

RAPPORT DE STAGE

Centre Hospitalier de BEZIERS



Michèle BOIS-PLAISANCE

Concepteur Développeur Informatique

SOMMAIRE

Table des matières

Chapitre 1 - Présentation.....	5
1.1 - Avant-propos.....	6
1.2 - Abstract.....	7
1.3 - Remerciements.....	8
1.4 - La démarche générale.....	9
Chapitre 2- Cahier des charges.....	10
2.1 - Présentation.....	11
2.2 - Technologies imposées :.....	12
2.3 - Synoptique fonctionnel :.....	13
Chapitre 3- Analyse.....	14
3.1- Les Diagrammes de Cas d'Utilisation.....	15
3.1.1 - Représentation graphique.....	15
3.1.2 - DCU Back-Office.....	15
3.1.3 - DCU Front-Office.....	16
3.1.4- Fiche de description textuelle d'un cas d'utilisation.....	17
3.1.5- Les diagrammes de Séquence Système.....	18
3.1.6- Diagramme d'activité.....	20
3.1.6.1 – FRONT-OFFICE.....	20
3.1.6.2 – BACK OFFICE.....	21
3.2- Les maquettes.....	22
3.3- Les diagrammes de navigation.....	29
3.3.1 - Diagramme de Navigation du Back-Office :.....	29
3.3.2 - Diagramme de Navigation du Front-End :.....	30
Chapitre 4- Conception de la base de données.....	31
4.1- La démarche utilisée.....	32
4.2- Le MCD (Merise).....	33
4.3- Le MLD (Modèle logique des données).....	34
4.4- Le schéma de la Base de Données.....	35
4.5- Quelques exemples de code SQL.....	36
4.5.1 - Création de la Base de données : `annuaire_medecins`.....	36
4.5.2 - Création des Procédures Stockées : `annuaire_medecins`.....	37
4.6 - Le diagramme de classes du domaine.....	39
Chapitre 5- Conception de l'application.....	40
5.1- Les diagrammes de séquence détaillés.....	41
5.1.1 - Diagramme de Séquence Détaillé : Authentification avec AJAX.....	41
5.1.2 - Diagramme de Séquence Détaillé : Création d'un compte avec AJAX.....	42
5.2- Les diagrammes de classes participantes.....	43
Chapitre 6- Développement.....	44
6.1 - Les catégories de code.....	45
6.1.1 - Codes HTML-PHP.....	45
6.1.2 - Codes CSS.....	45

6.1.3 - Codes JavaScript.....	45
6.1.4 - Codes AJAX.....	45
6.1.5 - Codes PHP.....	45
6.2 - Structure d'une page PHP : le principe des inclusions.....	46
6.3 - Les codes.....	48
6.3.1 - Exemple de code HTML-PHP.....	48
6.3.2 - Exemple de code CSS.....	50
6.3.3 - Exemple de code JavaScript.....	51
6.3.4 - Exemple de code AJAX.....	52
6.3.5 - Exemples de code PHP.....	59
6.3.5.1 - Boundary.....	59
6.3.5.2 - Contrôleurs.....	61
6.3.5.3 - Entity.....	65
6.3.5.4 - Connexion.....	66
6.3.5.5 - Properties.....	68
6.3.5.6 - DAO.....	69
Chapitre 7- Déploiement.....	73
7.1- Le diagramme de déploiement.....	74
7.2- Le déploiement.....	75
Chapitre 8 - Formation des utilisateurs.....	76
Chapitre 9- Conclusion.....	77
Chapitre 10- Annexes.....	78
10.1- Correspondances Projet / REAC.....	79
10.2- Outils utilisés.....	81
10.3- Webographie et Bibliographie.....	82
10.4- Glossaire/Lexique ou/et Liste de mots-clés et sigles.....	83
10.5- Code de création de la BD.....	84
10.6- Code des procédures stockées.....	87

Chapitre 1 - *PRÉSENTATION*

«La plus grande attention doit être portée à la compréhension du problème,
faute de quoi l'algorithme n'a aucune chance d'être correct»

Denis LAPOIRE

«C'est toujours l'impatience de gagner qui fait perdre»

Louis XIV (cité dans "L'immortel" de FOG)

«J'écoute et j'oublie. Je lis et je retiens. Je fais et j'apprends»

Proverbe chinois

1.1 - AVANT-PROPOS

L'ENTREPRISE

Le Centre Hospitalier de BEZIERS est l'établissement public de santé de référence du Bassin de l'Ouest Hérault et le 4ème de la région Languedoc-Roussillon par sa taille et son activité. Il assure des missions de soins, d'enseignement et de formation des futurs médecins et personnels soignants.

Chiffres activités 2013 :

- **Hospitalisations** : 31 133 entrées totales (+ 3%)
- **Hôpital de jour** Médecine : 4 924 entrées (+ 4%)
- **Hôpital de semaine** médico-chirurgical : 611 entrées (+ 198%)
- **Chirurgie ambulatoire** : 2 391 entrées (- 1%)
- **Urgences** : 56 159 passages (+ 1%)
- **Nombre de naissances** : 1 449 (+ 4%)
- **Bloc opératoire** : 11 476 interventions (+ 2%)
- **Consultations externes** :
 - 92 429** en Médecine Chirurgie Obstétrique - MCO (+ 4%)
 - 9 829** à l'Unité de Consultation et de Soins Ambulatoires - UCSA (0%)
 - 18 859** en Psychiatrie (+ 2%)

Effectifs : 2 699 agents (médical et non médical)

LE STAGIAIRE

Il y a quelques années de cela, j'ai travaillé, en tant que vendeuse et formatrice-relais chez Darty, dans le rayon Informatique et Téléphonie, pendant 8 ans où j'ai découvert l'univers des PC, Mac, Android,.... Passionnée par l'informatique mais autodidacte, ma famille me « pousse » donc à reprendre mes études, à tenter un virage professionnel audacieux et très important pour moi.

En 2012, j'obtiens mon titre de « Technicien en Assistance Informatique » (niveau IV) à l'AFPA St Jérôme de Marseille. Je travaille comme technicienne réseaux chez DartyBox Marseille pendant 6 mois et enchaîne avec une formation de « Développeur de Logiciels » (niveau III), toujours à Marseille en 2013. Je n'ai pas pu passer mon titre suite à un grave problème de santé qui m'a immobilisé pendant 2 mois. Ne pouvant reprendre ma formation sur Marseille, je suis entrée en contact avec Natacha M2i qui m'a proposé de suivre la formation CDI. Et me voilà.... J'envisage, rapidement, de suivre une formation SYMPHONY afin de bien maîtriser ce framework.

1.2 - ABSTRACT

Ce projet s'inscrit dans la démarche d'urbanisation du système d'information hospitalier. En effet, beaucoup de communications inter applicatives reposent sur des données structurantes et dont le contenu doit être référencé de façon identique entre plusieurs logiciels.

Le référentiel des prescripteurs est un exemple parfaitement adapté à la situation. Lorsqu'une prescription, quelle que soit sa nature (Imagerie, Analyse de sang, Produit Médicamenteux, ...), est émise par un logiciel, l'application réceptrice de cette prescription n'a peut-être pas connaissance du médecin à l'origine de la demande. Cela va induire un travail de rattrapage et de correction qui pourrait parfaitement être évité si la donnée du prescripteur était synchrone dès le départ entre les deux logiciels.

This project joins in the initiative of urbanization of the hospitable information system. Indeed, many communications inter application base on structuring data and the contents of which must be referenced in a identical way between several software.

The reference table of the influencers is an example perfectly adapted to the situation. When a prescription, whatever is its nature (Imaging, Blood test, Medicinal product), is emitted by a software, the receiving application of this prescription maybe has no knowledge of the doctor at the origin of the demand. It is going to infer a work of catching up and correction which could be perfectly avoided if the datum of the influencer was synchronous from the beginning between both software.

1.3 - REMERCIEMENTS

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la rédaction de ce rapport.

Tout d'abord, j'adresse mes plus profonds remerciements à mon professeur, **Mr Pascal BUGUET** qui, grâce à son écoute et ses conseils, m'a énormément aidé durant ce stage. Il fut d'une aide précieuse dans les moments les plus délicats.

Je tiens à remercier mon maître de stage, **Mr Thierry DOMALAIN**, du SSIH, pour son accueil, le temps passé ensemble et le partage de son expertise. Grâce aussi à sa confiance, j'ai pu m'accomplir dans mes missions.

Je remercie également l'équipe du SSIH pour leur accueil, leur esprit d'équipe et tout particulièrement Philippe, Christophe, Nathalie qui m'ont supporté pendant ce stage ...

Un grand merci à toute l'équipe pédagogique de *M2i* et les intervenants professionnels responsables de la formation *Concepteur Développeur Informatique*, pour avoir assuré la partie théorique de celle-ci.

Enfin, je tiens à remercier ma famille, pour son soutien et sa patience. Elle m'a permis de réaliser ce projet et accepté que je m'absente pendant 5 mois pour suivre cette formation à Paris.

1.4 - LA DÉMARCHE GÉNÉRALE

Cette démarche est préconisée par Pascal Roques dans son livre « UML2 - Modéliser une application web » publié chez Eyrolles. C'est celle que nous suivons avec quelques aménagements.

