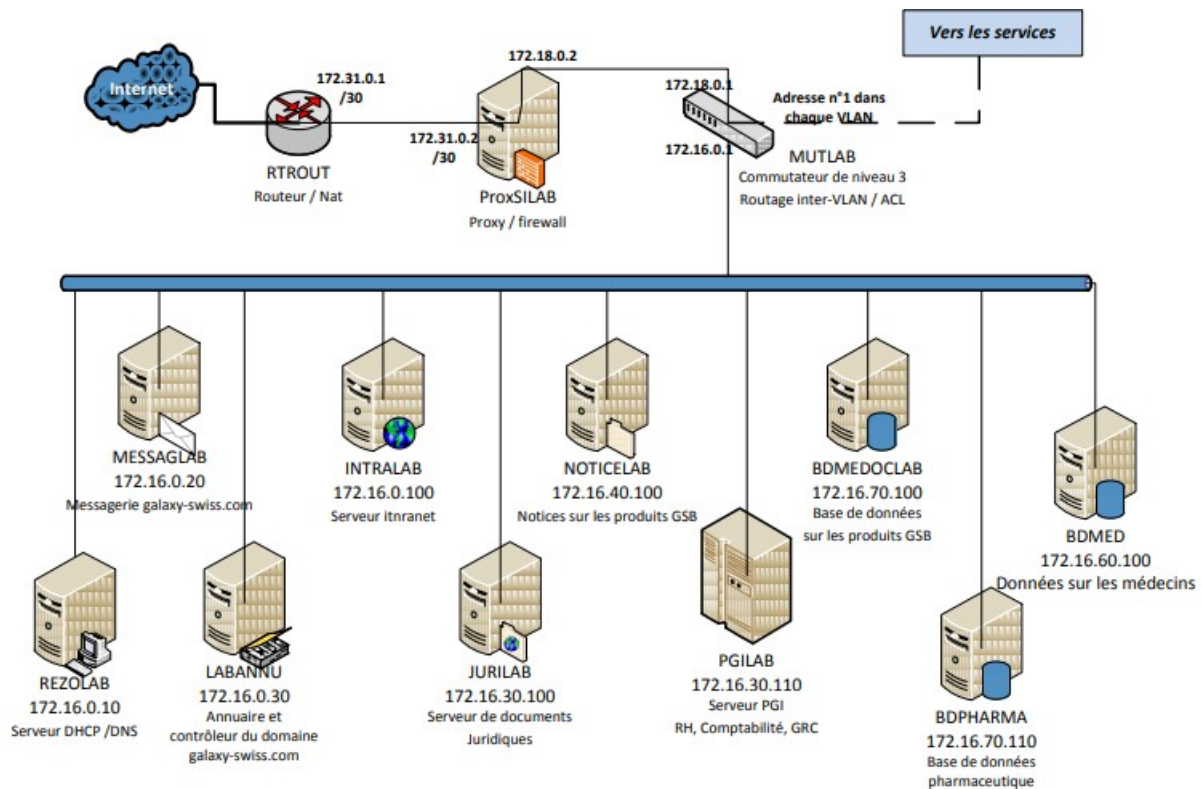


GSB - DOCUMENTATION TECHNIQUE

I. Présentation du contexte GSB

Le laboratoire Galaxy Swiss Bourdin (GSB) est issu de la fusion entre le géant américain Galaxy (spécialisé dans le secteur des maladies virales dont le SIDA et les hépatites) et le conglomérat européen Swiss Bourdin (travaillant sur des médicaments plus conventionnels), lui même déjà union de trois petits laboratoires. En 2009, les deux géants pharmaceutiques ont uni leurs forces pour créer un leader de ce secteur industriel. L'entité Galaxy Swiss Bourdin Europe a établi son siège administratif à Paris. Le siège social de la multinationale est situé à Philadelphie, Pennsylvanie, aux Etats-Unis.

Système actuel :



On souhaite améliorer le contact entre ces acteurs mobiles autonomes et les différents services du siège parisien de l'entité Europe. Il s'agit d'uniformiser la gestion du suivi des visites.

Une gestion unique de ces frais et remboursement pour l'ensemble de la flotte visite est souhaitée. Un accès plus direct aux données de personnel est nécessaire.

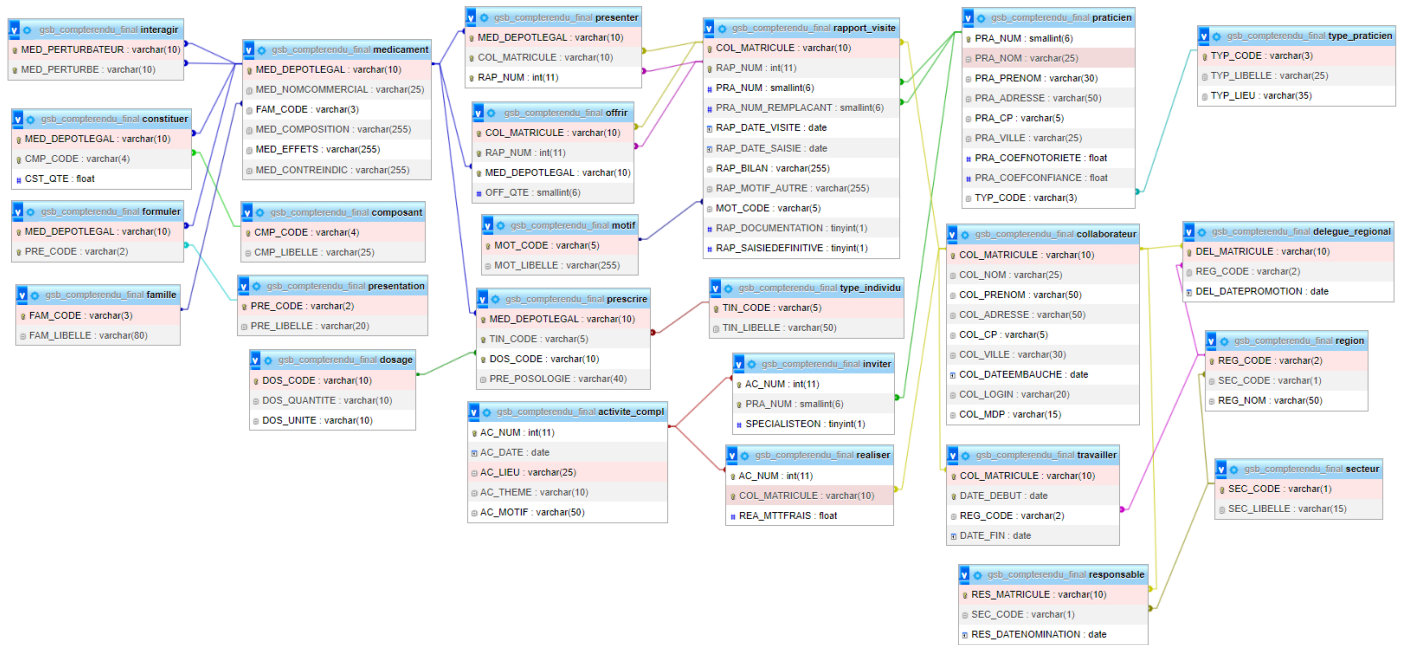
Les équipes du service développement auront notamment à produire puis à fournir les éléments applicatifs permettant :

