

AFIP-FORMATIONS

Projet personnalisé encadré

Application GSB-Base de données

Sarah Ce-Ougna



AFIP

BTS Services informatiques aux organisations	
Session 2018	
E4 – Conception et maintenance de solutions informatiques	
Coefficient 4	
DESCRIPTION D'UNE SITUATION PROFESSIONNELLE	
Épreuve ponctuelle <input checked="" type="checkbox"/>	Contrôle en cours de formation <input type="checkbox"/>
PARCOURS SISR <input type="checkbox"/>	PARCOURS SLAM <input checked="" type="checkbox"/>
NOM et prénom du candidat : CE-OUGNA Sarah	
N° candidat : 0514962401	
Contexte de la situation professionnelle¹ Contexte « Galaxy-Swiss Bourdin », laboratoire pharmaceutique.	
Intitulé de la situation professionnelle Création d'un site web avec authentification permettant aux visiteurs de présenter aux médecins un catalogue des produits pharmaceutiques proposés par GSB lors de leurs visites médicales.	
Période de réalisation : 19/04/2017 au 30/06/2018	
Lieu : Centre de formation	
Modalité : <input checked="" type="checkbox"/> Seul <input type="checkbox"/> En équipe	
Principale(s) activité(s) concernée(s)² <div style="margin-left: 40px;"> Situations obligatoires : - Production d'une solution logicielle et d'infrastructure - Élaboration de documents relatifs à la production A1.1.1 Analyse du cahier des charges d'un service à produire A1.1.3 Etude des exigences liées à la qualité attendue d'un service A1.2.2 Rédaction des spécifications techniques de la solution retenue (adaptation d'une solution existante ou réalisation d'une nouvelle solution) A1.2.4 Détermination des tests nécessaires à la validation d'un service A1.2.5 Définition des niveaux d'habilitation associés à un service A1.3.4 Déploiement d'un service A3.2.1 Installation et configuration d'éléments d'infrastructure A4.1.2 Conception ou adaptation de l'interface utilisateur d'une solution applicative A4.1.3 Conception ou adaptation d'une base de données A4.1.4 Définition des caractéristiques d'une solution applicative A4.1.6 Gestion d'environnements de développement et de test A4.1.7 Développement, utilisation ou adaptation de composants logiciels A4.1.8 Réalisation des tests nécessaires à la validation d'éléments adaptés ou développés A4.1.9 Rédaction d'une documentation technique A4.2.1 Analyse et correction d'un dysfonctionnement, d'un problème de qualité de service ou de sécurité A4.2.3 Réalisation des tests nécessaires à la mise en production d'éléments mis à jour A4.2.4 Mise à jour d'une documentation technique </div>	

¹ Conformément au référentiel du BTS SIO, le contexte doit être conforme au cahier des charges national en matière d'environnement technologique dans le domaine de spécialité correspondant au parcours du candidat.

<p>Conditions de réalisation² (ressources fournies, résultats attendus)</p> <p>Travail réalisé en centre de formation.</p> <p>Ressources fournies : Documentation concernant GSB.</p> <p>Résultats attendus : La mise en place de ce site web devrait permettre aux chercheurs de publier les informations sur les molécules des médicaments, aux visiteurs de présenter les produits dans leur intégralité et de fournir aux médecins rencontrés une vision exhaustive des composantes médicinales.</p>
<p>Productions associées</p> <p>Base de données et site web, accessibles via un navigateur.</p>
<p>Modalités d'accès aux productions ³</p> <p>Adresse IP du server web : 172.18.0.118 (URL : http://172.18.0.118)</p> <p>Adresse IP de la base de données : 172.18.0.117</p> <p>Identifiants : sarah / azerty</p> <p>Nom de la base de données : PAPPE1</p>
<p>Au verso de cette page, le candidat présente un descriptif détaillé de la situation professionnelle et des productions réalisées sous forme d'un rapport d'activité permettant notamment de mettre en évidence la démarche suivie et les méthodes retenues.</p>

² En référence à la description des activités des processus prévue dans le référentiel de certification.

³ Conformément au référentiel du BTS SIO « Dans tous les cas, les candidats doivent se munir des outils et ressources techniques nécessaires au déroulement de l'épreuve. Ils sont seuls responsables de la disponibilité et de la mise en oeuvre de ces outils et ressources. Les candidats qui n'en sont pas munis sont pénalisés dans les limites prévues par la grille d'aide à l'évaluation proposée par la circulaire nationale d'organisation. ». Il s'agit par exemple des identifiant, mot de passe, URL d'un espace de stockage et de la présentation de l'organisation du stockage.

SOMMAIRE

I- Contexte _____ **p.4**

II- Environnement de développement _____ **p.8**

III- Base de données _____ **p.13**

IV- L'application Web _____ **p.18**

V- Conclusion _____ **p.27**

Annexe : Identifiants _____ **p.28**

I- CONTEXTE

Au sein de GSB

Galaxy-Swiss Bourdin est un laboratoire pharmaceutique, résultat de la fusion récente entre Galaxy, entreprise Américaine, et l'union de trois laboratoires européens Swiss Bourdin.

GSB emploie 480 visiteurs médicaux en France Métropolitaine, et 60 dans les DOM-TOM. Leur rôle est de démarcher auprès des métiers de la santé tels que les médecins, afin de leur présenter les produits créés au sein du service labo-recherche de l'entreprise.

L'objectif d'une visite est aussi d'actualiser et rafraîchir la connaissance des professionnels de santé sur les produits de l'entreprise. Les visiteurs ne font pas de vente, mais leurs interventions ont un impact certain sur la prescription de la pharmacopée du laboratoire.

Ils récupèrent une information directe sur le terrain. Ceci concerne aussi bien le niveau de la confiance qu'inspire le laboratoire que la lisibilité des notices d'utilisation des médicaments ou encore les éventuels problèmes rencontrés lors de leur utilisation, etc.

Pour l'instant, ces informations ne sont actuellement pas systématiquement remontées au siège, ou elles le sont dans des délais jugés trop longs. Le service *rédaction* qui produit les notices souhaite avoir des remontées plus régulières et directes. Ceci permettra également au service *labo-recherche* d'engager des évaluations complémentaires.

Les visiteurs sont donc amenés à beaucoup se déplacer, et possèdent tous du matériel informatique acheté grâce à une indemnité bisannuelle ou une dotation en équipement versée par GSB.

Au niveau informatique, GSB contient une salle de serveurs assurant les fonctions de base du réseau (DHCP, DNS, Annuaire et gestion centralisée des environnements) et les fonctions de communication (Intranet, Messagerie, Agenda partagé, etc.).

On trouve aussi de nombreuses applications métier (base d'information pharmaceutique, serveurs dédiés à la recherche, base de données des produits du laboratoire, base de données des licences d'exploitation pharmaceutique, etc.)

Contexte externe

Novembre 2009 : après plus de 30 ans de présence sur le marché pharmaceutique français, le médicament Médiator produit par le groupe pharmaceutique Servier est retiré du marché suite à de nombreuses apparitions de problèmes aux valves cardiaques chez ses usagers, et ce, dix ans après la première déclaration de pharmacovigilance à l'Afssaps (Agence française de sécurité sanitaire des produits de santé).

Cette affaire a profondément affecté l'industrie pharmaceutique et le métier de visiteur médical : en effet, au cœur du scandale se trouvent les pratiques du groupe Servier et de ses visiteurs, qui auraient respectivement caché les effets secondaires connus de la molécule responsable et poussé les médecins à prescrire le produit par le biais de cadeaux disproportionnés.