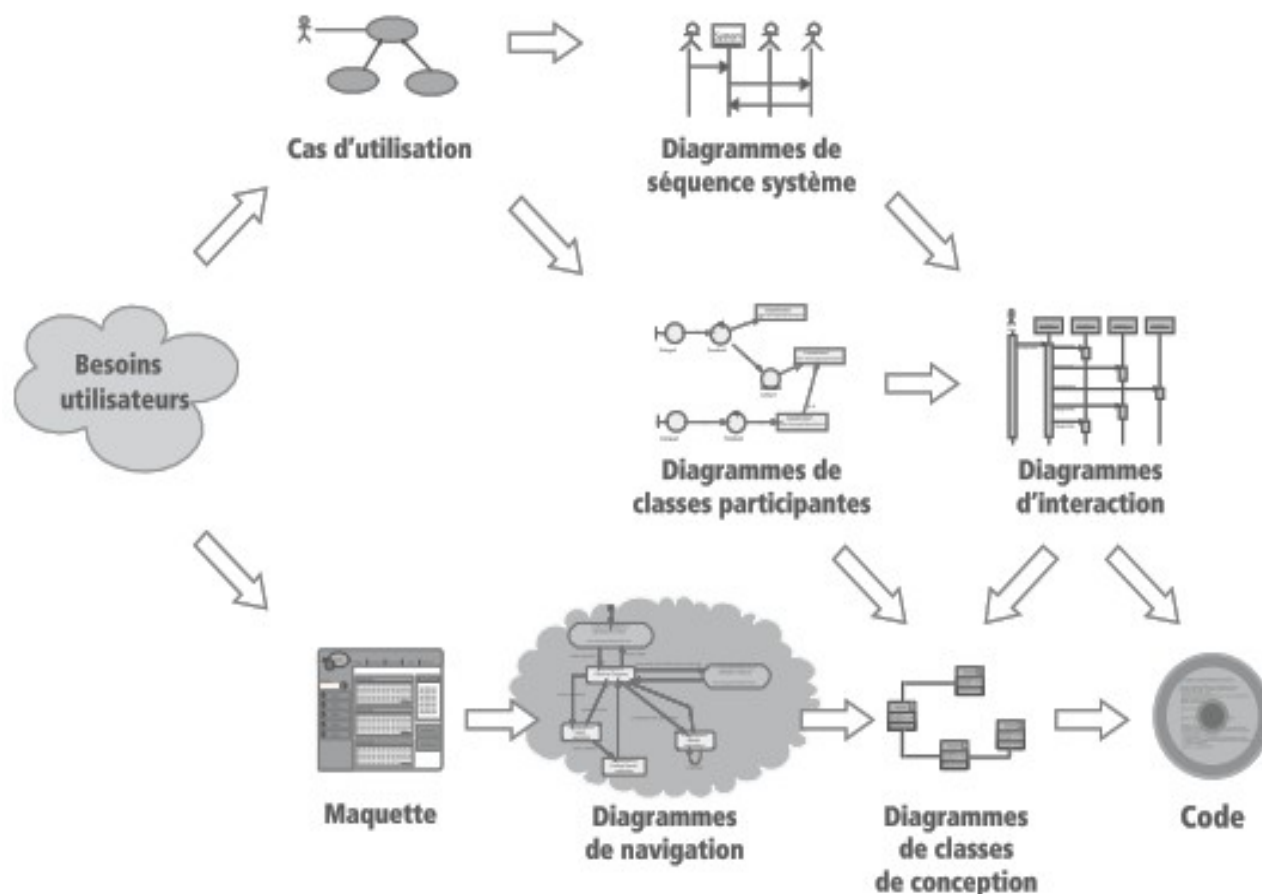


Figure 1–20 Schéma complet du processus de modélisation d'une application web

Chapitre 2 - *CAHIER DES CHARGES*

2.1 - PRÉSENTATION

Le projet consistera à mettre en œuvre un système qui va contenir les ensembles suivants :

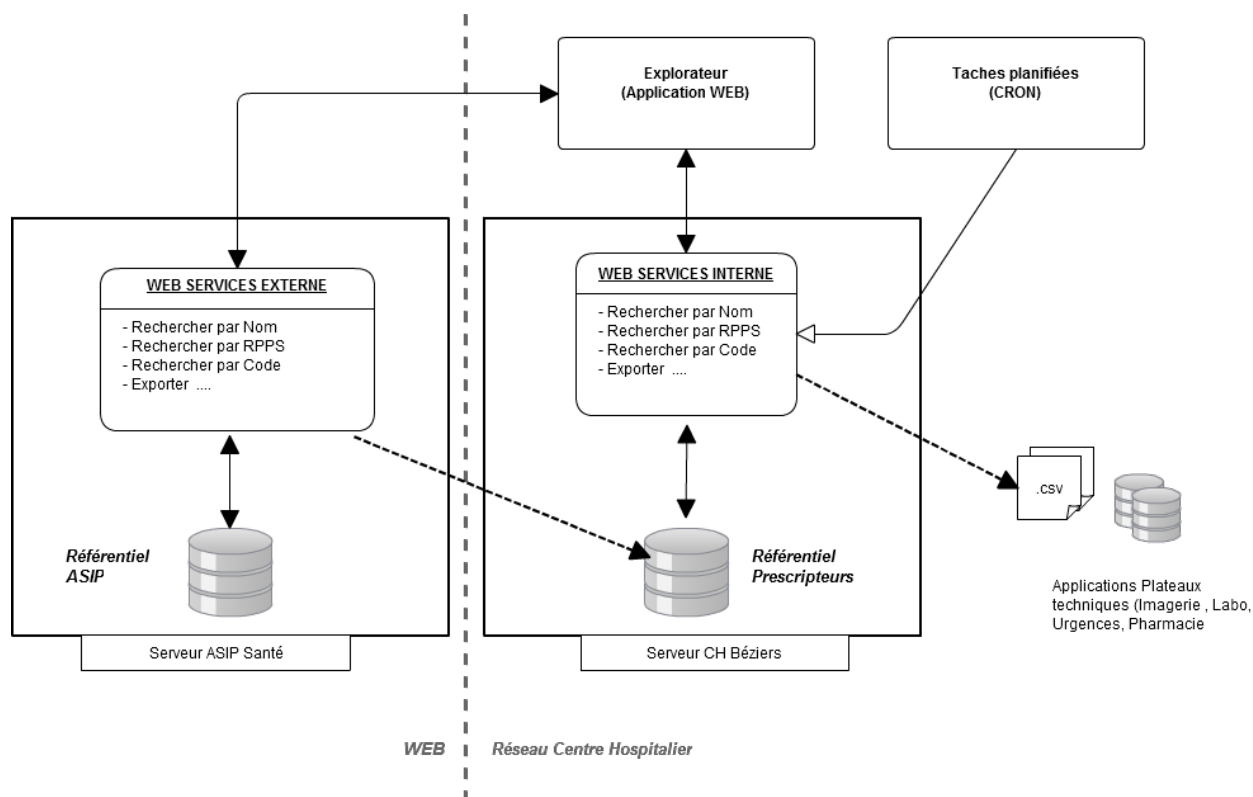
- Une base de données qui va contenir l'ensemble des données de Médecin (Nom, Prénom, E.mail, RPPS, etc..). La BDD ne doit pas être à l'identique à la base RPPS , il s'agira de notre base car elle va contenir des identifiants propres à nos applications. La base générique ne sera pas exhaustive.
- Pour les web services , commencer par monter quelques services en interne pour interroger la base que vous allez monter. Vous devrez donc définir votre propre bibliothèque...
Une couche applicative, qui va proposer un ensemble de web services pour interagir avec cette liste (fonctions de recherche, d'ajout, d'export).
- Une application web qui va interroger ces web services et proposer une interface d'exploration et des fonctionnalités minimales de recherche et de gestion (insérer ou modifier une donnée médecin).
- Cette même application web devra être en capacité d'interroger un web service distant (celui de l'ASIP santé), afin de rapatrier des données sur la base locale. On fera l'accès distant au serveur de l'ASIP dans un deuxième temps.
- Un module d'export (piloté par des taches planifiées/cron) pour pousser une partie de ces données à destination des logiciels tiers (Pharmacie, Imagerie, Labo,...). Les logiciels tiers sont des logiciels métiers propres au monde de la santé (PHARMA de Computer Engineering, CROSSWAY de Maincare, XPLORE de EDL, GLIMS de ClinSys, etc...) pour lesquels il faudra exporter les données de la base

2.2 - TECHNOLOGIES IMPOSÉES :

Le stagiaire est libre d'utiliser les méthodologies, environnements et frameworks de développement de son choix, mais les technologies imposées sont :

- MySQL pour la base de données de référence,
- HTML / JavaScript pour la partie cliente,
- PHP pour la partie serveur,
- des Frameworks CSS (Bootstrap, Foundation, ...),
- JavaScript/ jQuery sont recommandés.

2.3 - SYNOPTIQUE FONCTIONNEL :



Le cahier des charges n'a pas été respecté durant mon stage. En effet, il m'a été demandé de créer une interface « simplifiée » interne afin que tous les services puissent accéder à la base de données, des médecins prescripteurs, créée et de l'alimenter.

En ce qui concerne les web services, ils ont été créés afin d'accéder à toutes les bases de données des logiciels métiers des différents services hospitaliers mais n'ont pas pu être testés.

Et pour l'annuaire UDDI, l'Hôpital n'ayant pas fait les démarches de demande d'inscription auprès de l'ASIP Santé, le projet est resté en « stand by ».