- l'enregistrement d'informations en provenance des visiteurs
- la gestion des frais de déplacement

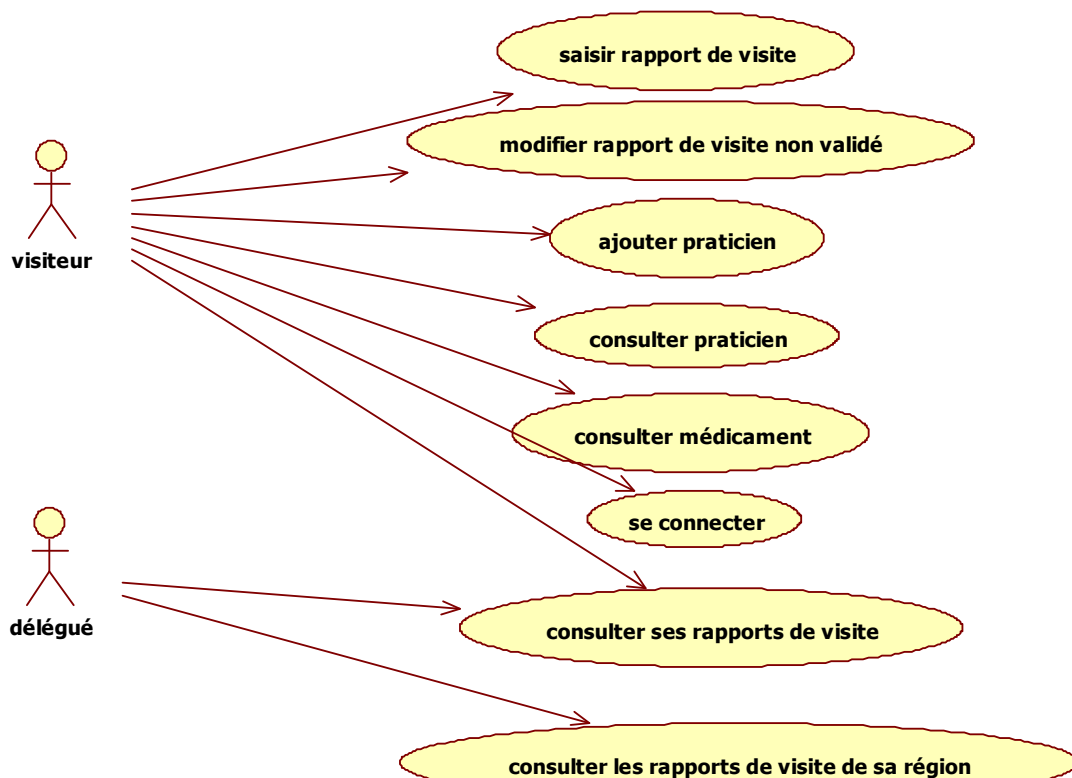
Les équipes du service Réseau et système fourniront les équipements et configuration réseau, ainsi que les ressources serveur nécessaires à héberger les applications mises à disposition de la flotte visite.

II. Analyse

A. Modélisation des données



B. Les traitements



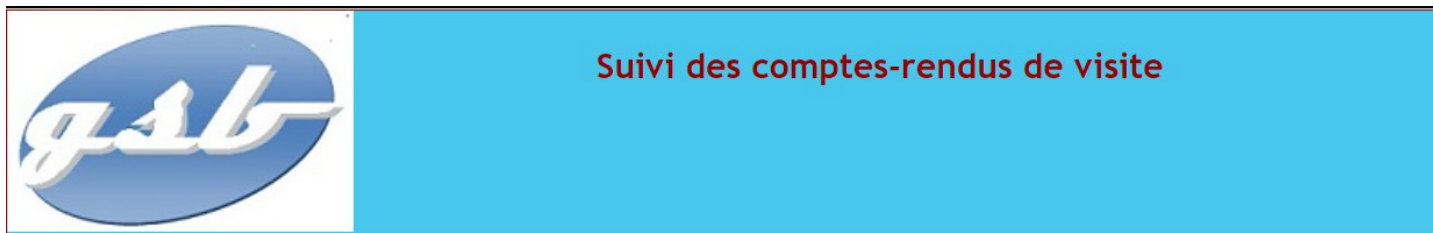
1. Connexion à l'application

Portail de connexion :

	Suivi des comptes-rendus de visite
Identification utilisateur	
Login* <input type="text"/>	
Mot de passe* <input type="password"/>	
<input type="button" value="Valider"/> <input type="button" value="Annuler"/>	

Accueil Délégué :

	Suivi des comptes-rendus de visite
David Andre Délégué régional Ajout compte-rendu Modification compte-rendu Consultation Praticiens Ajout praticien Consultation Médicaments Compte-rendus d'une période Compte-rendus de sa région Déconnexion	Bienvenue sur l'intranet GSB

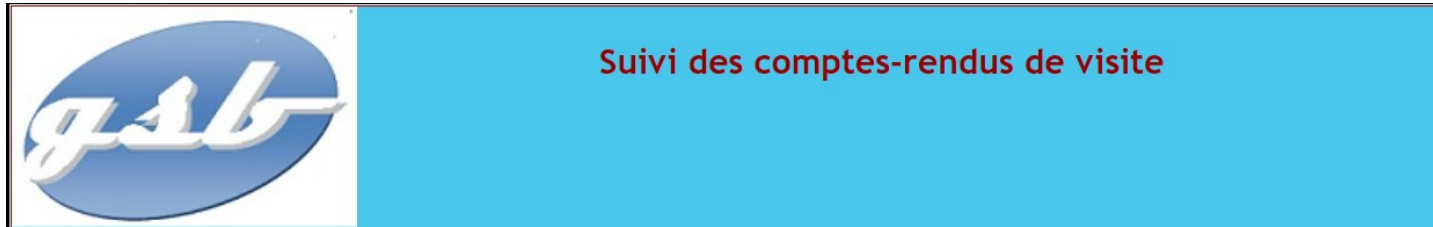


Eric Cadic

Responsable

[Consulter le compte rendu d'une période](#)
[consulter le compte Rendu de la Region](#)
[Consulte le compte rendu du visiteur](#)
[Consulte le compte rendu du secteur](#)
[Déconnexion](#)

Bienvenue sur l'intranet GSB



Louis Villechalane

Visiteur

[Ajout compte-rendu](#)
[Modification compte-rendu](#)
[Consultation Praticiens](#)
[Ajout praticien](#)
[Consultation Médicaments](#)
[Compte-rendus d'une période](#)
[Déconnexion](#)

Bienvenue sur l'intranet GSB

2. Saisie d'un compte rendu

Cas d'utilisation :

Page de saisie :



Suivi des comptes-rendus de visite

David Andre
Délégué régional
[Ajout compte-rendu](#)
[Modification compte-rendu](#)
[Consultation Praticiens](#)
[Ajout praticien](#)
[Consultation Médicaments](#)
[Compte-rendus d'une période](#)
[Compte-rendus de sa région](#)
[Déconnexion](#)

