *Database Handler*
- `$sth` : ordre SQL pour récupérer les noms de champs de la base de données
- `@ListeChamps` : pour stocker les noms de champs obtenus avec `$sth`

6.3 Subroutines

6.3.1 Vidange

Fonction

Cette fonction sert à vider le répertoire temporaire de tous les fichiers générés pour l'application avec la compilation de fichiers `.tex`. Étant donné que tous les fichiers générés sont de la forme `bds-*.*`, cette subroutine les efface tous. Elle ne nécessite aucun argument.

Variables

- `$stempdir` : le chemin d'accès du répertoire temporaire

6.3.2 ConfirmerDates

Fonction

Cette fonction scanne la base de données et récupère tous les stagiaires dont la date de fin de stage est dans un mois. Ensuite, elle appelle des fonctions du package `BdsMailauto` (cf. 10) pour envoyer un mail au tuteur.

Variables

- `$annee`, `$mois`, `$jour` : stocker la date du jour
- `$Date` : stocker la date au format SQL
- `$sth` : ordre SQL pour obtenir tous les stagiaires qui finissent dans un mois
- `$RefLigne` : pointeur vers tableau associatif obtenu grâce à `$sth`
- `$Nom`, `$Valeur` : variables de boucles
- `%Correspondance` : tableau associatif des noms de champ et des valeurs associées de la table `'stagiaires'`

6.3.3 InfoNotifSignee

Fonction

Cette fonction scanne la base de données et récupère tous les stagiaires dont la notification a été signée la veille. Elle appelle alors des subroutines du package `BdsMailauto` (cf. 10) pour envoyer les mails d'information à toutes les personnes concernées.

Variables

- `$annee`, `$mois`, `$jour` : stocker la date du jour
- `$Date` : stocker la date au format SQL
- `$verif` : ordre SQL pour obtenir les numéros des stagiaires dont la notification a été validée la veille
- `@Listesuivi` : stocke le numéro du stagiaire, si la notification est signée, et la date de signature

- \$RefLigne : pointeur vers tableau associatif obtenu grâce à \$verif
- \$Nom, \$Valeur : variables de boucles
- \$sth : ordre SQL pour obtenir les informations sur les stagiaires dont les numéros ont été renvoyés par \$verif
- \$RefStagiaire : pointeur vers tableau associatif obtenu grâce à \$sth
- \$Attribut, \$Association : variables de boucles.

Chapitre 7

Package BdsMep.pm

7.1 Description

Ce package regroupe toutes les fonctions de mise en page html de l'application : les sous-routines pour générer les en-têtes et pieds de page, ainsi que les fonctions de conversion des dates au format SQL ou pour l'affichage, et d'autres fonctions d'affichage (champ texte, ...).

En-tête du package

Le package CGI est utilisé pour générer de l'html. De plus, les fonctions `Decode_Date_EU`, `Today` et `Date_To_Text` sont importées du package `Date : :Calc`.

Enfin, la variable `$q` représente un objet CGI, utilisé pour générer du code html.

7.2 Subroutines

7.2.1 Entete

Fonction

Cette fonction génère le code html du début du document. Elle prend en argument le titre de la fenêtre (pour le navigateur) et le titre de la page. Elle affiche alors un tableau contenant le logo de l'école, le titre de la page, et dans la dernière case, des liens vers l'accueil de l'application et vers l'accueil de l'Intranet de l'école.

Variables

- `$TitreFenetre` : titre de la fenêtre, dans la balise `<head>`
- `$Titre` : titre de la page, à afficher dans le tableau en début de page

7.2.2 PiedPage

Cette fonction referme la page html, et comporte des liens vers les adresses mail des concepteurs et du *superuser*, chargé de la maintenance.

7.2.3 ConvertirDateEntree

Fonction

Cette fonction convertit les dates au format SQL, soit `aaaammjj`. Elle prend en argument la date au format saisi par l'utilisateur.