Grâce à ce cahier des charges, reçu 15 jours avant mon entrée en stage, je me suis vivement intéressée aux Services Web et aux annuaires UDDI (Universal Description Discovery and Integration). J'ai partiellement compris le système de ceux-ci et de l'impact qu'ils peuvent avoir pour des structures comme un hôpital.

Chapitre 3 - *ANALYSE*

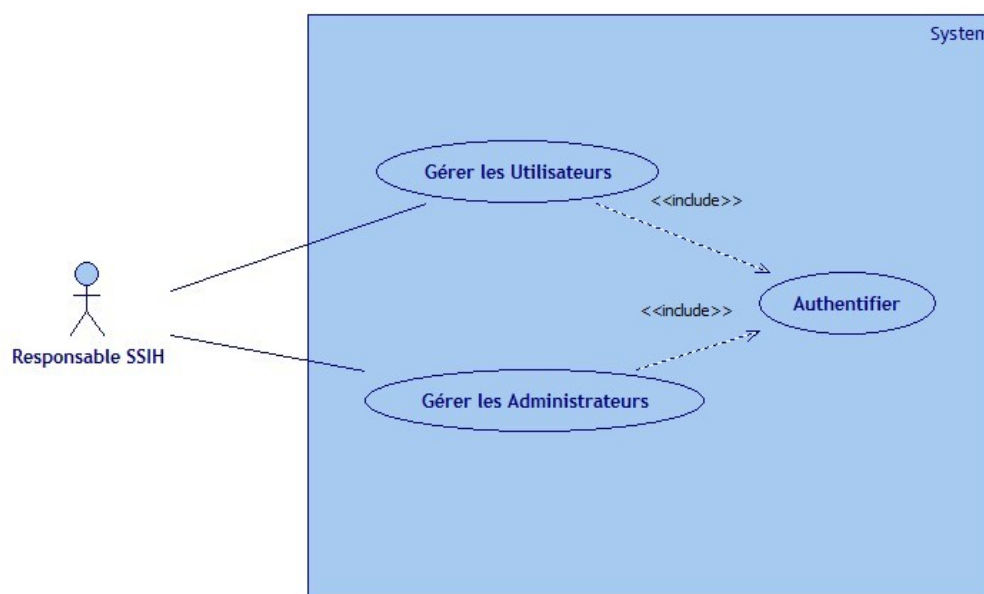
3.1 - LES DIAGRAMMES DE CAS D'UTILISATION

3.1.1 - Représentation graphique

Un **Diagramme de Cas d'Utilisation** représente les fonctionnalités (ou dit cas d'utilisation) nécessaires aux utilisateurs.

3.1.2 - DCU Back-Office

Diagramme de Cas d'Utilisation BackOffice
CH Béziers

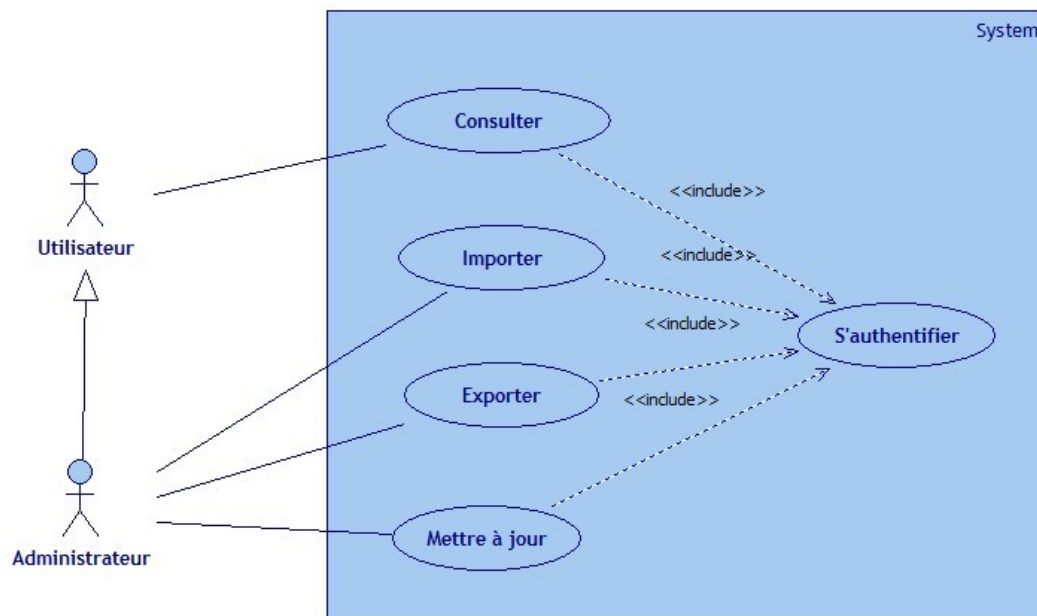


La création d'une interface Back-Office va permettre au Responsable SSIH de gérer les profils pour un :

- Utilisateur : consultation simple de la base de données,
- Administrateur : consultation, importation, exportation et mise à jour de la base de données.

3.1.3 - DCU Front-Office

Diagramme de Cas d'Utilisation FrontOffice
CH Béziers



La création d'une interface Front-Office va permettre aux :

- Utilisateur : de consulter si un médecin prescripteur est enregistré sur la base de données,
- Administrateur : de consulter, importer, exporter et mettre à jour un médecin prescripteur sur la base de données.

3.1.4 - Fiche de description textuelle d'un cas d'utilisation

Rubriques	Description
Identification	Titre, type, résumé, acteurs, date de création, date de dernière modification, version, auteur
Pré-conditions	Liste des conditions qui doivent être vérifiées pour le déclenchement d'un des scénarios (authentification, écran disponible, service disponible, ...)
Scenario nominal	Liste des actions élémentaires pour le cas le plus courant
Scénarii d'erreurs du cas nominal	Liste des actions exécutées en cas d'erreurs
Scénarii alternatifs	Liste des actions élémentaires pour les cas exceptionnels
Scénarii d'erreurs des cas alternatifs	Liste des actions exécutées en cas d'erreurs
Post-conditions	Liste des conditions vérifiées après l'arrêt des scénarios
Exigences non fonctionnelles (Optionnel)	Temps de réponse, concurrentialité, disponibilité, ...
Besoins d'IHM (Optionnel)	Liste des dispositifs d'entrée/sortie

3.1.5 - Les diagrammes de Séquence Système

Un Diagramme de Séquence Système (DSS) permet de décrire et de formaliser les interactions entre les acteurs et le système ou les classes, par l'intermédiaire de messages, d'un point de vue chronologique et spatial. Le temps est représenté à la verticale et l'espace à l'horizontal.

Diagramme de Séquence Système : Authentification

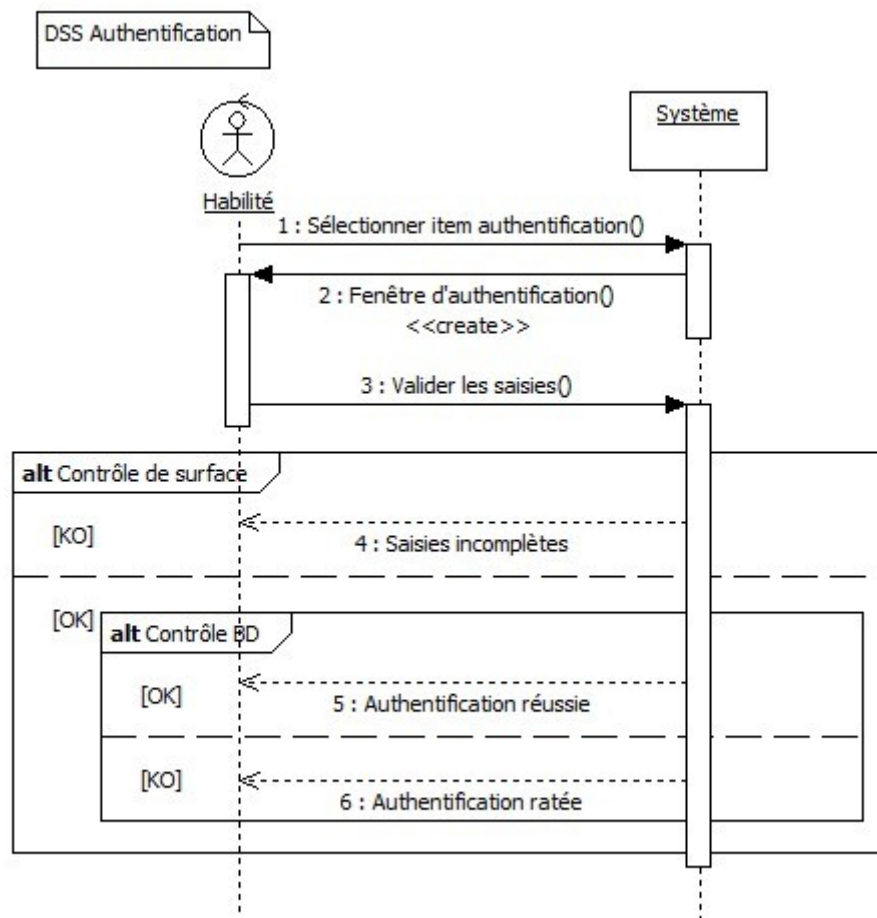
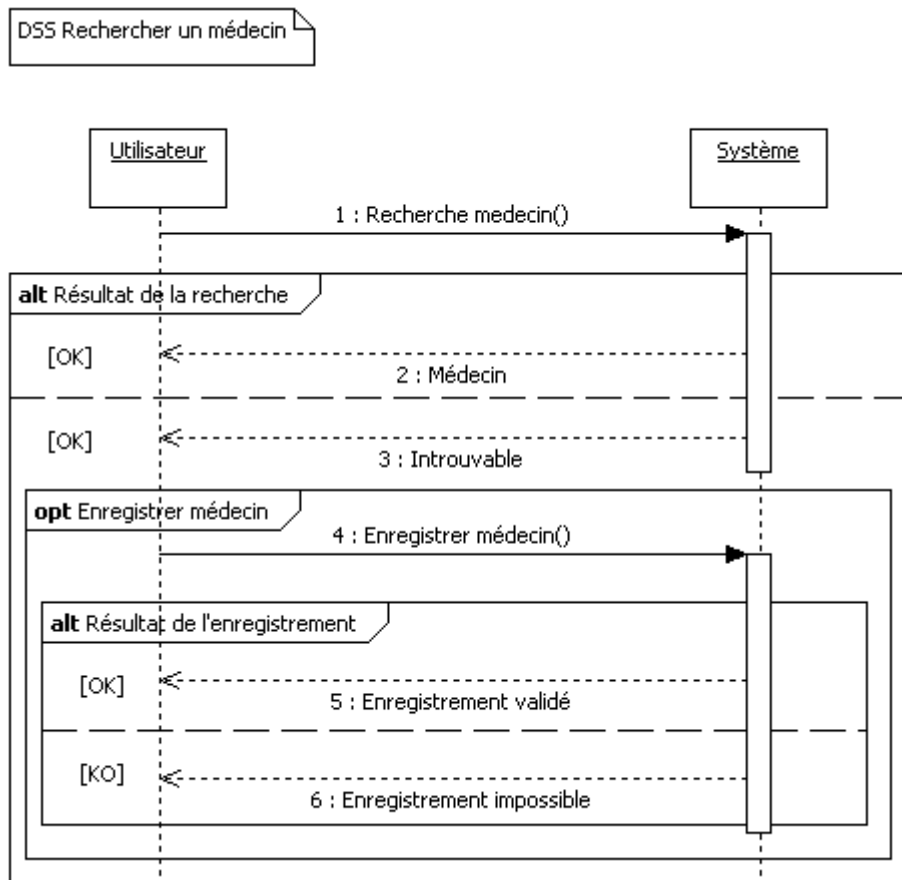