Compte-rendu de visite

NUMERO :

DATE VISITE :

PRATICIEN :

COEFFICIENT CONFIANCE :

REPLACANT : ☒ Boireaux Philippe

MOTIF :

Bilan

Eléments présentés
COCHEZ LA CASE : ☒ CHOISIR PRODUIT 1 :
COCHEZ LA CASE : ☐ CHOISIR PRODUIT 2 :
DOCUMENTATION OFFERTE : ☐

Echantillons
PRODUIT 1 : PRODUIT 2 :
PRODUIT 3 : PRODUIT 4 :
PRODUIT 5 : PRODUIT 6 :
PRODUIT 7 : PRODUIT 8 :
PRODUIT 9 : PRODUIT 10 :
DATE SAISIE :
SAISIE DEFINITIVE : ☐

Page de message de validation:



Suivi des comptes-rendus de visite

David Andre
Délégué régional
[Ajout compte-rendu](#)
[Modification compte-rendu](#)
[Consultation Praticiens](#)
[Ajout praticien](#)
[Consultation Médicaments](#)
[Compte-rendus d'une période](#)
[Compte-rendus de sa région](#)
[Déconnexion](#)

Le compte-rendu a été validé

3. Modification d'un compte rendu

Cas d'utilisation :

Page de choix du rapport :




Suivi des comptes-rendus de visite

David Andre
Délégué régional
Ajout compte-rendu
Modification compte-rendu
Consultation Praticiens
Ajout praticien
Consultation
Médicaments
Compte-rendus d'une période
Compte-rendus de sa région
Déconnexion

Modifier un compte rendu

Choisir le rapport numéro 11 ▼
Valider Effacer

Page de saisie :



Suivi des comptes-rendus de visite

David Andre
Délégué régional
Ajout compte-rendu
Modification compte-rendu
Consultation Praticiens
Ajout praticien
Consultation Médicaments
Compte-rendus d'une période
Compte-rendus de sa région
Déconnexion

Modifier un compte rendu

NUMERO : 11

DATE VISITE : 2024-02-06

PRATICIEN : Bessin-Grosdoit Christophe ▼

COEFFICIENT CONFIANCE : 40

REEMPLACANT : ☒ Coste Evelyne ▼

MOTIF : Rapport annuel ▼

Bilan

Visite Satisfaisante

Eléments présentés

COCHEZ LA CASE : ☒ CHOISIR PRODUIT 1 : BIVALIC ▼

COCHEZ LA CASE : ☒ CHOISIR PRODUIT 2 : EQUILAR ▼

DOCUMENTATION OFFERTE : ☐

Echantillons

PRODUIT 1 : ADIMOL ▼ 20

PRODUIT 2 : AMOPIL ▼ 10

PRODUIT 3 : APATOLUX Vitamine C ▼ 30

PRODUIT 4 : BACTIGEL ▼ 10

PRODUIT 5 : BIVALIC ▼ 15

PRODUIT 6 : CARTION ▼ 5

PRODUIT 7 : CLAZER ▼ 10

PRODUIT 8 : PARMOCODEINE ▼ 15

PRODUIT 9 : TOUXISOL Vitamine C ▼ 4

PRODUIT 10 : LURIREGUL ▼ 20

DATE SAISIE : 2023-11-12

SAISIE DEFINITIVE : ☐


Valider Annuler

6

	Suivi des comptes-rendus de visite
David Andre Délégué régional Ajout compte-rendu Modification compte-rendu Consultation Praticiens Ajout praticien Consultation Médicaments Compte-rendus d'une période Compte-rendus de sa région Déconnexion	Le compte-rendu a été validé

4. Consultation des médicaments

Cas d'utilisation :


	Suivi des comptes-rendus de visite
Louis Villechalane Visiteur Ajout compte-rendu Modification compte-rendu Consultation Praticiens Ajout praticien Consultation Médicaments Compte-rendus d'une période Déconnexion	<div>Consulter les médicaments</div> <div>MEDICAMENT : <input type="text" value="LITHORINE"/></div> <div><input type="button" value="Valider"/> <input type="button" value="Effacer"/></div>

	Suivi des comptes-rendus de visite	
	DEPOT LEGAL :	<input type="text" value="LITHOR12"/>
	NOM COMMERCIAL :	<input type="text" value="LITHORINE"/>
	FAMILLE :	<input type="text" value="Antipsychotique normothymique"/>
	COMPOSITION :	<input type="text" value="Lithium"/>
	EFFETS :	<input type="text" value="Ce médicament est indiqué dans la prévention des psychoses maniaco-dépressives ou pour traiter les états maniaques."/>
	CONTRE INDICATIONS :	<input type="text" value="Ce médicament ne doit pas être utilisé si vous êtes allergique au lithium. Avant de prendre ce traitement, signalez à votre médecin traitant si vous souffrez d'insuffisance rénale, ou si vous avez un régime sans sel."/>

5. Consultation des praticiens

Cas d'utilisation :

Page de choix du praticien :

	Suivi des comptes-rendus de visite	
	Consulter les praticiens	
	PRATICIEN :	<input type="text" value="Chubilleau Louis"/>
	<input type="button" value="Valider"/> <input type="button" value="Effacer"/>	

Louis Villechalane

Visiteur

[Ajout compte-rendu](#)
[Modification compte-rendu](#)
[Consultation Praticiens](#)
[Ajout praticien](#)
[Consultation](#)
[Médicaments](#)
[Compte-rendus d'une période](#)
[Déconnexion](#)