Variables

- `$DateEntree` : la date à changer de format
- `($Annee, $Mois, $Jour)` : récupération de la date divisée en année, mois et jours après appel de la fonction `Decode_Date_EU`
- `$AaaaMmJj` : date, stockée au format SQL, ensuite retournée

7.2.4 ConvertirDateSortie

Fonction

Cette fonction convertit les dates du format SQL en un format plus 'lisible' par l'utilisateur, par exemple `jj/mm/aaaa`. Elle prend en argument la date sortie de la base de données.

Variables

- `$DateSgbd` : date sortie du SGBD, au format `aaaammjj`
- `$DateSortie` : date au format 'lisible', retournée par la fonction

7.2.5 ComposerTextField

Fonction

Cette fonction est le résultat de la factorisation de l'affichage de toutes les cases pour les champs à saisir. Elle demande en argument le nom à afficher devant le champ texte, le nom du champ texte, la taille et la longueur maximale, la valeur par défaut et le `colspan`, pour l'énoncé et le champ et un attribut indiquant si le champ est en lecture seule. Cet attribut vaut 'Oui' lorsque le champ doit être désactivé (fonction de l'état du dossier et du niveau de droits de l'utilisateur). Elle imprime alors en html le champ demandé, avec la légende correspondante.

Variables

- `$Enonce` : énoncé du champ texte
- `$Nom` : nom du champ texte
- `$Taille` : taille du champ
- `$Max` : longueur maximale du champ
- `$Defaut` : valeur par défaut
- `$Colspan` : `colspan` pour le champ texte
- `$ColspanEnonce` : `colspan` pour l'énoncé
- `$LectureSeule` : pour éventuellement désactiver le champ si la notification est signée

7.2.6 ComposerPopup

Fonction

Cette fonction est le résultat de la factorisation de l’affichage de toutes les cases pour les menus déroulants. Elle demande en argument le nom à afficher devant la liste, le nom de la liste, la taille, la valeur par défaut et le `colspan`, pour l’énoncé et le champ et un attribut indiquant si la liste est en lecture seule. Cet attribut vaut 'Oui' lorsque le menu doit être désactivé (fonction de l’état du dossier et du niveau de droits de l’utilisateur). Elle imprime alors en html la liste déroulante, avec la légende correspondante.

Variables

- `$Enonce` : énoncé du champ texte
- `$Nom` : nom du champ texte
- `$Taille` : taille du champ
- `$Max` : longueur maximale du champ
- `$Defaut` : valeur par défaut
- `$Colspan` : `colspan` pour le champ texte
- `$ColspanEnonce` : `colspan` pour l’énoncé
- `$LectureSeule` : pour éventuellement désactiver le champ si la notification est signée
- `@Liste` : tableau des valeurs à afficher dans la liste

7.2.7 ComposerRadioGroup

Fonction

Cette fonction est le résultat de la factorisation de l’affichage de toutes les cases pour les boutons radio. Elle demande en argument le nom à afficher devant les boutons, le nom du groupe de boutons, la taille, la valeur par défaut et le `colspan`, pour l’énoncé et le champ et un attribut indiquant si le groupe de boutons est en lecture seule. Cet attribut vaut 'Oui' lorsque les boutons doivent être désactivés (fonction de l’état du dossier et du niveau de droits de l’utilisateur). Elle imprime alors en html la liste déroulante, avec la légende correspondante.

Variables

- `$Enonce` : énoncé du champ texte
- `$Nom` : nom du champ texte
- `$Taille` : taille du champ
- `$Max` : longueur maximale du champ
- `$Defaut` : valeur par défaut
- `$Colspan` : `colspan` pour le champ texte
- `$ColspanEnonce` : `colspan` pour l’énoncé
- `$LectureSeule` : pour éventuellement désactiver le champ si la notification est signée
- `@Liste` : tableau des valeurs à afficher pour les boutons radio

7.2.8 ObtenirDroit

Fonction

Cette fonction renvoie le niveau de droits pour chaque utilisateur. Elle récupère d'abord le login de l'utilisateur connecté, puis obtient son service grâce aux fonctions de `BdsLDAP` (cf. 9). À chaque service correspond ensuite un niveau de droits (de 0 → aucun droit à 9 → administrateur). Elle demande en argument les noms du tuteur et du suppléant du stagiaire pour lequel on veut consulter la fiche.