Diagramme de Séquence Système : Rechercher un médecin

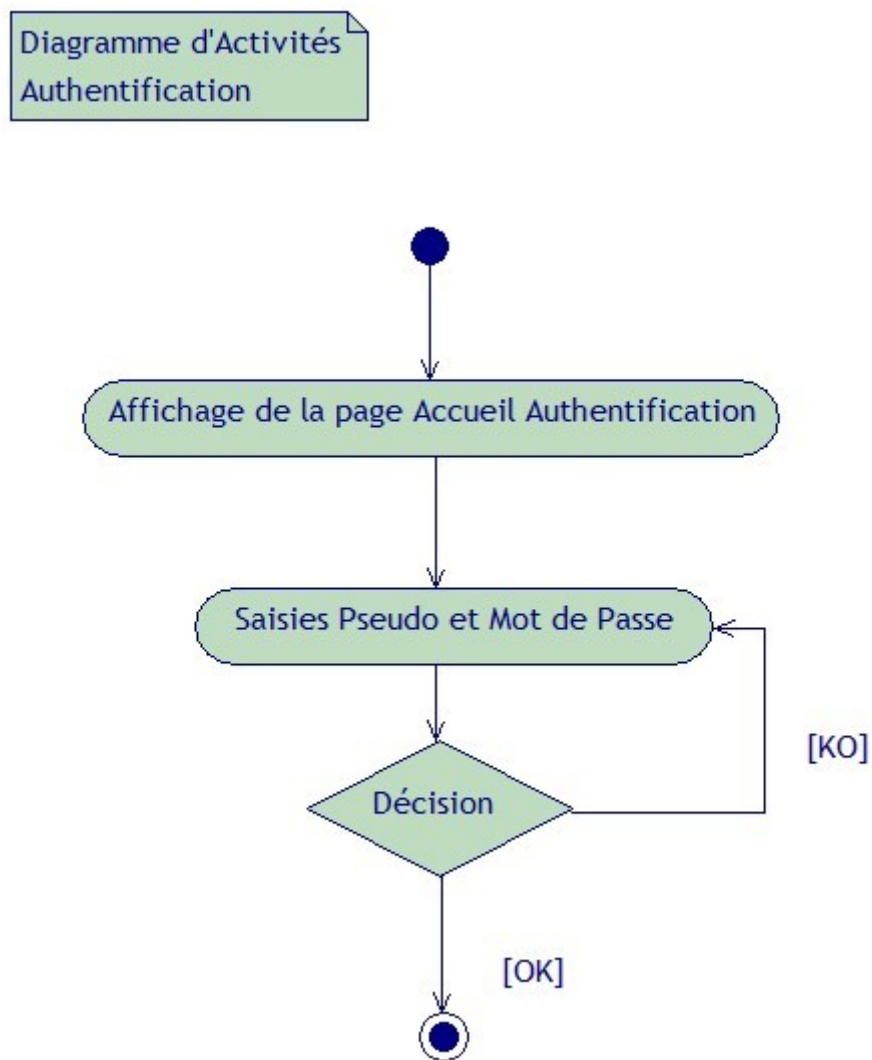


3.1.6 - Diagramme d'activité

Un **Diagramme d'Activité** permet de représenter le déroulement d'un cas d'utilisation.

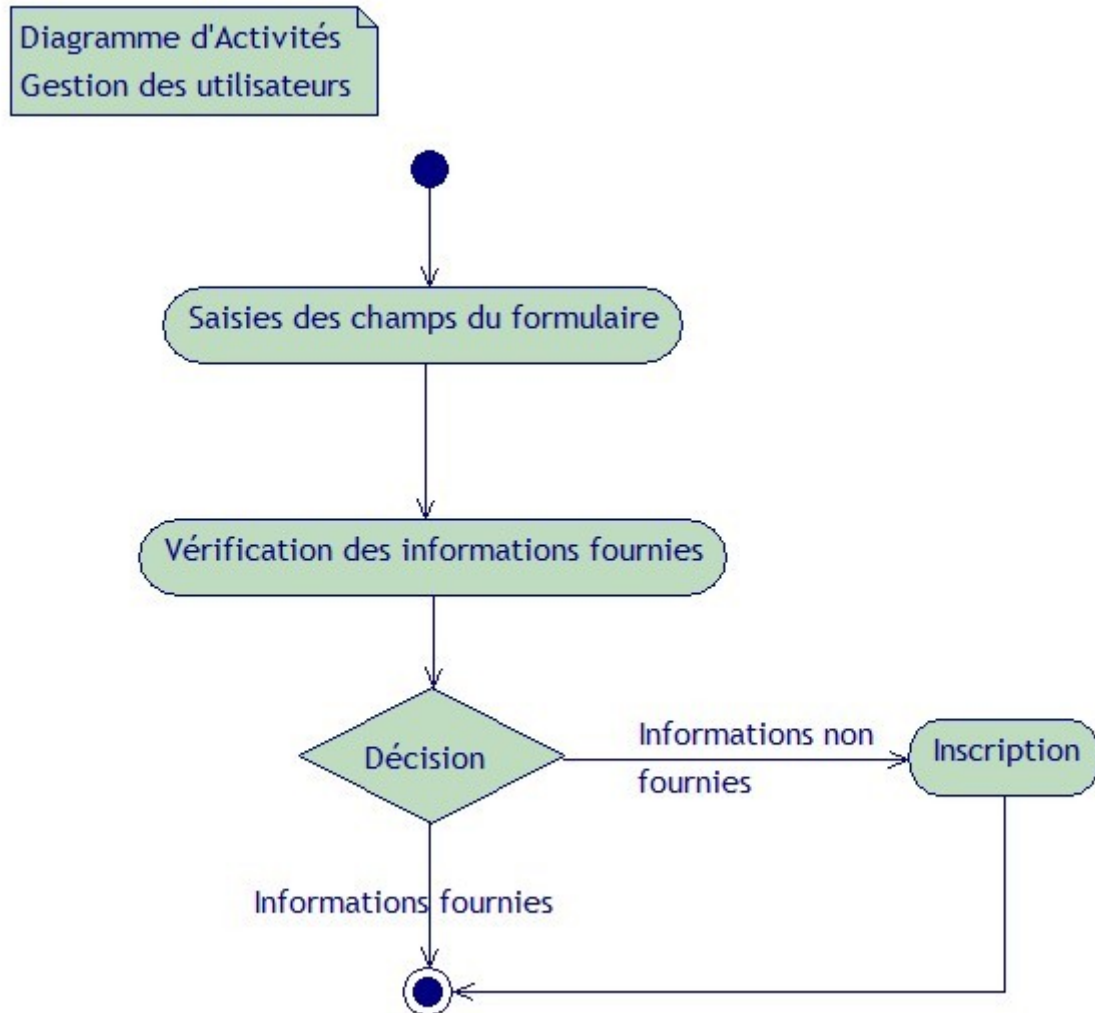
3.1.6.1 - **FRONT-OFFICE**

Page Accueil pour l'Authentification



3.1.6.2 - **BACK OFFICE**

Gestion des utilisateurs



3.2 - LES MAQUETTES

L'ACCUEIL



Bandeau de navigation et la Connexion :