De fait, l'affaire Mediator a eu pour conséquence une attaque en justice du groupe Servier, une plus stricte réglementation des pratiques des visiteurs et la baisse de confiance du public et des métiers de la santé envers l'industrie pharmaceutique.

En effet, bien que des codes de bonnes pratiques aient déjà été mis en place, une nouvelle charte éthique de la visite médicale a été rédigée par l'Etat à la suite de l'affaire. De plus, un décret sur la transparence des liens d'intérêts oblige désormais les visiteurs à rendre publics les cadeaux distribués aux professionnels de santé, et les noms de ces derniers.

Le métier de visiteur médical a aussi connu une lourde baisse au niveau des effectifs, avec une chute de 28% entre 2010 et 2015 et plus 9 700 emplois perdus entre 2014 et 2015 seulement.

De plus en plus de médecins refusent à présent d'accepter les visiteurs au sein de

leurs cabinets, et les visites médicales se font souvent collectivement à l'hôpital.

Ainsi, afin de s'aligner avec les nouvelles exigences en matière de sécurité médicamenteuse et de pratiques de la visite médicale, l'entreprise pharmaceutique GSB a opté pour l'adoption d'une politique de transparence.

Projet

Pour répondre à ce besoin, une application web a été mise en place : elle permet tout d'abord aux chercheurs du labo-recherche de publier dans la base de données des produits GSB les informations de pharmacodynamique et pharmacovigilances associées. Ainsi les molécules contenues dans les médicaments et les effets secondaires qui les concernent peuvent être affichés.

Cette plate-forme permet de plus aux visiteurs d'accéder pendant leurs visites médicales à cette même base de données, et d'ainsi présenter aux médecins de façon transparente toutes les informations pour qu'ils puissent faire un choix informé.

De plus, si les médecins acceptent de prescrire le médicament, leur adresse email peut être ajoutée à une liste de participants pour chaque médicament, et en cas d'évolution de la recherche/ de l'étude clinique, un courriel leur est envoyé automatiquement.

La forme de site web a été choisie pour faciliter l'accès aux visiteurs qui sont mobiles, et qui n'ont ainsi besoin que de leur matériel informatique (PC, tablette, ou téléphone, peu importe le fabriquant) et d'une connexion internet. De même, cela permet aux chercheurs d'y accéder depuis leurs postes fixes ou mobiles.

Dans un premier temps, la plate-forme a donc été développée sous forme d'un site Web sécurisé et authentifié. Dans le futur, cela pourra être étendue vers une application mobile pour les visiteurs, une fois que leur matériel aura été uniformisé.

Le site web est hébergé sur un serveur dédié déjà présent au sein des locaux de GSB.

Le site est avant tout un outil pratique, permettant aux différents agents d'avoir un accès rapide aux informations dont ils ont besoin, et de mettre en ligne efficacement les données qu'ils souhaitent communiquer.

Il a été conçu de plus pour être très intuitif, car les visiteurs se déplaçant constamment,

ils ne sont pas nécessairement formés à son utilisation.

Une documentation leur présente néanmoins les fonctionnalités des différents modules et est distribuée aux utilisateurs selon leur profil (visiteur ou chercheur).

Le site a ainsi été développé en PHP, HTML/CSS, et a trois fonctionnalités principales :

Tout d'abord un moteur de recherche dans la base de données, pour trouver et afficher les informations concernant les médicaments. Cette recherche peut être effectuée à partir du nom du médicament, mais aussi à partir des molécules qu'il contient, des symptômes traités et des informations de pharmacovigilance.

Une fonctionnalité de modification des informations des médicaments a été mise en place également, pour que les chercheurs puissent ajouter des informations sur les nouveaux produits et modifier les informations des autres en cas d'évolution des recherches.

Enfin, une méthode permet d'envoyer un courriel automatiquement aux médecins qui prescrivent un médicament lorsque ses informations sont modifiées.

Au niveau des contraintes, le site internet devait être terminé et fonctionnel avant Avril 2018. Il ne doit être accessible qu'aux chercheurs et aux visiteurs. Ils doivent ainsi en premier lieu entrer leur identifiant/mot de passe, afin d'être redirigé vers la page d'accueil correspondant à leur profil.

Comme chaque employé de GSB possédait déjà une adresse de messagerie de la forme *nomUtilisateur@swiss-galaxy.com*, elles ont été utilisées comme identifiants.

Les différents profils utilisateurs ont donc des droits différents : les visiteurs peuvent lire et chercher, les chercheurs peuvent également modifier. Ils ont donc été catégorisés grâce à un champ spécifiant leur statut dans la base de données.

L'application étant avant tout orientée vers la productivité, on a limité tous les éléments graphiques superflus (pas de menu animé, pas de module *flash*).

L'architecture du programme devait aussi respecter les conventions d'usage : soit développement basé sur des bibliothèques de fonctions et architecture MVP.

II- ENVIRONNEMENT DE DEVELOPPEMENT

Le projet étant une application web, il était nécessaire d'installer et de configurer un serveur web. Le système d'exploitation choisi était une machine virtuelle Debian (Linux), protégée par mot de passe, et le serveur Apache2.

Configuration d'Apache

Puis pour installer le serveur Apache sur la machine, la commande suivante a été lancée via le terminal :

```
apt install apache2
```

Le serveur n'ayant pas de moteur d'exécution PHP par défaut, l'étape suivante fut l'installation de php7.0 :

```
apt install php7.0 php7.0-cli php7.0-mbstring php7.0-mcrypt
```

Après avoir autorisé les services d'apache, le serveur était prêt à fonctionner ; voici un extrait de sa configuration complète (obtenue grâce phpinfo()) :

This program makes use of the Zend Scripting Language Engine:
Zend Engine v3.0.0, Copyright (c) 1998-2017 Zend Technologies
with Zend OPcache v7.0.19-1, Copyright (c) 1999-2017, by Zend Technologies
with Xdebug v2.5.0, Copyright (c) 2002-2016, by Derick Rethans

zend®engine

Configuration

apache2handler

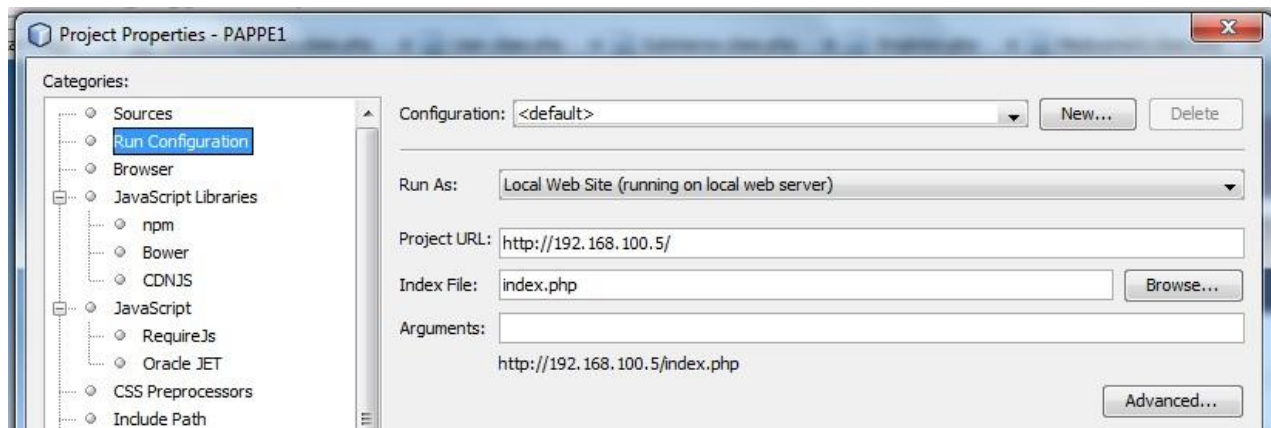
Apache Version	Apache/2.4.25 (Debian)
Apache API Version	20120211
Server Administrator	webmaster@localhost
Hostname:Port	debian9.0:80
User/Group	www-data(33)/33
Max Requests	Per Child: 0 - Keep Alive: on - Max Per Connection: 100
Timeouts	Connection: 300 - Keep-Alive: 5
Virtual Server	Yes
Server Root	/etc/apache2
Loaded Modules	core mod_so mod_watchdog http_core mod_log_config mod_logio mod_version mod_unixd mod_access_compat mod_alias mod_auth_basic mod_auth_core mod_auth_file mod_authz_core mod_authz_host mod_authz_user mod_autoindex mod_deflate mod_dir mod_env mod_filter mod_mime prefork mod_negotiation mod_php7 mod_reqtimeout mod_setenvif mod_status