Variables

- `$Tuteur`, `$TuteurSup` : les noms passés en argument
- `%ENV` : table stockant des variables d'environnement (`'REMOTE_USER'` pour avoir le login de l'utilisateur connecté)

7.2.9 ListePays

Cette fonction renvoie un tableau contenant la liste des pays, par ordre alphabétique.

Chapitre 8

Package BdsRecup.pm

8.1 Description

Ce package sert à regrouper certaines fonctions d'accès à la base de données. En particulier, il contient des sous-routines permettant de créer des tableaux associatifs ayant pour clés les noms des champs, et pour valeurs les données extraites des tables correspondants à un stagiaire.

En-tête du package

Cet en-tête contient seulement l'appel au package DBI et la définition du *Database Handler* y correspondant.

- `$dbh` : le *Database Handler*, permettant d'accéder à la base de données '*stagiaire*'

8.2 Subroutines

RecupTabCorrespondance

Fonction

Cette fonction renvoie le tableau associatif de correspondance entre les noms des champs de la table '*stagiaires*' et les valeurs associées pour un stagiaire. Elle demande en argument l'identifiant d'un stagiaire : son numéro.

Variables

- `$NumStagiaire` : le numéro du stagiaire, récupéré dans les arguments
- `$sth`, `@ListeChamps` : ordre SQL permettant de récupérer la liste des noms de champ de la table '*stagiaires*' et de la stocker dans `@ListeChamps`
- `$ctc` : ordre SQL accédant aux données de la table '*stagiaires*' pour le stagiaire n° `$NumStagiaire`
- `$RefLigne` : pointeur vers le tableau associatif voulu
- `%Correspondance` : tableau pointé par `$RefLigne`
- `$Attribut` : variable de boucle

Chapitre 9

Package BdsLDAP.pm

9.1 Description

Ce package regroupe toutes les fonctions d'accès à la base de données LDAP de l'école. Toutes les fonctions pour générer les menus déroulants comprenant des noms de personnes de l'école sont situées ici, ainsi que les sous-routines permettant d'obtenir une information (adresse mail, ...) sur une personne.

En-tête du package

On utilise le package `Net : :LDAP` pour pouvoir utiliser toutes les fonctions d'accès aux bases LDAP.

- `%utilisateurLDAP` : stocke les valeurs et noms de champ récupérés dans la base LDAP
- `@Tab`, `@Tab1`, `@Tab2` : tables temporaires
- `$serveur` : l'URL du serveur LDAP de l'école
- `$ldap` : connexion au serveur LDAP

9.2 Subroutines

9.2.1 ListeTuteur

Fonction

Cette sous-routine sert à récupérer la liste des tuteurs pour pouvoir ensuite l'afficher sous forme de menu déroulant dans l'application.

Variables

- `$resultat` : résultat de la recherche dans la base de données LDAP
- `$entree`, `$valeur`, `$var` : variables de boucles

9.2.2 ListeService

Fonction

Cette subroutine sert à récupérer la liste des services pour pouvoir ensuite l'afficher sous forme de menu déroulant dans l'application.

Variables

- \$resultat : résultat de la recherche dans la base de données LDAP
- \$entree, \$Valeur, \$Var : variables de boucles

9.2.3 Inverser

Fonction

Cette fonction inverse une chaîne de caractères composée de deux mots — exemple :

`monsieur Emac -> Emac monsieur`

Cette fonction est utile pour classer les noms de personnes par ordre alphabétique sur le nom, car les noms dans la base LDAP sont stockés en commençant par le prénom.

Variables

- \$chaîne : paramètre d'entrée, la chaîne à inverser, mais aussi variable de retour
- @Elements : tableau contenant les deux mots séparés

9.2.4 MailPersonne

Fonction

Cette subroutine sert à récupérer l'adresse mail d'une personne de l'école, connaissant son nom. Cette fonction prend en argument le nom de la personne et renvoie l'adresse mail de cette personne.