Annuaire des Médecins

[Accueil](#) [Gérer les profils utilisateur](#) [Gérer les utilisateurs](#) [Rechercher un médecin](#)

Pseudo : Mot de passe :

Connexion réussie :

Connecté(e)

Déconnexion ou authentification échouée :

Déconnecté(e)

Si les champs de formulaire ne sont pas renseignés :

Pseudo : Mot de passe :

Toutes les saisies sont obligatoires

GESTION DES PROFILS D'UTILISATEURS

Lors d'un ajout de profil :

Gérer les profils

ID profil :

Code profil :

*

Libellé profil :

*

Le libellé est obligatoire et est du texte

Annuler

Ajouter

Modifier

Supprimer

Lors de la suppression d'un profil avec sélection par liste déroulante :

Gérer les profils

Pour la mise à jour, sélectionnez un profil

-Sélectionnez un profil- ▼

ID profil :

Code profil :

*

Libellé profil :

*

Annuler

Ajouter

Modifier

Supprimer

GESTION DES UTILISATEURS

Gestion des utilisateurs avec sélection par liste déroulante :

Gérer les utilisateurs

Pour la mise à jour, sélectionnez un utilisateur

ID utilisateur :



Nom :

*

Prénom :

*

Pseudo :

*

Mot de passe :

*

E.mail :

*

Profil ?

Annuler

Ajouter

Modifier

Supprimer

RECHERCHE D'UN MEDECIN

Recherche d'un médecin (résultat positif) :

[Accueil](#) [Rechercher un médecin](#) [Lister les médecins](#)

Pseudo : Mot de passe : Connecté(e)

CRUD médecin

Nom :	* <input type="text" value="AARON"/>
Prénom :	* <input type="text" value="Caroline"/>
Téléphone :	* <input type="text"/>
E-mail :	* <input type="text" value="caroline@aarons.org"/>
RPPS :	* <input type="text" value="10000782101"/>
Spécialité :	* <input type="text"/>

Le médecin existe dans la BD

© 2015 Michèle BOIS-PLAISANCE

Recherche d'un médecin (résultat négatif) :

Pseudo : Mot de passe : Connecté(e)

CRUD médecin

Nom :
 Prénom :
 Téléphone :
 E-mail :
 RPPS :
 Spécialité :

Voulez-vous créer ce médecin ?

Médecin introuvable !

© 2015 Michèle BOIS-PLAISANCE

Si l'utilisateur clique sur OUI, le bouton « Enregistrer » apparaît :

[Accueil](#) [Rechercher un médecin](#) [Lister les médecins](#)

Pseudo : Mot de passe : Connecté(e)

CRUD médecin

Nom :
 Prénom :
 Téléphone :
 E-mail :
 RPPS :
 Spécialité :

Médecin introuvable !

© 2015 Michèle BOIS-PLAISANCE

Lister les médecins :

Afficher 10 entrées

Recherche:

ID	Nom	Prénom	E-mail	RPPS
1	AARON	Caroline	caroline@aaron.org	10000782101
2	ABADIE	Benjamin	benjamin@abadie.org	10100461556
3	CHOLLET	Nicolas	nicolas@chollet.org	10100436160
4	DESCHAMPS	Eric	eric@deschamps.org	10002398476
5	GRUEZ	Laurent	laurent@gruez.org	10002242997
6	LEROI	Sébastien	sebastien@leroi.org	10002090735

Affiche 1 à 6 des 6 entrées

Précédente 1 Suivante

Lister les médecins à partir de la recherche alphabétique :

Afficher 10 entrées

Recherche: GR

ID	Nom	Prénom	E-mail	RPPS
5	GRUEZ	Laurent	laurent@gruez.org	10002242997

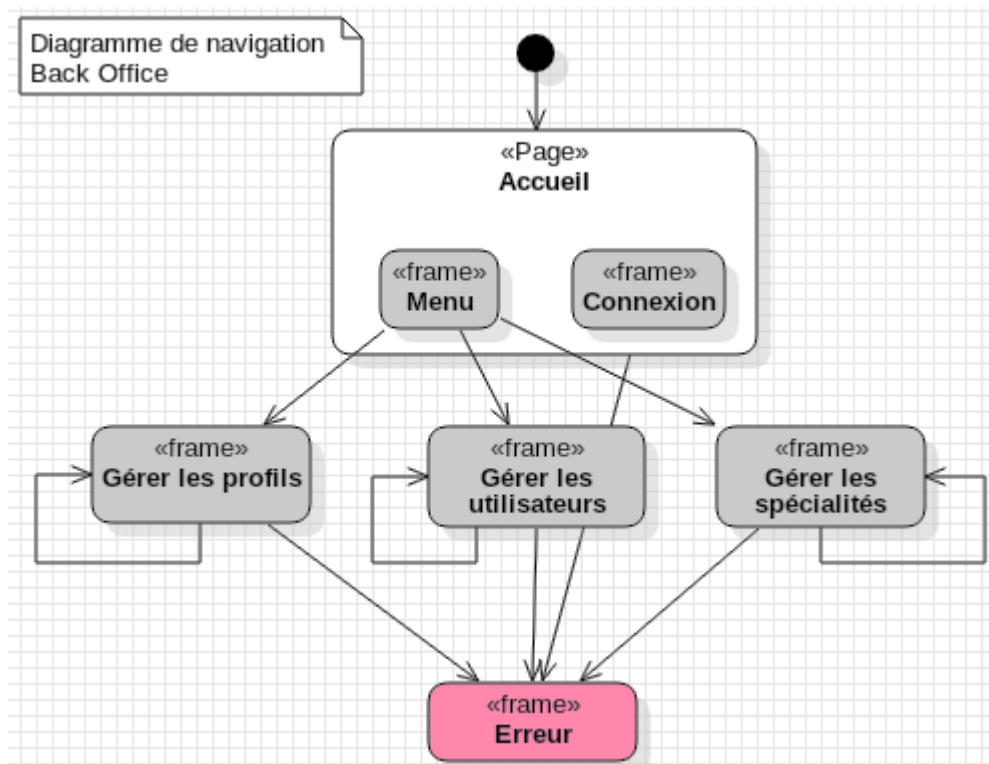
Affiche 1 à 1 des 1 entrées (filtered from 6 total entries)

Précédente 1 Suivante

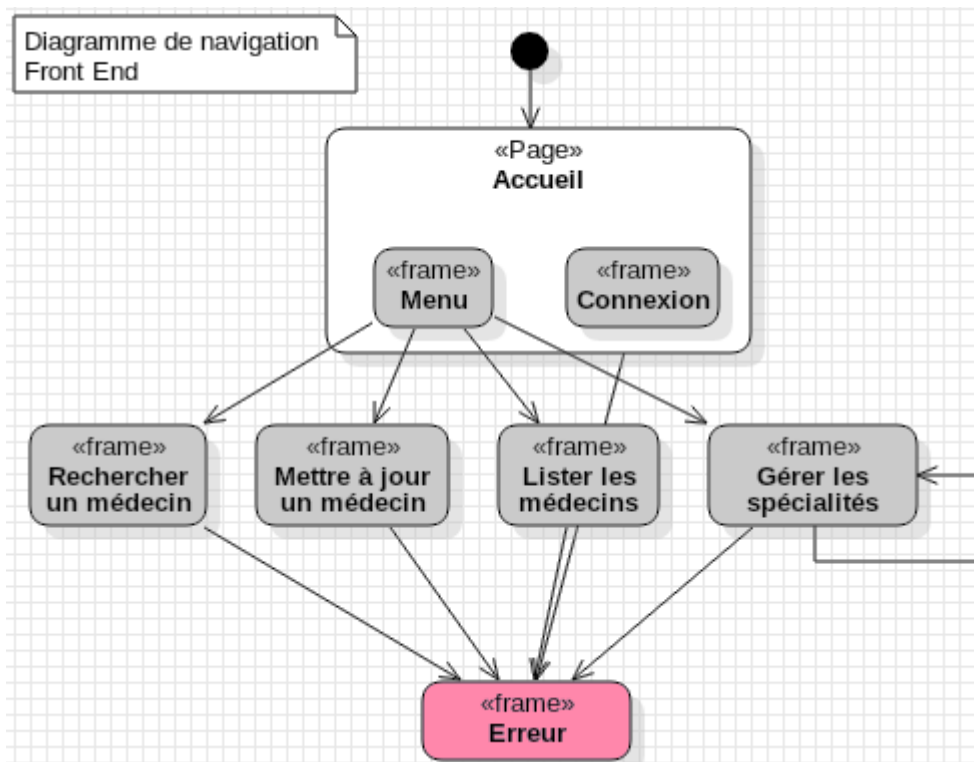
3.3 - LES DIAGRAMMES DE NAVIGATION

Les Diagrammes de Navigation servent à représenter le cheminement de l'application ou couche contrôle, entre les différents écrans (IHM) de l'application.