Directive	Local Value	Master Value
engine	1	1
last_modified	0	0
xbithack	0	0

Fonctionnement avec l'IDE

L'IDE choisie pour ce projet fut NetBeans : il s'agit en effet d'une IDE très complète, et particulièrement efficace pour le développement web en PHP.

Afin de configurer son utilisation connexe au serveur virtuel Apache, les propriétés de l'application (projet sous NetBeans) ont été configurées comme suit, avec 192.18.0.118 pour URL du serveur (192.168.100.5 pour la version test en local) :



Puis, afin d'avoir tous les outils d'assistance à disposition, le debugger XDebug a été ajouté au serveur puis configuré :

```
apt install php-xdebug
```

Dans le fichier 20-xdebug, des commandes ont aussi été ajoutées, afin de préciser l'IP du serveur, l'IDE utilisée et pour permettre la connexion et le débogage à distance. Après ces étapes, le serveur apache et ses outils étaient prêts à fonctionner avec le projet de l'application débuté dans NetBeans.

Sendmail

Un dernier élément a été ajouté par la suite : sendmail. Il s'agit d'un serveur électronique nécessaire pour cette application web désireuse d'envoyer des courriels aux médecins, et son utilisation était définie par défaut dans la configuration du serveur web :

standard

Dynamic Library Support	enabled
Path to sendmail	/usr/sbin/sendmail -t -i

L'installation a été simple et rapide : `apt-get install sendmail`, et aucune configuration n'était requise

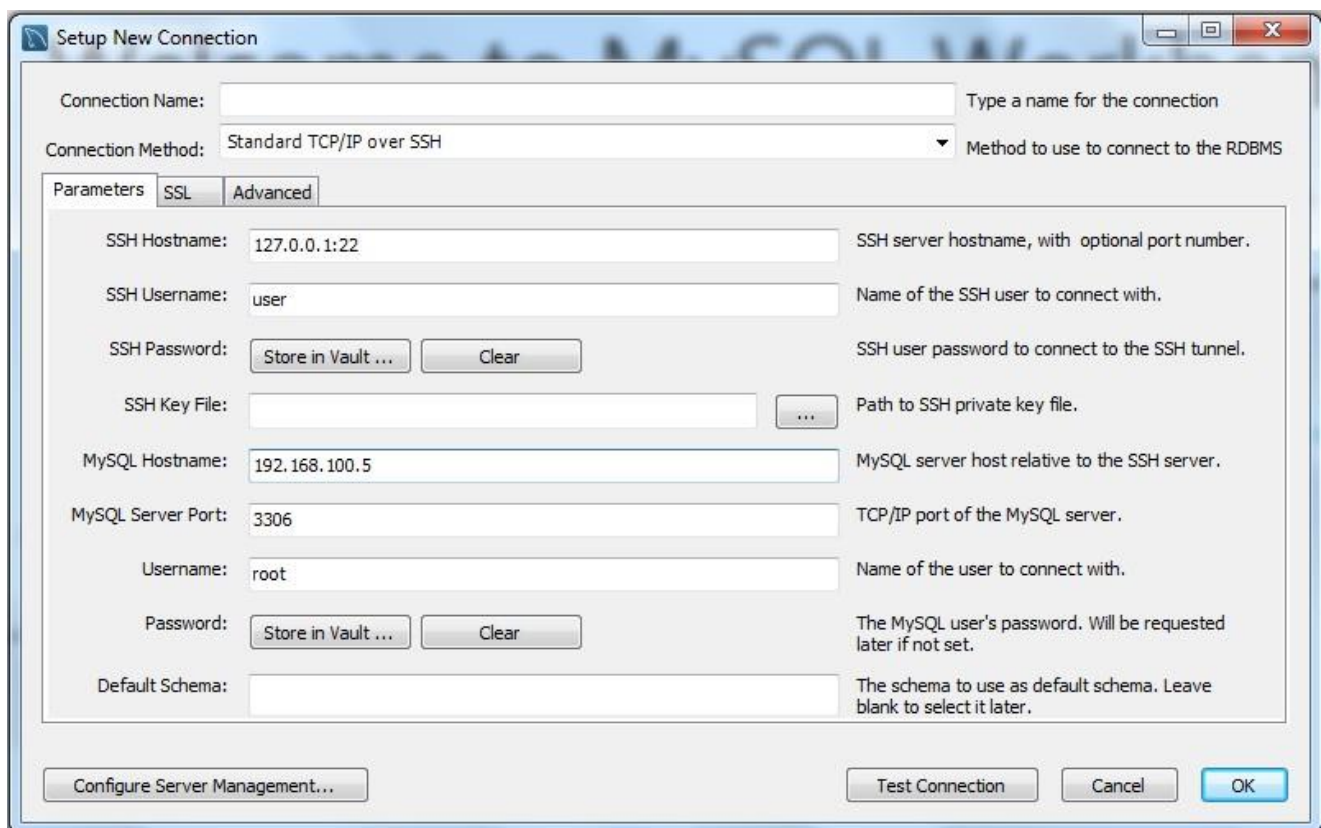
Configuration de la base de données

Après le serveur web, il a fallu créer une nouvelle machine virtuelle (Debian 9 également) et installer mysql-server pour la base de données :