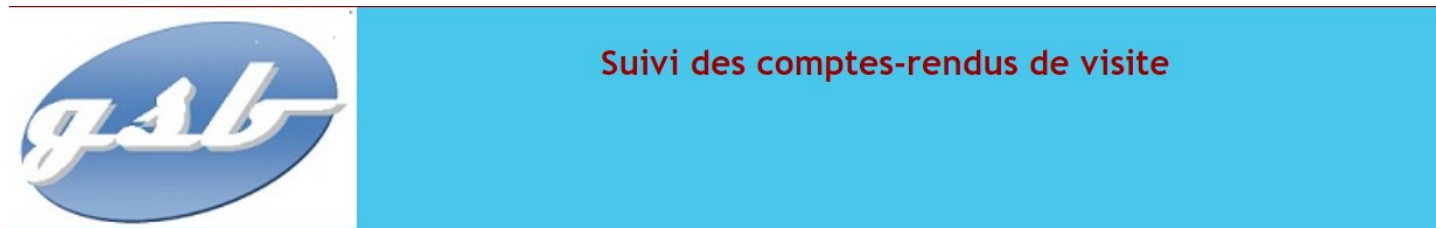
	Suivi des comptes-rendus de visite																				
Louis Villechalane Visiteur Ajout compte-rendu Modification compte-rendu Consultation Praticiens Ajout praticien Consultation Médicaments Compte-rendus d'une période Déconnexion	<table><tr><td>NUMERO :</td><td><input type="text" value="47"/></td></tr><tr><td>NOM :</td><td><input type="text" value="Chubilleau"/></td></tr><tr><td>PRENOM :</td><td><input type="text" value="Louis"/></td></tr><tr><td>ADRESSE :</td><td><input type="text" value="46 r Eglise"/></td></tr><tr><td>CODE POSTAL :</td><td><input type="text" value="17000"/></td></tr><tr><td>VILLE :</td><td><input type="text" value="SAINTES"/></td></tr><tr><td>COEFF. NOTORIETE :</td><td><input type="text" value="202.07"/></td></tr><tr><td>COEFF. CONFIANCE :</td><td><input type="text" value="16"/></td></tr><tr><td>TYPE :</td><td><input type="text" value="Médecine de Ville"/></td></tr><tr><td>LIEU :</td><td><input type="text" value="Cabinet"/></td></tr></table>	NUMERO :	<input type="text" value="47"/>	NOM :	<input type="text" value="Chubilleau"/>	PRENOM :	<input type="text" value="Louis"/>	ADRESSE :	<input type="text" value="46 r Eglise"/>	CODE POSTAL :	<input type="text" value="17000"/>	VILLE :	<input type="text" value="SAINTES"/>	COEFF. NOTORIETE :	<input type="text" value="202.07"/>	COEFF. CONFIANCE :	<input type="text" value="16"/>	TYPE :	<input type="text" value="Médecine de Ville"/>	LIEU :	<input type="text" value="Cabinet"/>
NUMERO :	<input type="text" value="47"/>																				
NOM :	<input type="text" value="Chubilleau"/>																				
PRENOM :	<input type="text" value="Louis"/>																				
ADRESSE :	<input type="text" value="46 r Eglise"/>																				
CODE POSTAL :	<input type="text" value="17000"/>																				
VILLE :	<input type="text" value="SAINTES"/>																				
COEFF. NOTORIETE :	<input type="text" value="202.07"/>																				
COEFF. CONFIANCE :	<input type="text" value="16"/>																				
TYPE :	<input type="text" value="Médecine de Ville"/>																				
LIEU :	<input type="text" value="Cabinet"/>																				

6. Ajout des praticiens

Cas d'utilisation :

Page de saisie :

	Suivi des comptes-rendus de visite																						
Louis Villechalane Visiteur Ajout compte-rendu Modification compte-rendu Consultation Praticiens Ajout praticien Consultation Médicaments Compte-rendus d'une période Déconnexion	<table><tr><td colspan="2">Saisie d'un praticien</td></tr><tr><td>NUMERO :</td><td><input type="text" value="88"/></td></tr><tr><td>NOM :</td><td><input type="text" value="RADA"/></td></tr><tr><td>PRENOM :</td><td><input type="text" value="Elise"/></td></tr><tr><td>ADRESSE :</td><td><input type="text" value="15 place de la capitale"/></td></tr><tr><td>CODE POSTAL :</td><td><input type="text" value="51960"/></td></tr><tr><td>VILLE :</td><td><input type="text" value="Forgeville"/></td></tr><tr><td>COEFF. NOTORIETE :</td><td><input type="text" value="15"/></td></tr><tr><td>COEFF. CONFIANCE :</td><td><input type="text" value="12"/></td></tr><tr><td>TYPE :</td><td><input type="text" value="Pharmacien Officine"/> ▼</td></tr><tr><td colspan="2"><input type="button" value="Valider"/> <input type="button" value="Annuler"/></td></tr></table>	Saisie d'un praticien		NUMERO :	<input type="text" value="88"/>	NOM :	<input type="text" value="RADA"/>	PRENOM :	<input type="text" value="Elise"/>	ADRESSE :	<input type="text" value="15 place de la capitale"/>	CODE POSTAL :	<input type="text" value="51960"/>	VILLE :	<input type="text" value="Forgeville"/>	COEFF. NOTORIETE :	<input type="text" value="15"/>	COEFF. CONFIANCE :	<input type="text" value="12"/>	TYPE :	<input type="text" value="Pharmacien Officine"/> ▼	<input type="button" value="Valider"/> <input type="button" value="Annuler"/>	
Saisie d'un praticien																							
NUMERO :	<input type="text" value="88"/>																						
NOM :	<input type="text" value="RADA"/>																						
PRENOM :	<input type="text" value="Elise"/>																						
ADRESSE :	<input type="text" value="15 place de la capitale"/>																						
CODE POSTAL :	<input type="text" value="51960"/>																						
VILLE :	<input type="text" value="Forgeville"/>																						
COEFF. NOTORIETE :	<input type="text" value="15"/>																						
COEFF. CONFIANCE :	<input type="text" value="12"/>																						
TYPE :	<input type="text" value="Pharmacien Officine"/> ▼																						
<input type="button" value="Valider"/> <input type="button" value="Annuler"/>																							



**Louis
Villechalane**

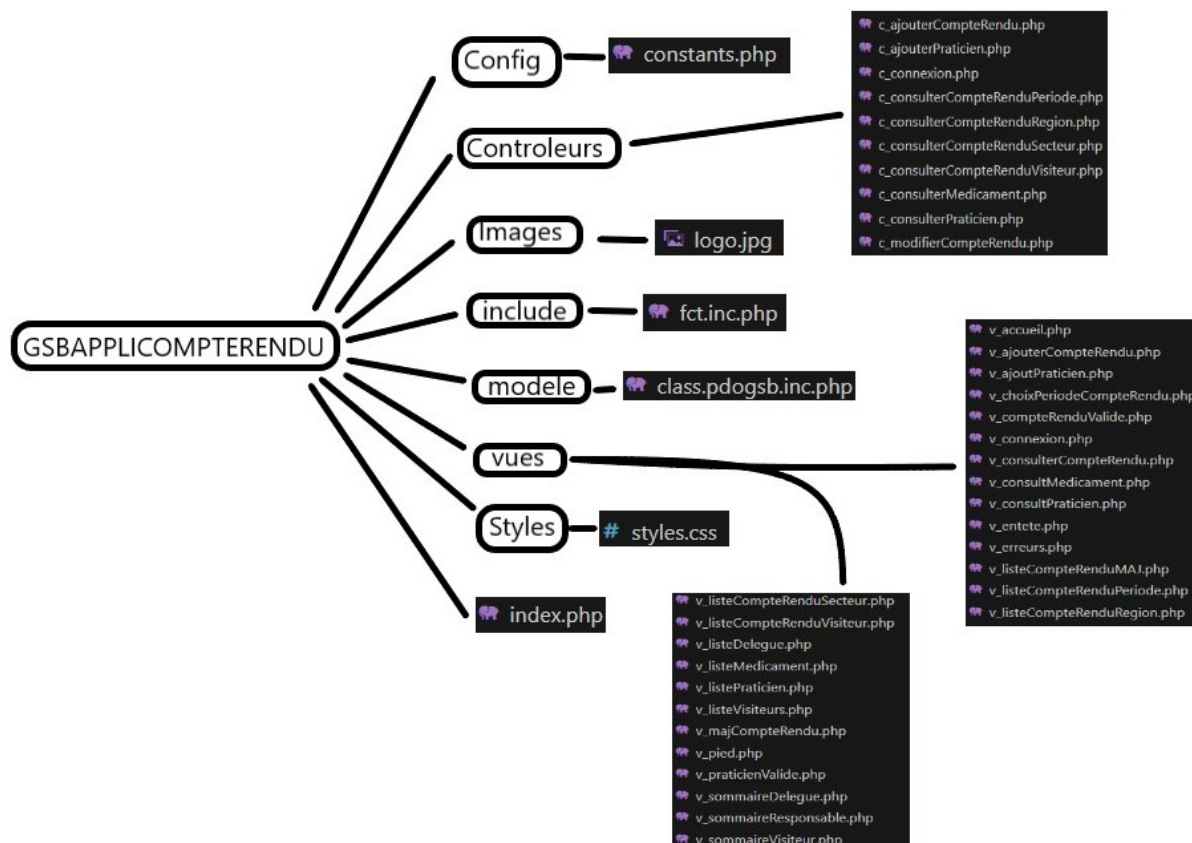
Visiteur

Ajout compte-rendu
Modification compte-rendu
Consultation Praticiens
Ajout praticien
Consultation
Médicaments
Compte-rendus d'une période
Déconnexion