3.3.1 - Diagramme de Navigation du Back-Office :



3.3.2 - Diagramme de Navigation du Front-End :



Chapitre 4 - *CONCEPTION DE LA BASE DE DONNÉES*

4.1 - LA DÉMARCHE UTILISÉE

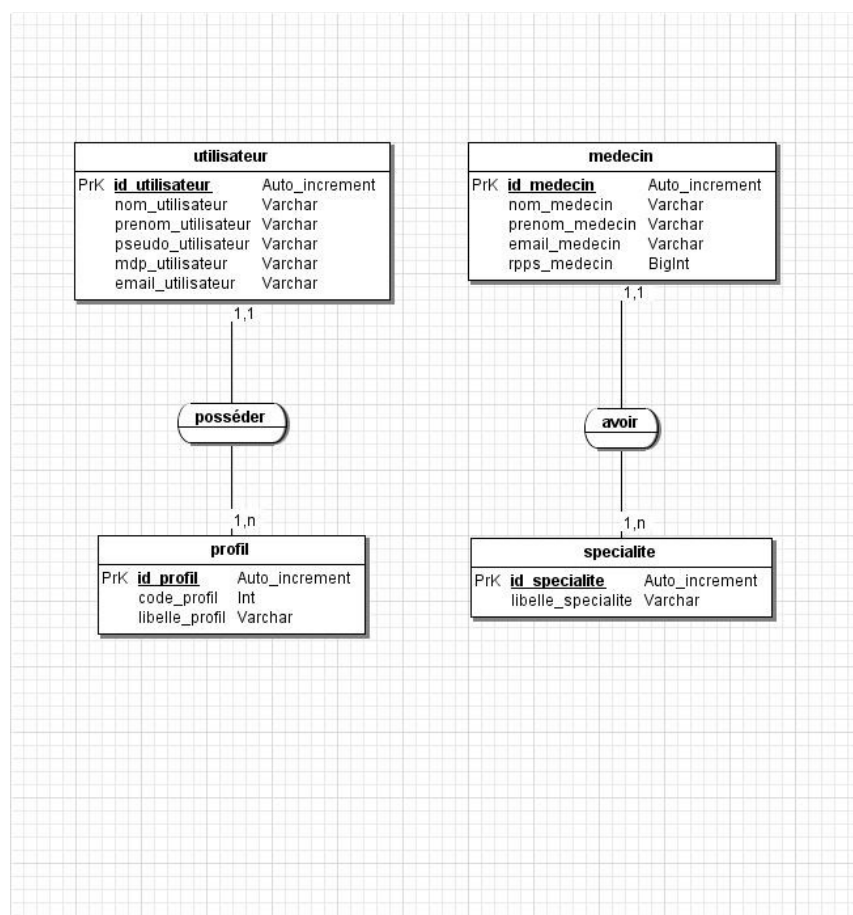
Soit la démarche par l'analyse du discours (Méthode Chen) : un nom égal une classe ou un attribut et un verbe est à une association.

Il faut passer ensuite le modèle à la « moulinette » des formes normales.

Soit la méthode des dépendances fonctionnelles. Et certains « compléments » sont obtenus avec la méthode précédente.

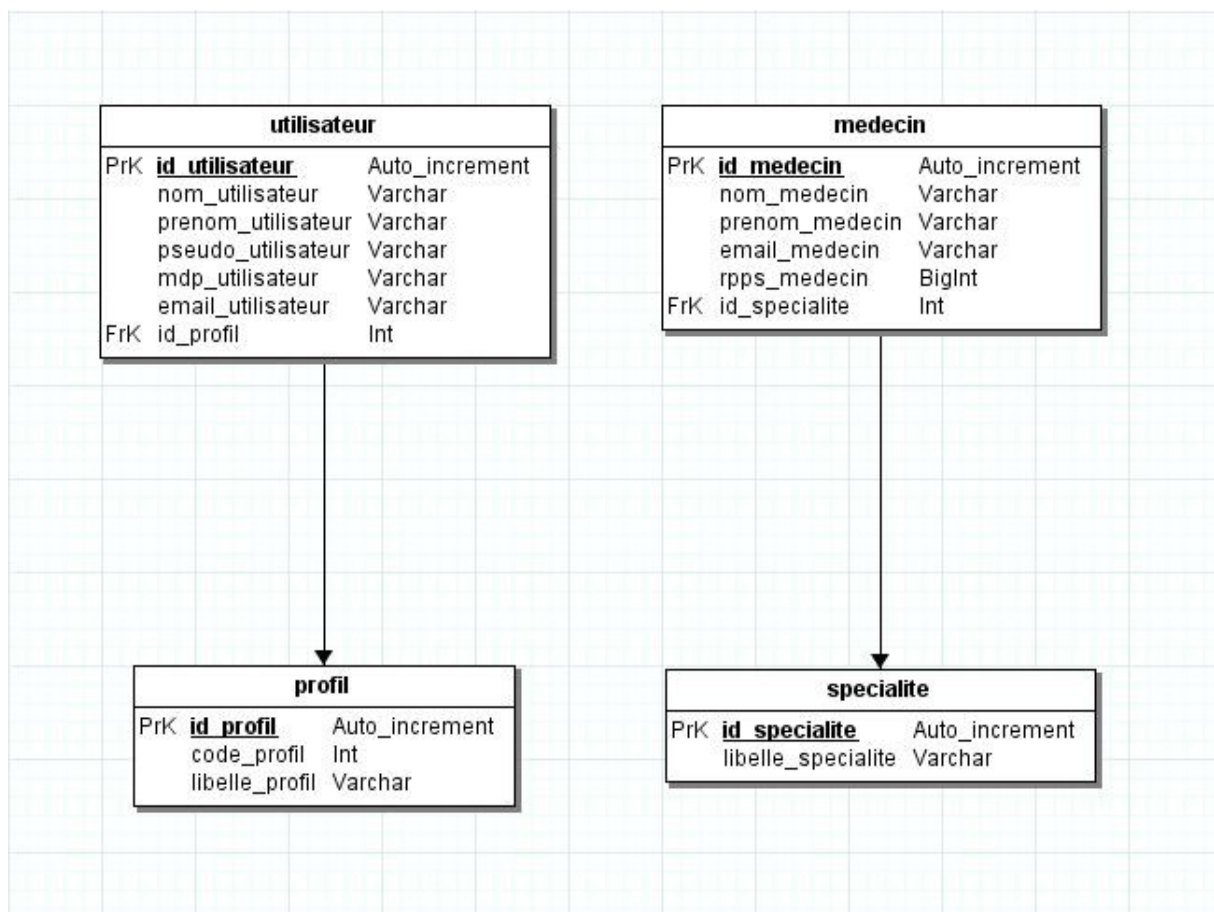
4.2 - LE MCD (MERISE)

Le Modèle Conceptuel des Données - MCD (ou Modèle entité-association), permet de représenter la structure du système d'information, du point de vue des données, et définit également les dépendances ou relations entre ces différentes données.



4.3 - LE MLD (MODÈLE LOGIQUE DES DONNÉES)

Le MLD (Modèle Logique de Données) représente la structure de la base de données.



4.4 - LE SCHÉMA DE LA BASE DE DONNÉES

4.5 - QUELQUES EXEMPLES DE CODE SQL

4.5.1 - Création de la Base de données : `annuaire_medecins`

```
DROP DATABASE IF EXISTS `annuaire_medecins`;
CREATE DATABASE IF NOT EXISTS `annuaire_medecins` DEFAULT CHARACTER SET
utf8 COLLATE utf8_general_ci;
USE `annuaire_medecins`;
```

```
--
-- Table structure for table `specialite`
--
```

```
DROP TABLE IF EXISTS `annuaire_medecins`.`specialite`;
CREATE TABLE `annuaire_medecins`.`specialite` (
  `id_specialite` int(11) NOT NULL AUTO_INCREMENT,
  `libelle_specialite` varchar(100) NOT NULL,
  PRIMARY KEY (`id_specialite`),
  UNIQUE KEY `libelle_specialite` (`libelle_specialite`)
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;
```

```
--
-- Table structure for table `medecin`
--
```

```
DROP TABLE IF EXISTS `annuaire_medecins`.`medecin`;
CREATE TABLE `annuaire_medecins`.`medecin` (
  `id_medecin` int(11) NOT NULL AUTO_INCREMENT,
  `nom_medecin` varchar(50) DEFAULT NULL,
  `prenom_medecin` varchar(50) DEFAULT NULL,
  `email_medecin` varchar(100) DEFAULT NULL,
  `rpps_medecin` bigint(20) DEFAULT NULL,
  `id_specialite` int(11) NOT NULL,
  PRIMARY KEY (`id_medecin`),
  UNIQUE KEY `email_medecin` (`email_medecin`),
  UNIQUE KEY `idx_medecin_rpps_medecin` (`rpps_medecin`),
  KEY `id_specialite` (`id_specialite`),
  CONSTRAINT `medecin_ibfk_1` FOREIGN KEY (`id_specialite`) REFERENCES
`specialite` (`id_specialite`)
) ENGINE=InnoDB AUTO_INCREMENT=15 DEFAULT CHARSET=utf8;
```

cf dans les Annexes le code complet.

4.5.2 - Création des Procédures Stockées : `annuaire_medecins`

Une procédure stockée est une sous routine stockée dans le catalogue de la base de données. Les applications peuvent appeler et exécuter une procédure stockée avec la requête CALL.

L'intérêt des procédures stockées est qu'elles permettent une exécution plus rapide du code, et plus sécurisée. De plus si nous devons changer de logiciel client, ou en créer un autre, ce qui pourrait être le cas dans le futur avec des modules Java, l'intérêt des procédures stockées est encore plus grand puisqu'il n'y aura pas à réécrire du code SQL dans du code Java.

Le script qui suit a été généré par un générateur de procédures stockées pour une Base de données spécifiée.