`Apt-get install mysql-server`

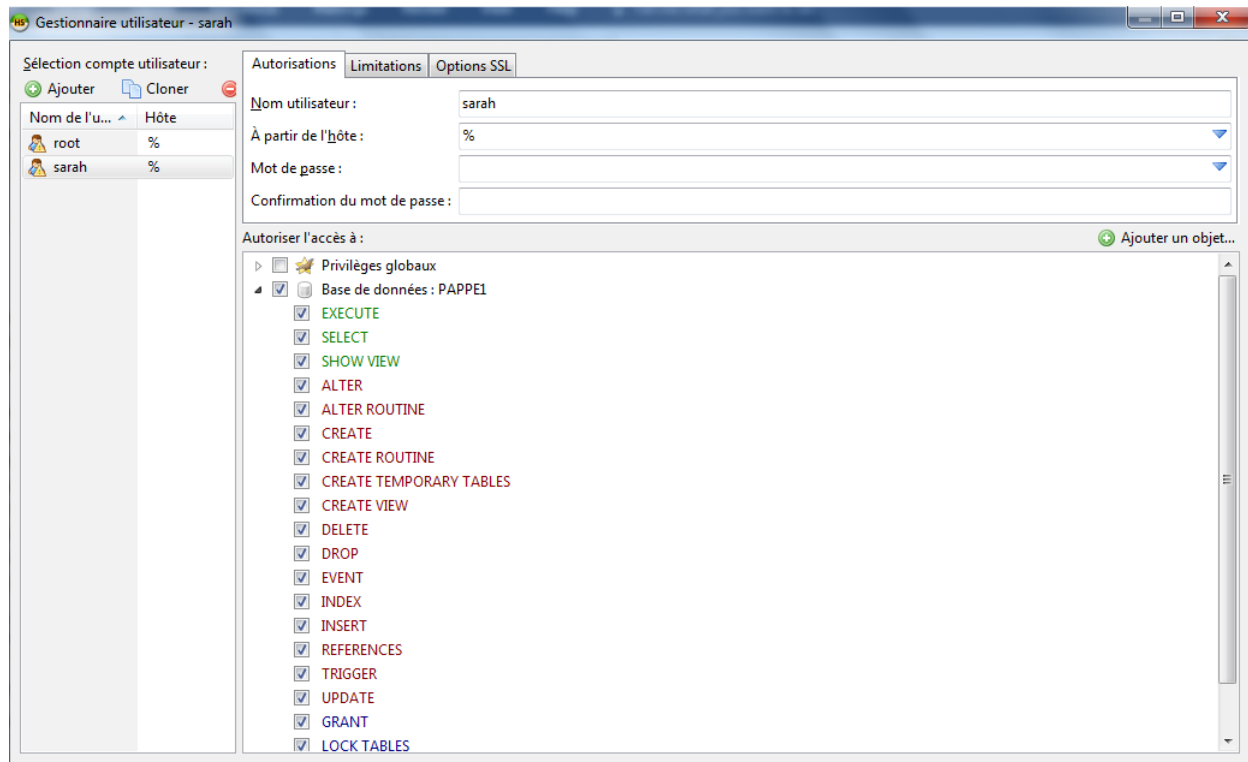
Puis, un compte root avec tous les droits sur la base de données a été créé, via `mysql_secure_installation` et `mysql_setpermission`. Ce compte est lui aussi bien entendu protégé par mot de passe.

Sur MySQL Workbench, la connexion au serveur peut être configurée pour avoir lieu en SSH, afin de maximiser la sécurité des données transmises :



Un autre compte utilisateur de la base de données a aussi été créé via le gestionnaire d'utilisateurs d'HeidiSQL : gérer les utilisateurs ainsi permet de restreindre les accès à la base de données.

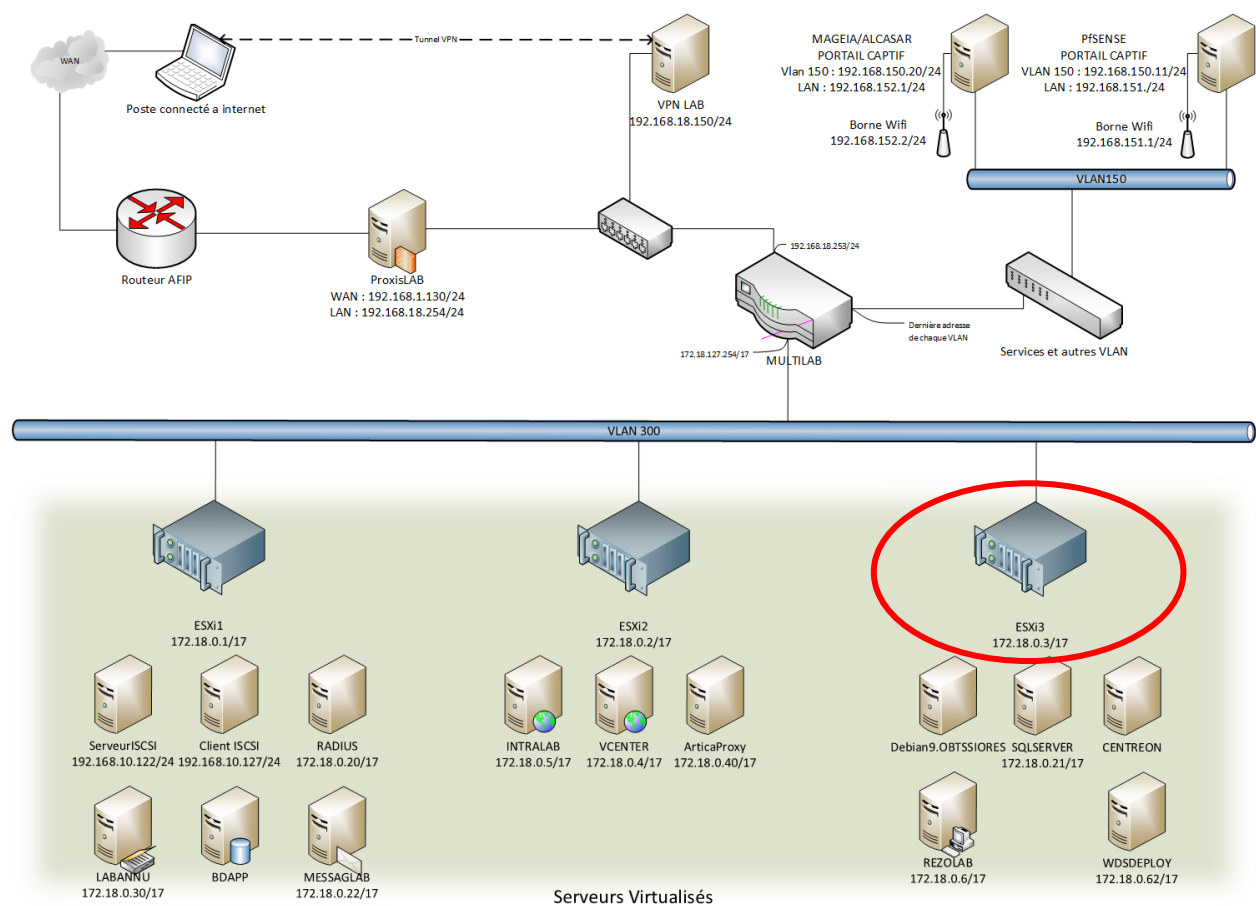
L'utilisateur créé ne possède en effet que les droits définis par root, et uniquement sur les schémas désignés.



Grâce à ces étapes, l'accès à la base de données de GSB est bien restreint et ses données sécurisées.

Infrastructure

Ces deux machines ont ensuite été intégrées à l'infrastructure de GSB, sur le serveur ESX3, d'adresse IP 172.18.0.3/17 :



La machine virtuelle du serveur web a ainsi pris pour adresse IP fixe 172.18.0.118, et la machine portant la base de données 172.18.0.117.

Elles sont toutes deux accessibles depuis partout, à condition d'être connecté(e) au VPN installé pour GSB.

III- BASE DE DONNEES

La base de données mise en place pour cette application a été créée sur MySQL Workbench : elle contient 12 tables au total.

Merise

La création du schéma a été faite grâce à la méthode MERISE. Le contexte de départ était le suivant : l'application devait tout d'abord permettre de restituer les médicaments fabriqués par GSB. Ces médicaments contiennent des informations telles que leur nom, leur prix, leur dosage etc... Chaque médicament est constitué d'une ou plusieurs substances actives, qui sont les composantes à effet des produits.