Le praticien a été validé

III. Réalisation technique

A. Architecture de l'application



B. L'accès à la base de données

fichier class.pdogsdb.inc.php :

[<?php](#)

```
/**
 * Classe d'accès aux données.

 * Utilise les services de la classe PDO
 * pour l'application GSB
 * Les attributs sont tous statiques,
 * les 4 premiers pour la connexion
 * $monPdo de type PDO
 * $monPdoGsb qui contiendra l'unique instance de la classe

 * @package default
 * @author Cheri Bibi
 * @version 1.0
 * @link http://www.php.net/manual/fr/book.pdo.php
 */

class PdoGsb{
    private static $serveur=DB_HOST;
    private static $bdd=DB_NAME;
    private static $user=DB_LOGIN;
    private static $mdp=DB_MDP;
    private static $monPdo;
    private static $monPdoGsb=null;
/**
 * Constructeur privé, crée l'instance de PDO qui sera sollicitée
 * pour toutes les méthodes de la classe
 */
    private function __construct(){
        PdoGsb::$monPdo = new PDO(PdoGsb::$serveur.':'.PdoGsb::$bdd, PdoGsb::$user, PdoGsb::$mdp);
        PdoGsb::$monPdo->query("SET CHARACTER SET utf8");
        PdoGsb::$monPdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    }
    public function __destruct(){
        PdoGsb::$monPdo = null;
    }
/**
 * Fonction statique qui crée l'unique instance de la classe

 * Appel : $instancePdoGsb = PdoGsb::getPdoGsb();

 * @return l'unique objet de la classe PdoGsb
 */
    public static function getPdoGsb(){
        if(PdoGsb::$monPdoGsb==null){
            PdoGsb::$monPdoGsb= new PdoGsb();
        }
        return PdoGsb::$monPdoGsb;
    }
}
```

```

/**----- */
/**----- LES TRANSACTIONS ----- */
/**----- */

/**
 * Démarre une transaction
 */
public function démarreTransaction(){
    PdoGsb::$monPdo->beginTransaction();
}

/**
 * Valide une transaction
 */
public function valideTransaction(){
    PdoGsb::$monPdo->commit();
}

/**
 * Annule une transaction
 */
public function annuleTransaction(){
    PdoGsb::$monPdo->rollback();
}

/**----- */
/**----- LES UTILISATEURS ----- */
/**----- */

/**
 * Retourne le matricule d'un collaborateur

 * @param $login
 * @param $mdp
 * @return le matricule sous la forme d'un tableau associatif
 */
public function getIdCollaborateur($login, $mdp){
    $req = "select COLLABORATEUR.COL_MATRICULE as matricule from COLLABORATEUR
    where COLLABORATEUR.COL_LOGIN= :login and COLLABORATEUR.COL_MDP= :mdp";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array( ':login' => $login, ':mdp' => $mdp));
    $ligne = $idJeuRes->fetch();
    return $ligne;
}

/**

 * Retourne les informations d'un collaborateur

 * @param $matricule
 * @return le nom et le prénom sous la forme d'un tableau associatif
 */
public function getInfosCollaborateur($matricule){
    $req = "select COLLABORATEUR.COL_NOM as nom, COLLABORATEUR.COL_PRENOM as prenom, COLLABORATEUR.COL_MATRICULE as
matricule from COLLABORATEUR
    where COLLABORATEUR.COL_MATRICULE = :matricule";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array( ':matricule' => $matricule));
    $ligne = $idJeuRes->fetch();
    return $ligne;
}

```

```

/**
 * Retourne tous les visiteurs (collaborateurs non responsables)
 * @param string $nomTable nom de la table demandée
 * @param string $nomColonne critère de tri
 * @return array
 */
function getLesVisiteurs(){
    $req = "SELECT COL_MATRICULE, COL_NOM, COL_PRENOM FROM collaborateur LEFT JOIN responsable ON COL_MATRICULE =
RES_MATRICULE WHERE RES_MATRICULE IS NULL";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute();
    $lignes = $idJeuRes->fetchAll();
    return $lignes;
}

/**
 * @param $matricule
 * @return un char qui correspond au grade du collaborateur
 */
function getGradeCollaborateur($matricule){
    $req = "select RESPONSABLE.RES_MATRICULE as matricule from RESPONSABLE where RESPONSABLE.RES_MATRICULE
= :matricule";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array( ':matricule' => $matricule));
    $ligne = $idJeuRes->fetch();
    if(isset($ligne[0])){
        return "R";
    }
    else{
        $req = "select DELEGUE_REGIONAL.DEL_MATRICULE as matricule from DELEGUE_REGIONAL where
DELEGUE_REGIONAL.DEL_MATRICULE = :matricule";
        $idJeuRes = PdoGsb::$monPdo->prepare($req);
        $idJeuRes->execute(array( ':matricule' => $matricule));
        $ligne = $idJeuRes->fetch();
        if(isset($ligne[0])){
            return "D";
        }
        else{
            return "V";
        }
    }
}

/**----- */
/**----- LES MEDICAMENTS ----- */
/**----- */

/**
 * @return les Médicaments
 */

function getLesMedicaments(){
    $req = "SELECT MED_DEPOTLEGAL, MED_NOMCOMMERCIAL, FAM_CODE, MED_COMPOSITION, MED_EFFETS, MED_CONTREINDIC FROM
medicament ORDER BY MED_NOMCOMMERCIAL";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute();
    $result = $idJeuRes->fetchAll();
    return $result;
}