Il a été ensuite exécuté via un client MySQL.

Cf le code complet dans les Annexes.

```
DELIMITER $$
```

```
USE annuaire_medecins $$
```

```
--
```

```
-- LES DROPS
```

```
--
```

```
-- DROPS POUR medecin --
```

```
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinSelectAll $$
```

```
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinSelectOne $$
```

```
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinInsert $$
```

```
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinUpdate $$
```

```
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinDelete $$
```

```
DELIMITER ;
```

```
--
```

```
-- LES CREATEs
```

```
--
```

```
DELIMITER $$
```

```
-- CREATEs POUR medecin --
```

```
-- CREATE SelectAll POUR annuaire_medecins.medecin
```

```
CREATE PROCEDURE annuaire_medecins.medecinSelectAll()
```

```
BEGIN
```

```
    SELECT *
```

```
    FROM annuaire_medecins.medecin ;
```

```
END $$
```

```
-- CREATE SelectOne POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinSelectOne(paramID_MEDECIN
INT(10) )
BEGIN
    SELECT *
    FROM annuaire_medecins.medecin
    WHERE id_medecin = paramID_MEDECIN ;
END $$

-- CREATE Insert POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinInsert(paramID_MEDECIN
INT(10),paramNOM_MEDECIN VARCHAR(50),paramPRENOM_MEDECIN
VARCHAR(50),paramEMAIL_MEDECIN VARCHAR(100),paramRPPS_MEDECIN
BIGINT(19),paramID_SPECIALITE INT(10))
BEGIN
    INSERT INTO
annuaire_medecins.medecin(id_medecin,nom_medecin,prenom_medecin,email_
medecin,rpps_medecin,id_specialite)

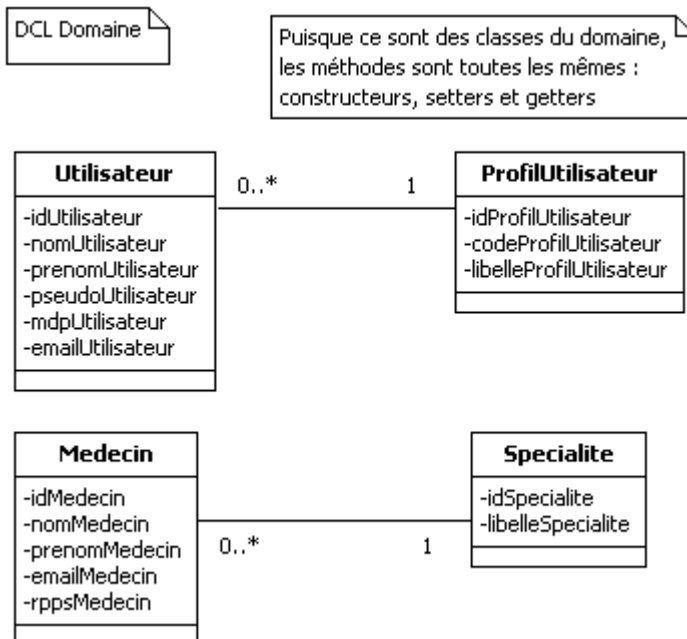
VALUES(paramID_MEDECIN,paramNOM_MEDECIN,paramPRENOM_MEDECIN,para
mEMAIL_MEDECIN,paramRPPS_MEDECIN,paramID_SPECIALITE) ;
END $$

-- CREATE Update POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinUpdate(paramID_MEDECIN
INT(10),paramNOM_MEDECIN VARCHAR(50),paramPRENOM_MEDECIN
VARCHAR(50),paramEMAIL_MEDECIN VARCHAR(100),paramRPPS_MEDECIN
BIGINT(19),paramID_SPECIALITE INT(10))
BEGIN
    UPDATE annuaire_medecins.medecin
    SET nom_medecin = paramNOM_MEDECIN,prenom_medecin =
paramPRENOM_MEDECIN,email_medecin = paramEMAIL_MEDECIN,rpps_medecin
= paramRPPS_MEDECIN,id_specialite = paramID_SPECIALITE
    WHERE id_medecin = paramID_MEDECIN ;
END $$

-- CREATE Delete POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinDelete(paramID_MEDECIN
INT(10) )
BEGIN
    DELETE
    FROM annuaire_medecins.medecin
    WHERE id_medecin = paramID_MEDECIN ;
END $$

DELIMITER ;
```

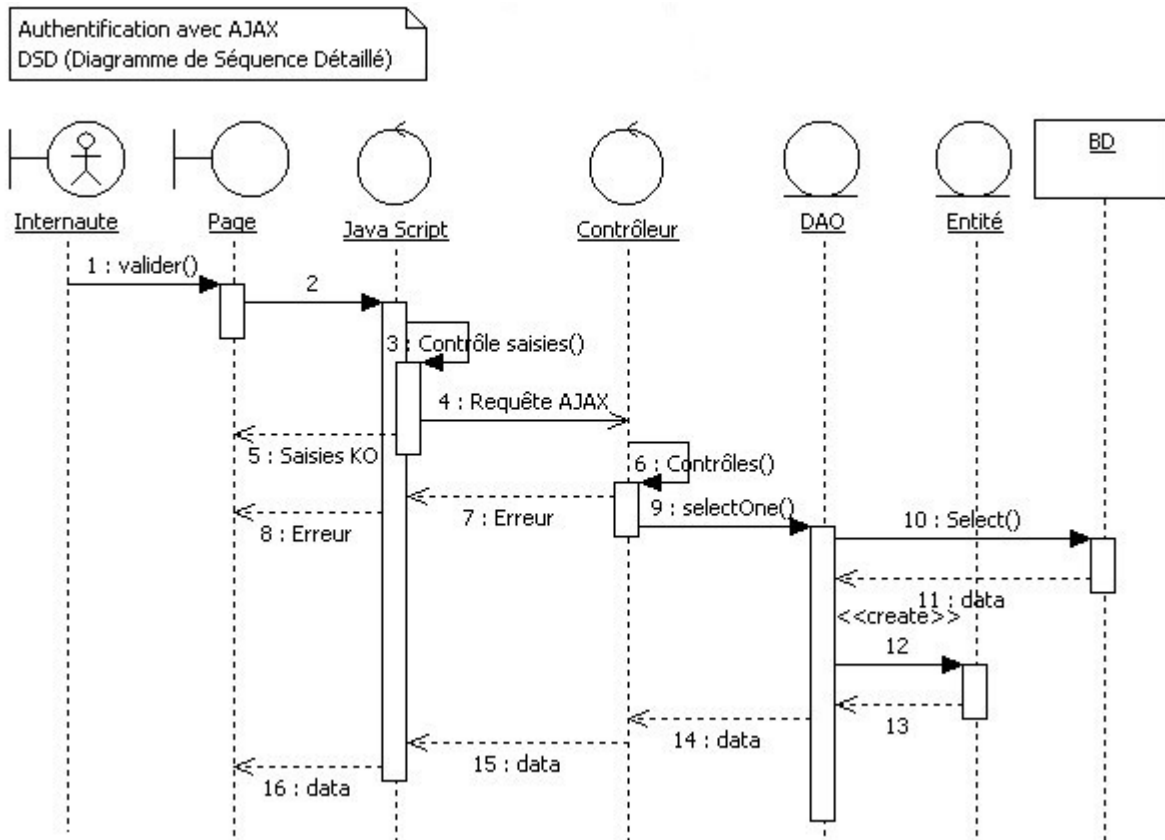
4.6 - LE DIAGRAMME DE CLASSES DU DOMAINE



Chapitre 5 - *CONCEPTION DE L'APPLICATION*

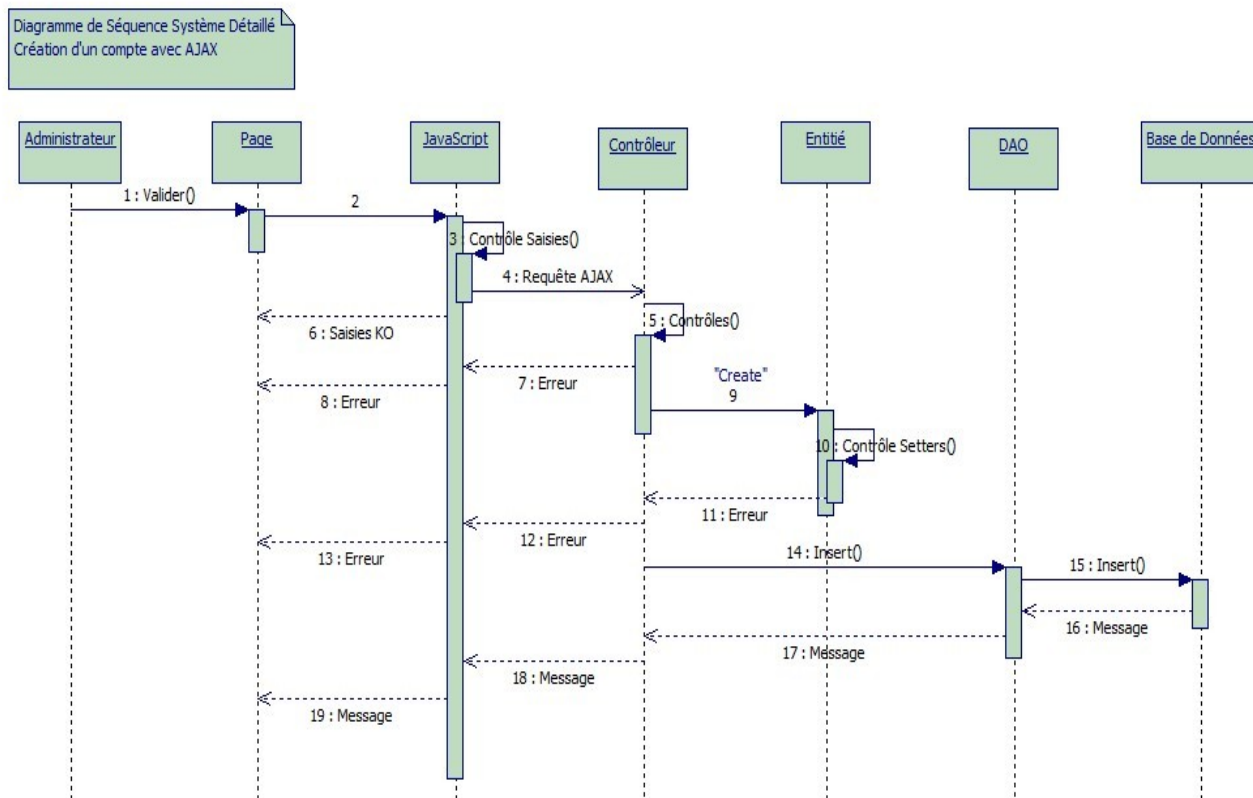
5.1 - LES DIAGRAMMES DE SÉQUENCE DÉTAILLÉS