Les substances ont une Dénomination Commune Internationale (DCI), et elles peuvent causer des effets secondaires. La classification ATC (anatomique, thérapeutique et chimique) permet de regrouper en différentes catégories les produits médicaux, dont les substances actives et les médicaments, selon leurs bienfaits. La classification totale comporte 5 niveaux, allant au plus en plus spécifique, le nombre de sous- groupes se multipliant au fur et à mesure : le premier niveau comporte en effet 14 groupes, tandis que le second en a déjà 94, les chiffres se multipliant presque exponentiellement.

Ainsi, il a été choisi pour cette application de n'utiliser que les deux premiers niveaux de la classification, permettant d'alléger le poids des données. Les substances actives ont été reliées à cette classification, ainsi une substance n'appartient qu'à un sous-groupe de niveau 2. Cela est plus simple que d'utiliser les médicaments qui sont constitués de plusieurs substances et peuvent donc appartenir à plusieurs catégories.

Ces médicaments peuvent toutefois posséder plusieurs voies d'administration (par exemple le doliprane, qui peut être trouvé sous la forme de comprimé ou soluble). Une table listant ces différentes posologies a donc été créée.

Outre les substances actives et les voies d'administration, les médicaments sont aussi reliés aux médecins qui acceptent de les prescrire à partir d'une date précise.

PAPPE1 – MCD

```
graph TD
    ES[Effets_secondaires] --- C1(( ))
    C1 --- SA[Substances_actives]
    SA --- C2(( ))
    C2 --- M[Medicaments]
    M --- C3(( ))
    C3 --- MD[Medecins]
    M --- C4(( ))
    C4 --- VA[Voies_administration]
    M --- C5(( ))
    C5 --- DI([date_inscription])
```

The diagram shows the following entities and their attributes:

- Effets_secondaires**: -code_es, -denomination_es
- Substances_actives**: -code_substance, -DCI (Denomination Commune Internationale), -informations_importantes
- Medicaments**: -code_medicament, -denomination_medicament, -etat_commercialisation, -DAMM (Date d'Autorisation de Mise sur le Marche), -niveau_surveillance, -conditions_prescription, -prix_medicament, -taux_remboursement, -dosages
- Medecins**: -code_medecin, -nom_medecin, -prenom_medecin, -adresse_cabinet, -adresse_mail
- Voies_administration**: -code_va, -denomination_va

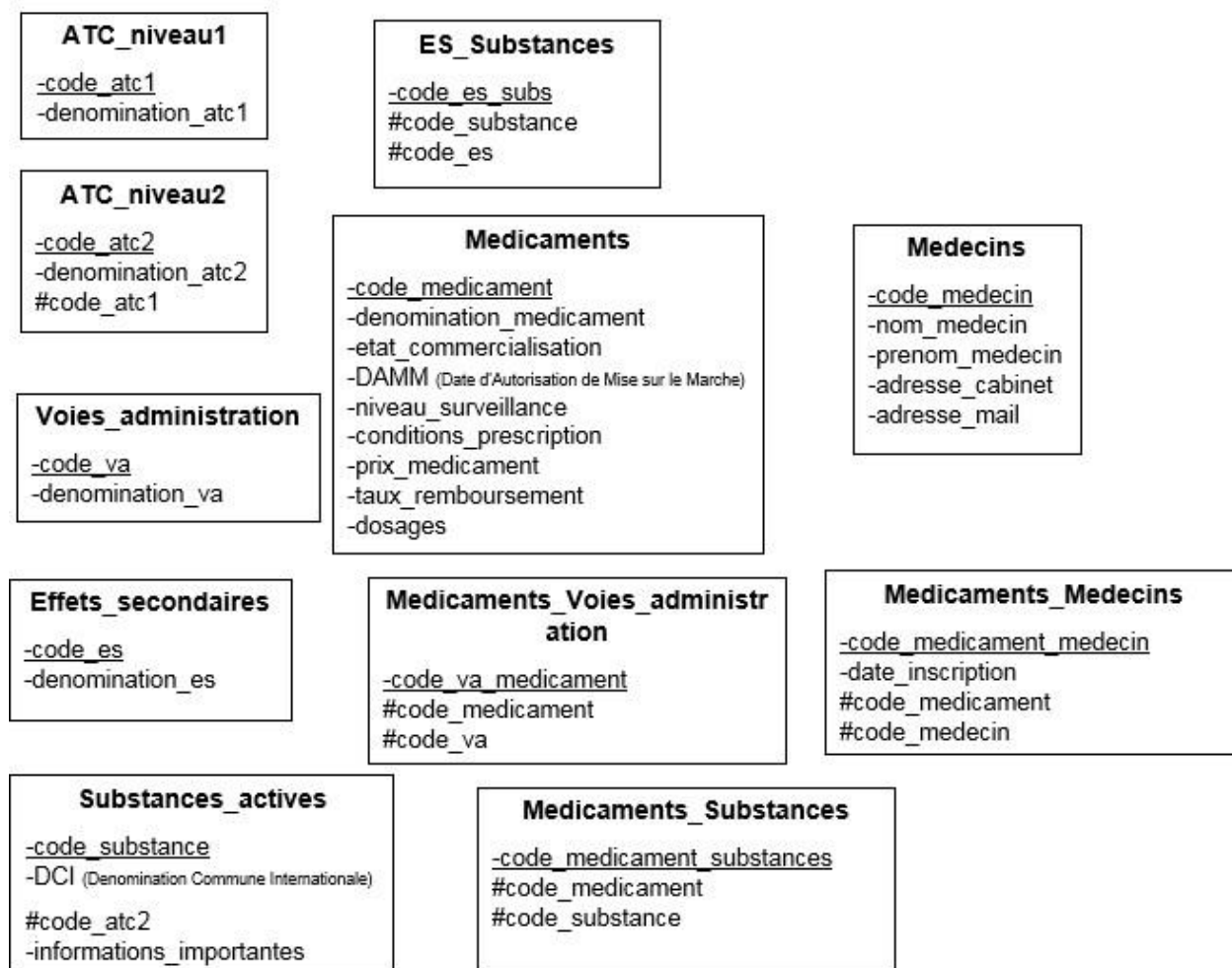
The relationships and their cardinalities are:

- Effets_secondaires to Substances_actives: 1, n
- Substances_actives to Medicaments: 0, n
- Medicaments to Medecins: 1, n
- Medicaments to Voies_administration: 1, n
- Medicaments to date_inscription: 0, n to 1, n

Le modèle comporte ainsi 6 associations, dont 4 n-n, soit 11 tables pour débiter. Il est à noter que la relation médicaments-médecins comporte un attribut : la date d'inscription. Chaque entité possède une clé primaire de la forme 'code_entité', par souci d'uniformité.

14

PAPPE1 – MLD



Ainsi, les tables d'associations Effets secondaires-Substances, Médicaments-Voies d'administration, Médicaments-Substances et Médicaments-Médecins se rajoutent, tandis que l'ATC_niveau2 reçoit la clé étrangère #code_atc1 et les substances actives #code_atc2.

(Afin d'alléger les noms d'attributs, les entités 'Effets Secondaires' et 'Voies d'administration' se retrouvent régulièrement abrégées, respectivement en 'ES' et 'VA'.)

Ces tables ont été remplies de données trouvées à la fois dans les bases de données de médicaments publiques, mais aussi aléatoirement remplies grâce au site web « Mockaroo.com ».