```

```

/**
 * @param $id
 * @return médicament de l'identifiant entré
 */
function getUnMedicament($id){
    $req = "SELECT MED_NOMCOMMERCIAL, FAM_CODE, MED_COMPOSITION, MED_EFFETS, MED_CONTREINDIC FROM medicament WHERE
MED_DEPOTLEGAL = :id";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":id" => $id));
    $result = $idJeuRes->fetch();
    return $result;
}

/**
 * @param $famille
 * @return famille du code entré
 */

function getFamilleMedicament($famille){
    $req = "SELECT FAM_LIBELLE FROM famille WHERE FAM_CODE = :code";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":code" => $famille));
    $result = $idJeuRes->fetch();
    return $result[0];
}

/**----- */
/**----- LES PRATICIENS ----- */
/**----- */

/**
 * @return les praticiens
 */

function getLesPraticiens(){
    $req = "select PRATICIEN.PRA_NUM, PRATICIEN.PRA_NOM, PRATICIEN.PRA_PRENOM, PRATICIEN.PRA_ADRESSE,
PRATICIEN.PRA_CP, PRATICIEN.PRA_VILLE, PRATICIEN.PRA_COEFNOTORIETE, PRATICIEN.PRA_COEFCONFIANCE, PRATICIEN.TYP_CODE
from PRATICIEN ORDER BY PRA_NUM";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute();
    $lignes = $idJeuRes->fetchAll();
    return $lignes;
}

/**
 * @param $id
 * @return les praticiens
 */

function getUnPratitien($id){
    $req = "select PRATICIEN.PRA_NOM, PRATICIEN.PRA_PRENOM, PRATICIEN.PRA_ADRESSE, PRATICIEN.PRA_CP,
PRATICIEN.PRA_VILLE, PRATICIEN.PRA_COEFNOTORIETE, PRATICIEN.PRA_COEFCONFIANCE, PRATICIEN.TYP_CODE from PRATICIEN
WHERE PRATICIEN.PRA_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":num" => $id));
    $ligne = $idJeuRes->fetch();
    return $ligne;
}

```

```

/**
 * @param $type
 * @return le type praticien et le lieu du code type en paramètre
 */

function getTypePratitien($type){
    $req = "select TYP_LIBELLE, TYP_LIEU from TYPE_PRATICIEN where TYP_CODE = :type";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":type" => $type));
    $result = $idJeuRes->fetchAll();
    return $result[0];
}

/**
 * @return un numéro libre de praticien
 */

function getMaxNumPraticien(){
    $req = "SELECT MAX(PRA_NUM) FROM PRATICIEN";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute();
    $result = $idJeuRes->fetch();
    return ($result[0] + 1);
}

/**
 * @return les types d'un praticien
 */

function getLesTypesPraticien(){
    $req = "SELECT TYP_CODE, TYP_LIBELLE, TYP_LIEU FROM type_praticien ORDER BY TYP_LIBELLE";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute();
    $result = $idJeuRes->fetchAll();
    return $result;
}

/**
 * @param $num
 * @param $nom
 * @param $prenom
 * @param $adresse
 * @param $cp
 * @param $ville
 * @param $notoriete
 * @param $confiance
 * @param $idType
 */
function ajouterPraticien($num, $nom, $prenom, $adresse, $cp, $ville, $notoriete, $confiance, $idType){
    $req = "INSERT INTO praticien
VALUES(:num, :nom, :prenom, :adresse, :cp, :ville, :notoriete, :confiance, :idtype)";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $param = array(":num" => $num, ":nom" => $nom, ":prenom" => $prenom, ":adresse" => $adresse, ":cp" => $cp,
":ville" => $ville, ":notoriete" => $notoriete, ":confiance" => $confiance, "idtype" => $idType);
    $idJeuRes->execute($param);
}

```



```

/**
 * @param $num
 * @param $coef
 */
function updateConfiance($num, $coef){
    $req = "UPDATE praticien SET PRA_COEFCONFIANCE = :coef WHERE PRA_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $param = array(":num" => $num , ":coef" => $coef);
    $idJeuRes->execute($param);
}

/**----- */
/**----- LES COMPTES-RENDUS ----- */
/**----- */

/**
 * @return un numéro libre de rapport_visite
 */

function getMaxNumCompteRendu(){
    $req = "SELECT MAX(RAP_NUM) FROM rapport_visite";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute();
    $result = $idJeuRes->fetch();
    return ($result[0] + 1);
}

/**
 * @return les motifs de visites
 */

function getLesMotifs(){
    $req = "SELECT MOT_CODE, MOT_LIBELLE FROM motif ORDER BY MOT_LIBELLE";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute();
    $result = $idJeuRes->fetchAll();
    return $result;
}

/**
 * @param $idcolab
 * @param $numRapport
 * @param $numPraticien
 * @param $IDRemplacant
 * @param $dateVisite
 * @param $bilan
 * @param $motifAutre
 * @param $motif
 * @param $doc
 * @param $saisie
 */

function ajouterCompteRendu($idcolab, $numRapport, $numPraticien, $IDRemplacant, $dateVisite, $datesaisie, $bilan,
    $motifAutre, $motif, $doc, $saisie){
    $req = "INSERT INTO rapport_visite
VALUES(:idColab, :numRapport, :numPraticien, :remplacant, :dateVisite, :dateSaisie, :bilan, :motifAutre, :motif, :do
cumentation, :saisieDef)";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $param = array(":idColab" => $idcolab, ":numRapport" => $numRapport, ":numPraticien" => $numPraticien,
":remplacant" => $IDRemplacant, ":dateVisite" => $dateVisite, ":dateSaisie" => $datesaisie, ":bilan" => $bilan,
":motifAutre" => $motifAutre, ":motif" => $motif, ":documentation" => $doc, ":saisieDef" => $saisie);
    $idJeuRes->execute($param);
}

```

```

/**
 * @param $medicament
 * @param $collaborateur
 * @param $rapport
 */
function ajouterPresentation($medicament, $collaborateur, $rapport){
    $req = "INSERT INTO presenter VALUES(:med, :collab, :rapport)";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $param = array(":med" => $medicament, ":collab" => $collaborateur, ":rapport" => $rapport);
    $idJeuRes->execute($param);
}

/**
 * @param $collaborateur
 * @param $rapport
 * @param $medicament
 * @param $qte
 */
function ajouterEchantillon($collaborateur, $rapport, $medicament, $qte){
    $req = "INSERT INTO offrir VALUES(:collab, :rapport, :med, :qte)";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $param = array(":collab" => $collaborateur, ":rapport" => $rapport, ":med" => $medicament, ":qte" => $qte);
    $idJeuRes->execute($param);
}

/**
 * @param $num
 */
function supprimerCompteRendu($num){
    $req = "DELETE FROM rapport_visite WHERE RAP_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $param = array(":num" => $num);
    $idJeuRes->execute($param);
}

/**
 * @return les rapports de visite
 */
function getLesComptesRendus(){
    $req = "SELECT COL_MATRICULE, RAP_NUM, PRA_NUM, PRA_NUM_REPLACANT, RAP_DATE_VISITE, RAP_DATE_SAISIE, RAP_BILAN,
RAP_MOTIF_AUTRE, MOT_CODE, RAP_DOCUMENTATION, RAP_SAISIEDEFINITIVE FROM rapport_visite";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute();
    $result = $idJeuRes->fetchAll();
    return $result;
}

/**
 * @param $num
 * @return le compte rendu du numéro
 */
function getLeCompteRendu($num){
    $req = "SELECT COL_MATRICULE, PRA_NUM, PRA_NUM_REPLACANT, RAP_DATE_VISITE, RAP_DATE_SAISIE, RAP_BILAN,
RAP_MOTIF_AUTRE, MOT_CODE, RAP_DOCUMENTATION, RAP_SAISIEDEFINITIVE FROM rapport_visite WHERE RAP_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":num" => $num));
    $result = $idJeuRes->fetch();
    return $result;
}