Variables

- \$Nom : nom de la personne dont il faut obtenir l'adresse mail
- \$resultat : résultat de la recherche dans la base de données LDAP
- \$Var : variable de sortie stockant l'adresse mail

9.2.5 Service

Fonction

Cette fonction est utilisée pour récupérer le service dans lequel travaille l'utilisateur connecté, pour gérer les droits. Elle demande en argument le login (`uid`) de l'utilisateur, et renvoie son service (`memberOfUnit`).

Variables

- \$login : le login de la personne dont on veut obtenir le service
- \$resultat : résultat de la recherche dans la base de données LDAP
- \$Var : variable de sortie stockant l'adresse mail

9.2.6 LoginPersonne

Fonction

Cette subroutine sert à récupérer le login d'une personne de l'école, connaissant son nom. Cette fonction prend en argument le nom de la personne et renvoie le login de cette personne.

Variables

- \$Nom : nom de la personne dont il faut obtenir le login
- \$resultat : résultat de la recherche dans la base de données LDAP
- \$Var : variable de sortie stockant le login

Chapitre 10

Package BdsMailAuto.pm

10.1 Description

Ce package sert à stocker les fonctions d'envoi de mail pour l'application de gestion des stagiaires. Ces fonctions sont en général appelées depuis un script Perl qui démarre toutes les nuits, scanne la base de données des stagiaires et appelle ensuite les fonctions contenues dans ce package pour générer les mails, et les envoyer aux bons destinataires.

10.2 En-tête du package

10.2.1 Packages utilisés

- DBI : pour utiliser les fonctions de connexion aux bases de données
- Net : :SMTP : pour pouvoir envoyer des mails par le protocole SMTP
- BdsLDAP : pour récupérer des valeurs de la base de données LDAP de l'école
- BdsRecup : pour charger un hash de correspondance entre les noms des champs et les valeurs de ces champs associés à un numéro de stagiaire

10.2.2 Variables communes

- \$dbh : la *poignée* vers la base de données '*stagiaire*'
- \$Ldap_Mail_Tuteur : variable contenant le mail du tuteur
- \$Ldap_Mail_Suppleant : idem, pour le suppléant
- %*ListeDestinataires* : un tableau associant une clé avec un destinataire, utilisé en particulier dans la subroutine MailNotifSignee
- %*ListeSujets* : idem, avec les sujets des mails
- %*ListeFichiersMails* : idem, avec les chemins d'accès aux fichiers contenant les textes des mails, avec des repères à remplacer par des valeurs de la base de données

10.3 Subroutines

10.3.1 EnvoiMail

Fonction

La subroutine EnvoiMail sert à envoyer un mail depuis le serveur mailhost.enstimac.fr ; le destinataire voit comme expéditeur "Application de gestion des stagiaires". Cette fonction prend en argument le(s) destinataire(s), le sujet, le message (corps de texte du mail) et éventuellement les destinataires en copie.

Variables

- \$To : liste des destinataires du mail
- \$Sujet : sujet du mail
- \$Message : texte du mail
- \$Copie : destinataires en copie éventuelle
- \$FromText : expéditeur du mail, qui apparaît dans la fenêtre du client mail
- \$From : expéditeur, nécessite une adresse au format [a-z]@[a-z].[a-z]
- \$smtp : URL du serveur de mails sortants utilisé

10.3.2 MailMoisAvant

Fonction

La subroutine MailMoisAvant sert à appeler la subroutine EnvoiMail avec les paramètres suivants : le destinataire est le tuteur, le sujet "Confirmation de date de fin de stage", et le message est une chaîne de caractères concaténée. Cette chaîne comprend un en-tête html, puis l'appel à la fonction ChoixTextMail (cf. ci-dessous). Elle demande en argument le numéro du stagiaire concerné.