Ainsi, les comptes sont de : 14 atc_niveau1, 94 atc_niveau2, 41 effets secondaires, 11 médecins, 13 médicaments, 29 substances actives et 18 voies d'administration. Les tables Effets secondaires-Substances, Médicaments-Voies d'administration,

Medicaments-Substances et Médicaments-Médecins comportent respectivement 34, 25, 19 et 22 entrées.

Procédures stockées

La base de données contient également de nombreuses procédures stockées : 45 au total. Ce mode de fonctionnement a été choisi car il assure plus de sécurité, au lieu d'écrire directement les requêtes au sein du PHP.

Ainsi, les requêtes émises depuis l'application web font appel à des procédures simples telles que :

UpdateSubstance() : procédure permettant de mettre à jour les données d'une substance

```
-- Création de la procédure UpdateSubstance
DELIMITER $$
DROP PROCEDURE IF EXISTS UpdateSubstance $$

CREATE PROCEDURE UpdateSubstance(codesubs INT(11), nom VARCHAR(100), infos VARCHAR (200), codeatc2 VARCHAR(3))
BEGIN
    UPDATE substances_actives SET DCI=nom, infos_sup = infos, code_atc2 = codeatc2 WHERE code_substance=codesubs;
END $$

DELIMITER ;
```

CreateMedecin() : procédure permettant d'ajouter un nouveau médecin à la base de données

```
-- Création de la procédure CreateMedecin
DELIMITER $$
DROP PROCEDURE IF EXISTS CreateMedecin $$

CREATE PROCEDURE CreateMedecin(nom VARCHAR(45), prenom VARCHAR(45), adresse_cabinet VARCHAR(100), adresse_mail VARCHAR(45))
]BEGIN
    INSERT INTO medecins (`nom_medecin`, `prenom_medecin`, `adresse_cabinet`, `adresse_mail`) VALUES (nom, prenom, adresse_cabinet, adresse_mail);
    SELECT LAST_INSERT_ID() FROM medecins;
-END $$

DELIMITER ;
```

Mais aussi à des procédures plus compliquées telles que les procédures pour la recherche :

- SearchMedicaments() : Cette procédure permet de sélectionner des médicaments selon des critères spécifiés par l'utilisateur (en amont).

```

-- Création de la procédure SearchMedicaments
DELIMITER $$
DROP PROCEDURE IF EXISTS SearchMedicaments $$

CREATE PROCEDURE SearchMedicaments(nom VARCHAR(100), etat ENUM('Commercialisé', 'Non-commercialisé'), dateop ENUM('Inférieur', 'Supérieur', 'Egal'), dateamm YEAR,
surveillance ENUM('Aucune', 'Renforcée'), conditions ENUM('Libre', 'Sur ordonnance'), prixop ENUM('Inférieur', 'Supérieur', 'Egal'), prix DECIMAL(2,0), dos VARCHAR(100),
lim INT, ofs INT)
BEGIN
    SELECT * FROM medicaments
    WHERE (
        (nom IS NULL OR denomination_medicament LIKE CONCAT('%', nom, '%'))
        AND (etat IS NULL OR etat_commercialisation = etat)
        AND (dateamm IS NULL OR CASE dateop WHEN 'Inférieur' THEN YEAR(DAAMM) < dateamm WHEN 'Supérieur' THEN YEAR(DAAMM) > dateamm WHEN 'Egal' THEN YEAR(DAAMM) = dateamm END)
        AND (surveillance IS NULL OR niveau_surveillance = surveillance)
        AND (conditions IS NULL OR conditions_prescription = conditions)
        AND (prix IS NULL OR CASE prixop WHEN 'Inférieur' THEN prix_medicament < prix WHEN 'Supérieur' THEN prix_medicament > prix WHEN 'Egal' THEN prix_medicament = prix END)
        AND (dos IS NULL OR dosages LIKE CONCAT('%', dos, '%'))
    ) LIMIT lim OFFSET ofs;
END $$

DELIMITER ;

```

Imaginons par exemple qu'un médecin veuille savoir combien de médicaments en surveillance renforcée GSB a mis sur le marché depuis 2013.

La procédure reçoit en paramètres un nom (complet ou incomplet), un état de commercialisation, un opérateur de date (<, >, =), une date de mise sur le marché, un état de surveillance etc...

Etant donné que tous ces champs ne sont pas obligatoires dans la recherche, chaque paramètre est inclus dans la condition de recherche, mais avec possibilité d'une valeur nulle (NULL).

Ainsi, la recherche se fait sur tous les attributs du médicament, bien que certains ne soient pas renseignés, et retourne la liste de médicaments correspondants.

Le même type de procédure a été créée pour la recherche de substances actives et de médecins. Il sera vu dans la prochaine partie comment ces procédures sont appelées en amont depuis l'application PHP.

IV- L'APPLICATION WEB

Comme indiqué plus-haut, le développement de l'application web en soi s'est fait en PHP, HTML et CSS, sur NetBeans.

L'application avait été imaginée ainsi par les responsables de GSB : une adresse url est entrée dans un navigateur, une page de connexion s'affiche demandant identifiant et mot de passe. Une fois identifié, l'utilisateur arrive immédiatement sur une page affichant les médicaments produits par GSB, limitant le nombre de manœuvres à accomplir par les visiteurs pour les montrer médecins démarchés. Une barre de recherche est visible, une barre de navigation permet d'accéder aux médecins et aux substances également, et les boutons permettant d'ajouter/modifier/supprimer les données sont bien visibles.

En somme, une application simple d'aspect, mais visuellement bien organisée et facile d'accès.

Singleton

Avant d'aborder la partie plus graphique de l'application, la partie interne sera discutée : pour se connecter à la base de données, le pattern Singleton a été mis en place. Il permet de n'avoir qu'une seule et même instance de connexion pour toutes les données à récupérer. Cela prodigue un avantage au niveau de la sécurité, mais aussi au niveau du temps et du travail à la charge du serveur.

Pour le mettre en place, une classe appelée Singleton a été créée : son constructeur est privé, si bien qu'on ne peut s'y connecter en dehors de la classe même. La fonction getConnection() vérifie si une instance du Singleton existe déjà, et si non, en crée une. La connexion que retourne le Singleton est ensuite utilisée par les autres classes qui interagissent avec la base de données.

```

/**
 * Constructeur utilisant le PDO de PHP pour se connecter à une base de données
 */
private function __construct() {
    self::$cnx = new PDO($this->dsn, $this->username, $this->password, $this->options);
}

/**
 * Fonction permettant de vérifier si une connexion existe déjà, et si non d'en instancier une
 * @return type
 */
static function getConnection() {
    if (is_null(self::$cnx)) {
        self::$instance = new Singleton();
    }
    return self::$cnx;
}

```

DAO

Parmi ces classes qui utilisent le Singleton se trouvent les DAO créés pour cette application : étant donné l'importance des tables médicaments, substances_actives et médecins, ces trois entités ont chacune leur propre DAO.

Il y a ainsi un DAOMedicament, un DAOSubstance et un DAOMedecin. C'est dans ces fichiers que les procédures de la base vues précédemment sont appelées.