```

```

/**
 * @param $num
 * @return le nombre de présentation d'un compte rendu
 */
function getNbPresentation($num){
    $req = "SELECT COUNT(MED_DEPOTLEGAL) FROM presenter WHERE RAP_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":num" => $num));
    $result = $idJeuRes->fetch();
    return $result[0];
}

/**
 * @param $num
 * @return les présentations d'un compte rendu
 */
function getLesPresentationRapport($num){
    $req = "SELECT MED_DEPOTLEGAL, COL_MATRICULE, RAP_NUM FROM presenter WHERE RAP_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":num" => $num));
    $result = $idJeuRes->fetchAll();
    return $result;
}

/**
 * @param $num
 * @return les échantillons d'un compte rendu
 */
function getLesEchantillonsRapport($num){
    $req = "SELECT COL_MATRICULE, MED_DEPOTLEGAL, OFF_QTE FROM offrir where RAP_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":num" => $num));
    $result = $idJeuRes->fetchAll();
    return $result;
}

/**
 * @param $num
 * @param $idCollaborateur
 * @param $IDPraticien
 * @param $IDRemplacant
 * @param $dateVisite
 * @param $dateSaisie
 * @param $bilan
 * @param $autre_motif
 * @param $motif
 * @param $docOffert
 * @param $saisieDef
 */
function majCompteRendu($num, $idCollaborateur, $IDPraticien, $IDRemplacant, $dateVisite, $dateSaisie, $bilan,
    $autre_motif, $motif, $docOffert, $saisieDef){
    $req = "UPDATE rapport_visite SET COL_MATRICULE = :idCollab, PRA_NUM = :idPrat, PRA_NUM_REPLACANT = :idRemp,
    RAP_DATE_VISITE = :dateVisite, RAP_DATE_SAISIE = :dateSaisie, RAP_BILAN = :bilan, RAP_MOTIF_AUTRE = :autreMotif,
    MOT_CODE = :motif, RAP_DOCUMENTATION = :doc, RAP_SAISIEDEFINITIVE = :saisieDef WHERE RAP_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $param = array(":idCollab" => $idCollaborateur, ":idPrat" => $IDPraticien, ":idRemp" => $IDRemplacant,
    ":dateVisite" => $dateVisite, ":dateSaisie" => $dateSaisie, ":bilan" => $bilan, ":autreMotif" => $autre_motif,
    ":motif" => $motif, "doc" => $docOffert, ":saisieDef" => $saisieDef, ":num" => $num);
    $idJeuRes->execute($param);
}

```

```

/**
 * @param $num
 */
function supprimerPresentationRapport($num){
    $req = "DELETE FROM presenter WHERE RAP_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":num" => $num));
}

/**
 * @param $num
 */
function supprimerEchantillonsRapport($num){
    $req = "DELETE FROM offrir WHERE RAP_NUM = :num";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":num" => $num));
}

/**
 * @param $leCode
 * @return le motif du code
 */
function getLeMotif($leCode){
    $req = "SELECT MOT_LIBELLE FROM motif WHERE MOT_CODE = :code";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":code" => $leCode));
    $result = $idJeuRes->fetch();
    return $result;
}

/**
 * @return les comptes rendus de la region
 */
function getLesComptesRendusRegion($delegue){
    $req = "SELECT RAP_NUM, PRA_NUM, PRA_NUM_REPLACANT, RAP_DATE_VISITE, RAP_DATE_SAISIE, RAP_BILAN, RAP_MOTIF_AUTRE, MOT_CODE, RAP_DOCUMENTATION, RAP_SAISIEDEFINITIVE FROM rapport_visite WHERE COL_MATRICULE = :matricule";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":matricule" => $delegue));
    $result = $idJeuRes->fetchAll();
    return $result;
}

/**
 * @param $matricule
 * @return le secteur du responsable
 */
function getSecteurResponsable($matricule){
    $req = "SELECT SEC_CODE FROM responsable WHERE RES_MATRICULE = :matricule";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":matricule" => $matricule));
    $result = $idJeuRes->fetch();
    return $result[0];
}

/**
 * @param $secteur
 * @return les regions du secteur
 */
function getRegionsDuSecteur($secteur){
    $req = "SELECT REG_CODE FROM region WHERE SEC_CODE = :secteur";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":secteur" => $secteur));
    $result = $idJeuRes->fetchAll();
    return $result;
}

```

```

/**
 * @param $region
 * @return le délégué de la région
 */
function getDelegueRegion($region){
    $req = "SELECT DEL_MATRICULE FROM delegue_regional WHERE REG_CODE = :region";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":region" => $region));
    $result = $idJeuRes->fetch();
    return $result;
}

function getLesComptesRendusVisiteur($matricule){
    $req = "SELECT RAP_NUM, PRA_NUM, PRA_NUM_REPLACANT, RAP_DATE_VISITE, RAP_DATE_SAISIE, RAP_BILAN, RAP_MOTIF_AUTRE,
MOT_CODE, RAP_DOCUMENTATION, RAP_SAISIEDEFINITIVE FROM rapport_visite WHERE COL_MATRICULE = :matricule";
    $idJeuRes = PdoGsb::$monPdo->prepare($req);
    $idJeuRes->execute(array(":matricule" => $matricule));
    $result = $idJeuRes->fetchAll();
    return $result;
}

}
?>

```

C. Les fonctions utilitaires

fichier fct.inc.php :

```

<?php
/**
 * Fonctions pour l'application GSB

 * @package default
 * @author Cheri Bibi
 * @version 1.0
 */

/**
 * Teste si un quelconque visiteur est connecté
 * @return vrai ou faux
 */
function estConnecteVisiteur(){
    return isset($_SESSION['idVisiteur']);
}

/**
 * Enregistre dans une variable session les infos d'un visiteur

 * @param $matricule
 * @param $nom
 * @param $prenom
 */
function connecterVisiteur($matricule,$nom,$prenom,$grade){
    $_SESSION['idVisiteur'] = $matricule;
    $_SESSION['nom'] = $nom;
    $_SESSION['prenom'] = $prenom;
    $_SESSION['grade'] = $grade;
}

/**

 * Détruit la session active
 */
function deconnecter(){
    session_destroy();
}