Variables

- \$Num : numéro du stagiaire, récupéré dans les arguments
- %*Correspondance* : tableau associatif des correspondances
- \$Sujet : sujet du mail
- \$Destinataire : destinataire du mail
- \$Message : corps de texte du mail

10.3.3 MailNotifSignee

Fonction

Cette subroutine ressemble à MailMoisAvant. Elle prend pour argument le numéro du stagiaire dont la notification a été signée. Sa fonction est aussi d'appeler la subroutine EnvoiMail, mais les paramètres sont différents : elle récupère dans le tableau associatif %*ListeDestinataires* chaque destinataire du mail (ce mail est envoyé dès que la notification d'accord est signée), et

dans le tableau associatif %*ListeSujets* le sujet correspondant à chaque destinataire. Elle a ensuite le même fonctionnement que MailMoisAvant, pour chaque destinataire.

Variables

- \$Num : numéro du stagiaire, récupéré dans les arguments
- %*Correspondance* : tableau associatif des correspondances
- \$Sujet : sujet du mail
- \$Destinataire : destinataire du mail
- \$Message : corps de texte du mail

10.3.4 ChoixTextMail

Fonctions

Cette fonction est appelée par les précédentes. Elle prend en argument un paramètre (clé des tableaux associatifs %*ListeDestinataires* et %*ListeSujets*) et le tableau associatif %*Correspondance*. Elle ouvre ensuite le fichier contenant le corps du texte du mail correspondant au bon destinataire, puis y détecte les chaînes de caractère entourées de deux dièses (#). Enfin, elle appelle la fonction Remplace pour récupérer les bonnes valeurs dans la base de données. Enfin, elle retourne une chaîne de caractères comprenant le texte définitif du mail.

Variables

- \$retour : stocke la chaîne de caractères retournée par la fonction
- \$ligne : stocke temporairement chaque ligne du texte du mail pour y remplacer les #[a-z]# par la valeur du tableau associatif %*Correspondance*.
- \$param : repère à remplacer dans le texte du mail

10.3.5 Remplace

Fonction

Cette fonction sert à remplacer les repères dans les textes des mails par les valeurs correspondantes extraites de la base de données. Elle prend en argument la clé du repère à changer, et le %*Correspondance*, qui contient les valeurs de la base de données correspondantes.

Un ordre SQL sert à récupérer d'autres valeurs dans la table '*gratifications*', en particulier le nombre de versements pour le stagiaire, ainsi que les dates et montants de ces versements.

Variables

- \$param : clé des tableaux associatifs, en argument
- \$Valeur : chaîne traitée à retourner
- \$Nbre_versts : variable stockant le nombre de versements
- \$compte : référence vers l'ordre SQL pour récupérer les valeurs de la table '*gratifications*'
- \$RefLigne : référence vers le tableau associatif des valeurs retournées par \$compte
- @Montants : tableau des montants des versements, triés par date croissante
- @Dates : tableau des dates des versements, en ordre croissant

- \$i : variable de boucle

Chapitre 11

Package BdsGenpack.pm

11.1 Description

Ce package regroupe les fonctions de génération des fichiers PDF : notification d'accord, circuits arrivée et départ et tableau des gratifications. Il suffit alors dans les scripts CGI de faire appel à ces fonctions pour générer dynamiquement un PDF regroupant les informations de la base de données désirées. Ce fichier peut alors s'afficher dans le navigateur, et peut être imprimé.

En-tête du package

- CGI pour gérer l'affichage dans le navigateur des PDF
- File : :Temp pour créer des fichiers temporaires
- File : :Spec pour récupérer le chemin du répertoire temporaire
- DBI pour l'accès aux bases de données
- BdsRecup pour le tableau associatif %*Correspondance*

11.2 Subroutines

11.2.1 TableauGratif

Fonction

Cette fonction génère le tableau des gratifications. Elle demande en argument deux dates : le premier et le dernier jour d'un mois. Lorsqu'un repère (mot entouré de deux parenthèses) est trouvé dans le fichier à traiter, elle fait appel à la fonction `traite_tab`. Le document de sortie contient alors tous les stagiaires à gratifier pour un mois choisi par l'utilisateur.