Voici un exemple d'une fonction du DAOMedecin :

```

/**
 * Fonction permettant de créer un nouveau médecin
 * @param type $nom
 * @param type $prenom
 * @param type $adresse_cabinet
 * @param type $adresse_mail
 * @return type
 */
function getCreateMedecin($nom, $prenom, $adresse_cabinet, $adresse_mail) {
    $SQL = "CALL CreateMedecin(?, ?, ?, ?)";
    $statement = $this->cnx->prepare($SQL);
    $statement->bindParam(1, $nom);
    $statement->bindParam(2, $prenom);
    $statement->bindParam(3, $adresse_cabinet);
    $statement->bindParam(4, $adresse_mail);
    $statement->execute();

    $last_insertid = $statement->fetchColumn();
    return $last_insertid;
}

```

Il s'agit de la fonction getCreateMedecin(), qui appelle la procédure existante

CreateMedecin(), en lui passant les arguments nécessaires.

De plus, pour optimiser le mappage avec la base de données, des classes correspondant aux tables ont été créées : il y a ainsi Medecin.class.php, Substance.class.php, Medicament.class.php, ATC.class.php, EffetSecondaire.class.php, VoieAdministration.class.php, etc...

Cela permet de récupérer des objets et de restituer plus facilement leurs attributs sur les pages de présentation.

Présentateurs

Pour correspondre au modèle de développement de PHP : le patron MVP, trois présentateurs ont été créés : PresenteurMedicaments, PresenteurMedecins et PresenteurSubstances.

Ils jouent le rôle d'intermédiaires entre les DAO et la page d'accueil index.php. Ainsi, tous les présentateurs ont ici des fonctions assez similaires, orientés vers des objets différents. Voici un exemple du PresenteurSubstances, qui fait appel à une fonction du DAOSubstance :

```
/**
 * Fonction retournant les médicaments associés à une substance
 * @return type
 */
function prepareListeMédicaments() {
    $listemeds = $this->daosubs->getMédicamentsBySubstance($this->code_substance, 10, 10 * ($this->pageCourante - 1));
    return $listemeds;
}
```

Cette fonction permet de récupérer la liste des médicaments qui contiennent une certaine substance. Cette fonction est par la suite appelée sur la page d'index.

Index

La page fait ici office de page d'accueil : une fois l'utilisateur connecté, il est redirigé vers cette page, qui fait appel aux présentateurs pour savoir ce qu'il doit afficher.

Une fois que le présentateur a préparé les données, la page d'index donne accès aux 3 onglets présentant les données des trois grandes entités ici utilisées :



Médicaments +

Code	Dénomination	Etat	Mise sur le marché	Niveau de surveillance	Conditions de prescription	Prix	Modifier	Supprimer
49	ATRIPLA	Commercialisé	1997-01-01	Aucune	Libre	10€	—	□
50	BARACLUDE	Commercialisé	2002-01-01	Renforcée	Libre	25€	—	□
51	DAKLINZA	Commercialisé	2004-01-01	Aucune	Sur ordonnance	10€	—	□
52	ELIQUIS	Non-commercialisé	1978-01-01	Renforcée	Sur ordonnance	6€	—	□
53	NULOJIX	Commercialisé	1998-01-01	Aucune	Sur ordonnance	10€	—	□
54	OPDIVO	Commercialisé	2002-01-01	Renforcée	Sur ordonnance	10€	—	□
55	ORENCIA	Non-commercialisé	2003-01-01	Aucune	Libre	8€	—	□
56	PERFALGAN	Commercialisé	1987-01-01	Aucune	Libre	3€	—	□
57	REYATAZ	Commercialisé	2001-01-01	Renforcée	Sur ordonnance	12€	—	□
58	SPRYCEL	Commercialisé	2002-01-01	Aucune	Libre	10€	—	□

Recherche avancée

Nom contenant :

Etat de commercialisation :

Date de Mise sur le Marché:

à : mm/dd/yyyy

Niveau de surveillance :

Prescription :

Prix : à :

Dosages :

Rechercher

Précédente 1 2 Suivante

Substances +

Code	Dénomination	Informations importantes	Modifier	Supprimer
30	clozapine		—	□
31	ropirinoles		—	□
32	méthadone		—	□
33	théophylline		—	□
34	caféine		—	□
35	paclitaxel		—	□
36	répaglinide		—	□
37	antivitamines K		—	□
38	phénytoïne		—	□
39	diazépam		—	□

Recherche avancée

Nom contenant:

Classification ATC (Niveau 1):

Classification ATC (Niveau 2):

Rechercher

Précédente 1 2 3 Suivante

Médecins +

Code	Nom	Prénom	Adresse du cabinet	Adresse email	Modifier	Supprimer
11	Lambdean	Christophe	5 rue de la République 69007 LYON	clambdean0@usnews.com	—	□
12	Mc Gee	Cello	5551 Rowland Park	cmcgee1@twitpic.com	—	□
13	Loving	Hollyanne	101 Veith Street	hloving2@mashable.com	—	□
14	O'Gavin	Jack	0 Kenwood Way	jogavin3@dell.com	—	□
15	Rittmeyer	Ros	6906 Schmedeman Road	rrittmeyer4@pcworld.com	—	□
16	Randles	Reginauld	87494 Sutherland Hill	rrandles5@diggs.com	—	□
17	Possek	Janessa	635 Fulton Street	jpossek6@netscape.com	—	□
18	Truwert	Danyette	1283 Walton Hill	dtruwert7@europa.eu	—	□
19	Hinken	Bridget	769 Vermont Road	bhinken8@mozilla.org	—	□
20	Brumen	Norbert	0 Nobel Crossing	nbrumen9@infoseek.co.jp	—	□

Recherche avancée

Nom contenant:

Prénom contenant:

Adresse (du cabinet) contenant:

Rechercher

Précédente 1 2 Suivante

Pour faire fonctionner la recherche, les barres de recherche situées à droite renvoient les données des champs remplis (et non-remplis en tant que : NULL) via \$_POST directement à cette même page d'index.

Les présentateurs récupèrent et filtrent ces données, les envoient à la procédure de recherche vu préalablement (par le biais des DAO) et ramènent les données correspondantes à l'index qui les affiche.

Pour que le présentateur puisse traiter toutes ces données en une fois, elles sont converties sous la forme d'une chaîne de valeurs séparées par des « : » : ex : DUBOIS:NULL :LYON, pour la recherche d'un docteur Dubois à Lyon.

La fonction formattant ces données se trouve dans les présentateurs :

```
function formatRechercheDoc() {
    if (isset($_POST['nom_medecin']) && isset($_POST['prenom_medecin']) && isset($_POST['adresse'])) {
        $nom = htmlspecialchars($_POST['nom_medecin']);
        if (empty($nom)) {
            $nom = gettype(NULL);
        }
        $prenom = htmlspecialchars($_POST['prenom_medecin']);
        if (empty($prenom)) {
            $prenom = gettype(NULL);
        }
        $adresse = htmlspecialchars($_POST['adresse']);
        if (empty($adresse)) {
            $adresse = gettype(NULL);
        }
        $this->recherche = $nom . ':' . $prenom . ':' . $adresse;
    } else {
        $this->recherche = "";
    }
}
```

En cliquant sur une ligne de données, on arrive sur leur page de détail :

 Accueil - Médicaments Substances actives Medecins Connecté(e): Sarah Ce-Ougna Se déconnecter