```

```

/**
 * Transforme une date au format français jj/mm/aaaa vers le format anglais aaaa-mm-jj
 * @param $date au format jj/mm/aaaa
 * @return string la date au format anglais aaaa-mm-jj
 */
function dateFrancaisVersAnglais($date){
    @list($jour,$mois,$annee) = explode('/', $date);
    return date("Y-m-d", mktime(0, 0, 0, $mois, $jour, $annee));
}

/**
 * Transforme une date au format format anglais aaaa-mm-jj vers le format
 * français jj/mm/aaaa
 * @param $date au format aaaa-mm-jj
 * @return string la date au format format français jj/mm/aaaa
 */
function dateAnglaisVersFrancais($date){
    @list($annee,$mois,$jour)=explode('-', $date);
    return date("d/m/Y", mktime(0, 0, 0, $mois, $jour, $annee));
}

/* gestion des erreurs*/
/**
 * Indique si une valeur est un entier positif ou nul

 * @param $valeur
 * @return vrai ou faux
 */
function estEntierPositif($valeur) {
    return preg_match("/^[^0-9]/", $valeur) == 0;
}

/**
 * Indique si un tableau de valeurs est constitué d'entiers positifs ou nuls

 * @param $tabEntiers : le tableau
 * @return vrai ou faux
 */
function estTableauEntiers($tabEntiers) {
    $ok = true;
    foreach($tabEntiers as $unEntier){
        if(!estEntierPositif($unEntier)){
            $ok=false;
        }
    }
    return $ok;
}

/**
 * Vérifie si une date est inférieure d'un an à la date actuelle

 * @param $dateTestee
 * @return vrai ou faux
 */
function estDateDepassee($dateTestee){
    $dateActuelle=date("d/m/Y");
    @list($jour,$mois,$annee) = explode('/', $dateActuelle);
    $annee--;
    $AnPasse = $annee.$mois.$jour;
    @list($jourTeste,$moisTeste,$anneeTeste) = explode('/', $dateTestee);
    return ($anneeTeste.$moisTeste.$jourTeste < $AnPasse);
}

```

```

/**
 * Vérifie la validité du format d'une date française jj/mm/aaaa

 * @param $date
 * @return vrai ou faux
 */
function estDateValide($date){
    $tabDate = explode('/', $date);
    $dateOK = true;
    if (count($tabDate) != 3) {
        $dateOK = false;
    }
    else {
        if (!estTableauEntiers($tabDate)) {
            $dateOK = false;
        }
        else {
            if (!checkdate($tabDate[1], $tabDate[0], $tabDate[2])) {
                $dateOK = false;
            }
        }
    }
    return $dateOK;
}

/**
 * Ajoute le libellé d'une erreur au tableau des erreurs

 * @param $msg : le libellé de l'erreur
 */
function ajouterErreur($msg){
    if (!isset($_REQUEST['erreurs'])){
        $_REQUEST['erreurs']=array();
    }
    $_REQUEST['erreurs'][]=$msg;
}

/**
 * Retourne le nombre de lignes du tableau des erreurs

 * @return le nombre d'erreurs
 */
function nbErreurs(){
    if (!isset($_REQUEST['erreurs'])){
        return 0;
    }
    else{
        return count($_REQUEST['erreurs']);
    }
}

?>

```


D. Architecture d'une page

entête:



Suivi des comptes-rendus de visite

sommaire visiteur:

**Louis
Villechalane**

Visiteur

Ajout compte-rendu
Modification compte-
rendu
Consultation Praticiens
Ajout praticien
Consultation
Médicaments
Compte-rendus d'une
période
Déconnexion

sommaire délégué:

David Andre

Délégué régional

Ajout compte-rendu
Modification compte-
rendu
Consultation Praticiens
Ajout praticien
Consultation
Médicaments
Compte-rendus d'une
période
Compte-rendus de sa
région
Déconnexion

sommaire responsable:

Eric Cadic

Responsable

Consulter le compte
rendu d'une période
consulter le compte
Rendu de la Région
Consulte le compte
rendu du visiteur
Consulte le compte
rendu du secteur
Déconnexion

contenu de la page variable selon le traitement à réaliser:

Modifier un compte rendu

Choisir le rapport

E. Les contrôleurs et les vues

fichier index.php :

```
<?php
session_start();
// Définition du chemin de la racine du site
define("APPATH", realpath(dirname("__FILE__")).DIRECTORY_SEPARATOR);
// Inclusion des paramètres de l'application
require_once APPATH."config".DIRECTORY_SEPARATOR."constants.php";
// Inclusion de la classe de gestion de la base de données
require_once (MODELSPATH."class.pdogs.inc.php");
// Inclusion de l'entête et de l'include
include(VIEWSPATH."v_entete.php");
require_once(INCLUDEPATH."fct.inc.php");

$pdo = PdoGsb::getPdoGsb();
$estConnecteVisiteur = estConnecteVisiteur();

if(!isset($_REQUEST['uc']) || !$estConnecteVisiteur)
{
    $_REQUEST['uc'] = 'connexion';
}
$uc = trim(htmlentities($_REQUEST['uc']));
switch($uc){
    case 'connexion':{ // (accessible visiteur, délégué et responsable)
        include(CONTROLLERSPATH."c_connexion.php");break; }
    case 'ajouterCompteRendu' :{ // (accessible visiteur, délégué)
        include(CONTROLLERSPATH."c_ajouterCompteRendu.php");
        break; }
    case 'majCompteRendu' :{ // (accessible visiteur, délégué)
        include(CONTROLLERSPATH."c_modifierCompteRendu.php");
        break; }
    case 'ajouterPraticien' :{ // (accessible visiteur, délégué)
        include(CONTROLLERSPATH."c_ajouterPraticien.php");
        break; }
    case 'consulterPraticien' :{ // (accessible visiteur, délégué)
        include(CONTROLLERSPATH."c_consulterPraticien.php");
        break; }
    case 'consulterMedicament':{ // (accessible visiteur, délégué)
        include(CONTROLLERSPATH."c_consulterMedicament.php");
        break; }
    case 'consulterCompteRenduPeriode' :{ // compte-rendus sur une période (accessible visiteur, délégué et responsable)
        include(CONTROLLERSPATH."c_consulterCompteRenduPeriode.php");
        break; }
    case 'consulterCompteRenduRegion' :{ // compte-rendus d'une région (accessible délégué et responsable)
        include(CONTROLLERSPATH."c_consulterCompteRenduRegion.php");
        break; }
    case 'consulterCompteRenduVisiteur' :{ // compte-rendus d'un visiteur (accessible délégué et responsable)
        include(CONTROLLERSPATH."c_consulterCompteRenduVisiteur.php");
        break; }
    case 'consulterCompteRenduSecteur' :{ // compte-rendus d'un secteur (accessible responsable)
        include(CONTROLLERSPATH."c_consulterCompteRenduSecteur.php");
        break; }
}
// Inclusion du pied
include(VIEWSPATH."v_pied.php");
?>
```

Cette application est composé de 10 contrôleurs différents pour les 10 fonctionnalités:

- c_ajouterCompteRendu.php
- c_ajouterPraticien.php
- c_connexion.php
- c_consulterCompteRenduPeriode.php
- c_consulterCompteRenduRegion.php
- c_consulterCompteRenduSecteur.php
- c_consulterCompteRenduVisiteur.php
- c_consulterMedicament.php
- c_consulterPraticien.php
- c_modifierCompteRendu.php

Le choix du contrôleur se fait par la variable *\$uc* (pour using case) et dans chaque contrôleur ils y a plusieurs actions choisis avec la variable *\$action*.

IV. Procédure d'installation de l'application

Cette application nécessite PHP ainsi que d'être installé sur un serveur web ou sur un serveur WAMP en local.

Mettez l'entièreté des fichiers / dossiers au même endroit dans le serveur, il faut simplement configurer un chemin pour accéder au fichier «index.php» à partir d'un navigateur.