Variables

- @**date_gratif** : premier et dernier jours du mois, dans les arguments
- IN : poignée vers `tab_gratif.tex`, fichier L^AT_EX de base contenant des repères à remplacer par des valeurs de la base de données
- \$tempdir : répertoire temporaire sur la machine

- `$FH` : poignée vers le fichier temporaire traité, généré avec un nom de la forme `bds-****.tex`
- `$fichier_traite` : le nom du fichier traité
- `$ligne` : chaque ligne lue du fichier à traiter
- `$fichier_traite_4_pdflatex` : pour stocker le chemin en arborescence Unix
- `$fichier_pdf` : nom du fichier PDF à afficher

11.2.2 TraiteTab

Fonction

Cette fonction va chercher dans la base de données les stagiaires qui ont une gratification prévue pour la période comprise entre les deux dates du tableau `@date_gratif` et génère pour chacun une ligne du tableau de mise en paiement des gratifications. Elle demande en argument le paramètre lu dans le fichier \LaTeX d'origine et le tableau des dates (premier et dernier jour du mois).

Variables

- `$dbh` : le *Database Handler*
- `@date_gratif` : premier et dernier jours du mois, dans les arguments
- `@ListeChamps` : tableau contenant la liste des champs à afficher dans le tableau sur le document PDF
- `$recup` : ordre SQL récupérant les stagiaires à payer pour le mois choisi
- `$RefLigne` : pointeur vers le tableau associatif résultant de l'ordre `$recup`
- `$LigneTex` : chaîne de caractères contenant une ligne à ajouter au fichier `$fichier_traite`
- `$TabTex` : chaîne de caractères contenant toutes les lignes à ajouter au fichier `$fichier_traite`
- `$Attribut` : variable de boucle

11.2.3 traitement_fichier

Fonction

Cette subroutine sert à générer les autres documents PDF : la notification d'accord et les circuits arrivée et départ. Elle prend en argument le nom du fichier à traiter (par exemple, `notif_accord.tex`) et le numéro du stagiaire pour lequel il faut éditer le document. Pour générer le fichier \LaTeX , cette subroutine fait appel à une autre fonction : `traite`, lorsque un mot entre deux repères est lu, pour le remplacer par des valeurs de la base de données.

Variables

- `$fichier_a_traiter` : nom du fichier \LaTeX à traiter
- `$NumStagiaire` : numéro du stagiaire pour lequel il faut générer le document
- `$tempdir` : répertoire temporaire dans lequel le fichier traité est généré
- `$FH` : poignée vers le fichier traité
- `$fichier_traite` : fichier \LaTeX traité
- `$fichier_traite_4_pdflatex` : fichier traité, avec le chemin sous la forme acceptée par l'OS
- `$fichier_pdf` : fichier PDF généré par $\PDF\LaTeX$, à envoyer au navigateur

11.2.4 traite

Fonction

Cette subroutine remplace les mots envoyés par la fonction `traitement_fichier` par les valeurs associées de la base de données. Pour cette association, `traite` appelle la fonction `RecupTabCorrespondance` cf. 8.2. Lorsque les repères du fichier original prennent des valeurs remarquables, cette fonction teste et ajoute éventuellement des données au fichier `LATEX`.

Variables

- `$FH` : poignée vers le fichier traité
- `$param` : nom à remplacer dans le fichier d'origine, la clé du tableau associatif donnant les correspondances
- `$NumStagiaire` : le numéro du stagiaire
- `$retour` : variable de retour
- `$i` : variable de boucle
- `%Correspondance` : tableau associatif des correspondances entre clés et valeurs de la base de données (généré grâce au package `BdsRecup`, cf. 8.2)
- `@Montants` : tableau stockant les montants des gratifications, par ordre chronologique
- `@Dates` : tableau stockant les dates par ordre croissant
- `$dbh` : le *Database Handler*
- `$compte` : ordre SQL pour remplir les tableaux `@Dates` et `@Montants`
- `$Nbre_versts` : nombre de versements
- `$RefLigne` : pointeur vers le tableau associatif renvoyé par l'ordre `$compte`

11.2.5 Ouvrir

Fonction

Cette subroutine sert à envoyer le fichier PDF généré par les fonctions précédentes au navigateur. L'utilisateur peut ensuite l'imprimer ou l'enregistrer sur le disque.

Variables

- `$filename` : nom du fichier à envoyer au navigateur