Modifier

Code : 11

Nom : Lambdean

Prénom : Christophe

Adresse du cabinet : 5 rue de la République 69007 LY


Adresse email : clambdean0@usnews.com

Médicaments associés

Code	Dénomination	Etat	Mise sur le marché	Niveau de surveillance	Conditions de prescription	Prix	Date de souscription	Modifier	Supprimer
49	ATRIPLA	Commercialisé	1997-01-01	Aucune	Libre	10	2018-01-15	✎	✕
51	DAKLINZA	Commercialisé	2004-01-01	Aucune	Sur ordonnance	10	2018-01-15	✎	✕
55	ORENCIA	Non-commercialisé	2003-01-01	Aucune	Libre	8	2018-01-15	✎	✕

CRUD

Sur la page d'index se trouvent les liens pour modifier les données : à côté de chaque élément, un « + » qui conduit vers le formulaire de création d'un nouvel élément :

 Accueil - Médicaments Substances actives Medecins Connecté(e): Sarah Ce-Ougna Se déconnecter

Nouveau médecin :

Nom :

Prénom :

Adresse du cabinet :

Adresse e-mail :

Medicaments souscrits:

☐ ATRIPLA

☐ BARACLUDE

☐ DAKLINZA

☐ ELIQUIS

☐ NULOJIX

☐ OPDIVO

☐ ORENCIA

☐ PERFALGAN

☐ REYATAZ

☐ SPRYCEL

☐ SUSTIVA

☐ YERVOY

☐ SADOPINE

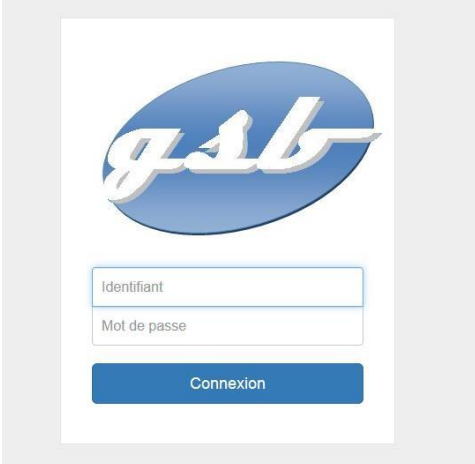
Ajouter

Réinitialiser le formulaire

Il s'agit du même type de formulaire pour les modifications (avec des données préremplies), et une simple demande de confirmation apparaît pour les suppressions. Les formulaires sont envoyés par méthode \$_POST vers une page de confirmation.

SESSIONS

Pour se connecter, l'utilisateur est redirigé vers une page de login :

The image shows a login form for a system with the logo 'gsb'. The logo is a blue oval with the letters 'gsb' in white, stylized, italicized font. Below the logo are two input fields: the first is labeled 'Identifiant' and the second is labeled 'Mot de passe'. Below these fields is a blue button with the text 'Connexion' in white.

Une fois les identifiants entrés, la page redirige vers le fichier check.php, qui vérifie si les identifiants sont corrects. Pour cela, le fichier DAOUser a été créé, ainsi qu'une table dans la base de données : users.

Cette table stocke les identifiants et mots de passe pour vérifier l'identité de ceux qui tentent de se connecter. Pour plus de sécurité, les mots de passe se voient d'abord accolés d'une variable \$salt, les rendant plus difficiles à découvrir.

Après cela, ils sont aussi encryptés grâce à la fonction SHA512 de MySQL.

Les fonctions stockées dans le DAOUser ne sont pas utilisées directement mais plutôt par l'intermédiaire du fichier AccessManager, qui agit en quelque sorte comme un présentateur.

Ces fonctions permettent non seulement d'accorder la connexion ou non, mais aussi de vérifier le statut de l'utilisateur : Administrateur, Chercheur ou bien Visiteur.

```

function CheckUser($login, $password) {
    $SQL = "SELECT * FROM users WHERE login=:log AND pass=SHA2 (:pass,512)";
    $statement = $this->cnx->prepare($SQL);
    $statement->bindParam("log", $login);
    $p = $password . $this->salt;
    $statement->bindParam("pass", $p);
    $statement->execute();
    $u = $statement->fetchObject("User");

    if ($u == false) {
        return false;
    } else {
        session_start();
        $_SESSION['auth'] = $u->categorie;
        $_SESSION["name"] = $u->prenom . " " . $u->nom;
        return true;
    }
}

/**
 * Fonction vérifiant si les droits de la session sont ceux d'un admin
 * @return boolean
 */
function IsAdministrateur() {
    if (isset($_SESSION["auth"])) {
        if ($_SESSION["auth"] == 'Administrateur') {
            return true;
        }
    }
    return false;
}

```

En effet comme mentionné plus haut, les visiteurs et chercheurs n'ont pas les mêmes fonctionnalités et donc pas les mêmes droits : selon leur catégorie et donc selon ce qui est stocké dans la variable de `$_SESSION`, les pages s'afficheront différemment selon la catégorie.

Les chercheurs peuvent ajouter, modifier et supprimer des médicaments et substances, mais ne peuvent que lire les données des médecins ; alors que le contraire est vrai pour les visiteurs. Si un visiteur tente malgré tout de modifier un médicament par exemple, il sera redirigé vers la page NoAccess qui affiche un message d'erreur puis redirige vers la page index.

La catégorie « Administrateur » possède bien sûr tous les droits.

CSS

Pour ce qui est de la partie graphique en CSS, les fonctionnalités de bootstrap ont été ajoutées au code et utilisées notamment pour les tableaux, boutons, formulaires, barre de navigation, etc...

Versioning

Ce projet a de plus été versionné à chaque étape et poussé vers un dépôt BitBucket. En effet, l'extension Git a été ajoutée à l'IDE, et une branche « dev » a été créée

Ainsi, après ajout de chaque fonctionnalité, un commit a été effectué, permettant de sauvegarder l'avancement du projet progressivement.

Une fois le développement terminé, la branche « dev » a été fusionnée à la branche principale « master », et le tout a été poussé sur le compte BitBucket configuré.



The screenshot shows the BitBucket interface for the repository 'PAPPE1'. The left sidebar contains navigation links: Overview, Source, Commits (selected), Branches, Pull requests, Pipelines, Downloads, Boards, and Settings. The main content area is titled 'Commits' and shows a list of four commits by 'UTILISATEUR' from 2018-01-05. A search bar 'Find commits' is in the top right. A vertical commit history graph is on the left of the table. Navigation links 'Prev' and 'Next' are at the bottom of the table.

Author	Commit	Message	Date	Builds
UTILISATEUR	e96896b	Ajout de la page details_substances	2018-01-05	
UTILISATEUR	1ce38ca	Ajout de la page details_medicaments	2018-01-05	
UTILISATEUR	eac6bcf	Ajout de la barre de navigation et des tables de la page index	2018-01-05	
UTILISATEUR	b92d18d	Création de la page login	2018-01-05	

V- CONCLUSION

Ce site web permet bien à GSB d'afficher une politique de transparence envers les médecins qui sont démarchés par les visiteurs. Ces derniers ont un support utile pour les accompagner dans leurs présentations, et l'application permet une communication plus directe entre le secteur recherche, leurs produits et les « clients ».

Ce site devrait donc à long terme améliorer la confiance que les médecins accordent à GSB.

Les développeurs du site travaillent actuellement sur le développement d'une table d'archive des produits de GSB, afin d'avoir une meilleure traçabilité des produits qui ont été commercialisés, mais ne le sont plus.

Annexe 1 : Identifiants

<u>Identifiant (adresse email)</u>	<u>Mot de passe</u>	<u>Catégorie</u>
admin	123+aze	Administrateur
paul.dubois@swiss-galaxy.com	DPpass1965	Chercheur
vincent.thomas@swiss-galaxy.com	Foot12Azerty	Visiteur