5.1.1 - Diagramme de Séquence Détaillé : Authentification avec AJAX



Ce diagramme est basé sur du READ et concerne la connexion des utilisateurs afin de mettre en évidence les étapes de vérification de la présence de l'utilisateur dans la base de données.

5.1.2 - Diagramme de Séquence Détaillé : Création d'un compte avec AJAX



Ce diagramme est basé sur du CREATE, UPDATE et DELETE. Il concerne la création d'utilisateurs dans la base de données.

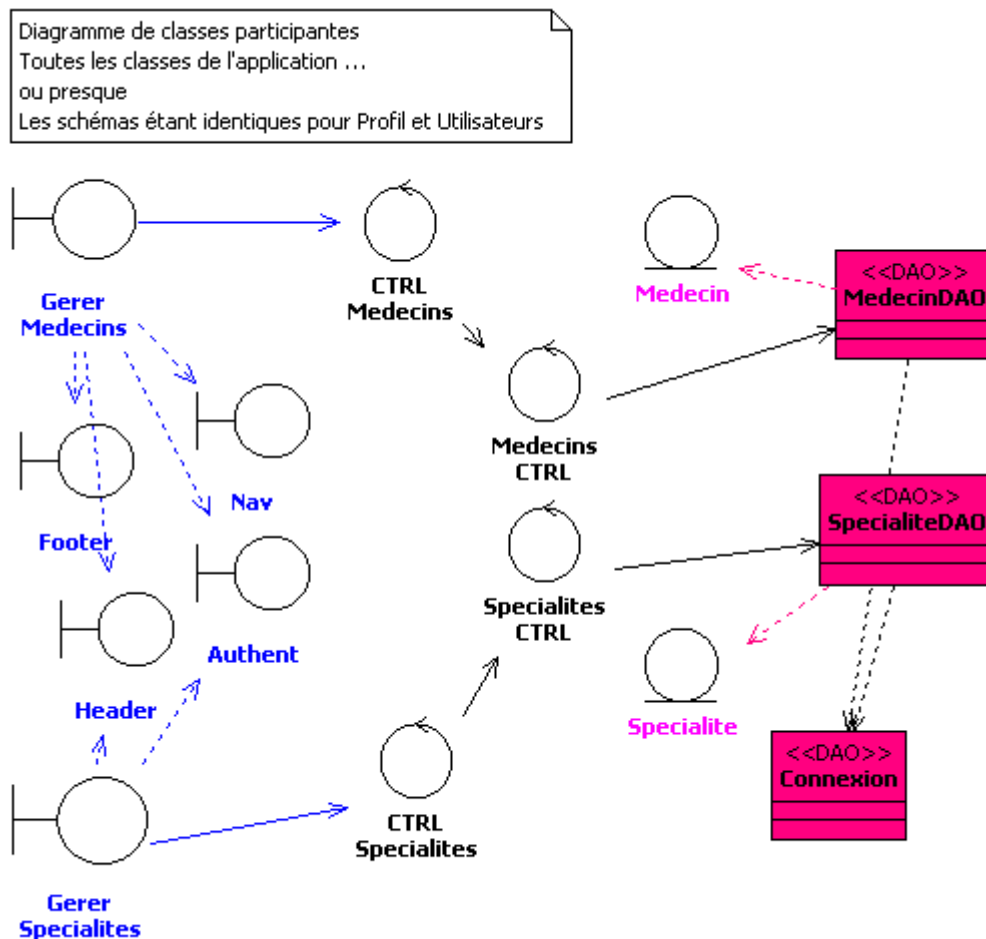
Les Boundaries (Pages Web, Fenêtres Swing, Activités Android, ...).

Les contrôleurs.

Les Entities et les DAOs.

5.2 - LES DIAGRAMMES DE CLASSES PARTICIPANTES

C'est un diagramme de classes où toutes les classes du Diagramme de séquence détaillé sont représentées.



Chapitre 6 - *DÉVELOPPEMENT*

6.1 - LES CATÉGORIES DE CODE

6.1.1 - Codes HTML-PHP

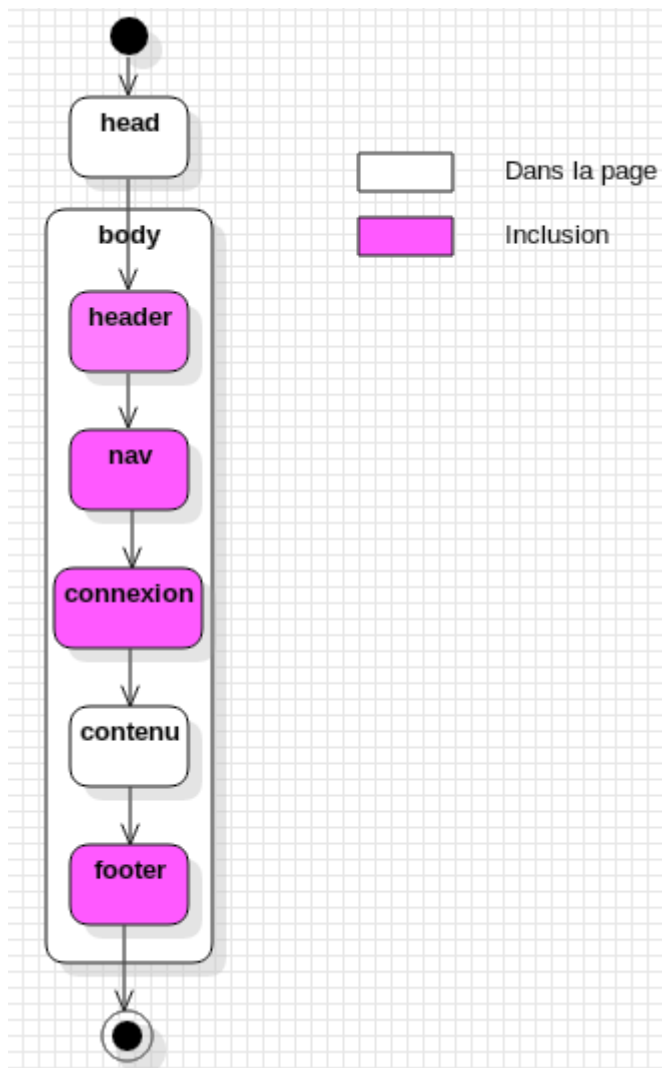
6.1.2 - Codes CSS

6.1.3 - Codes JavaScript

6.1.4 - Codes AJAX

6.1.5 - Codes PHP

6.2 - STRUCTURE D'UNE PAGE PHP : LE PRINCIPE DES INCLUSIONS



```
<?php
/*
 * accueil.php
 */

define('TITRE_PAGE', 'Annuaire des Médecins');

/*
 * require_once : PHP vérifie si le fichier a déjà été inclus,
 * et si c'est le cas, ne l'inclut pas une deuxième fois,
 * et génère une erreur si le fichier à inclure n'est pas trouvé.
 */
?>

<!DOCTYPE html>
<html>
  <head>
    <title>Accueil</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width">
    <link href="../../../css/chb.css" rel="stylesheet" />
    <link href="../../../css/formulaireHorizontal.css" rel="stylesheet" />
  </head>

  <body>
    <?php
    require_once('../../../parts/header.inc.php');
    ?>
    <br class="nettoyeur">
    <?php
    require_once('../../../parts/nav.inc.php');
    ?>

    <?php
    require_once('../../../parts/formConnexion.inc.php');
    ?>

    <br class="nettoyeur">

    <div id="contenu">
      <h1 id="h1Accueil">ACCUEIL</h1>
      
    </div>

    <?php
    require_once('../../../parts/footer.inc.php');
    ?>

  </body>
</html>
```

6.3 - LES CODES

6.3.1 - Exemple de code HTML-PHP

```
<?php
/*
 * Accueil.php
 */

define('TITRE_PAGE', 'Annuaire des Médecins');

/*
 * require_once : PHP verifie si le fichier a deja ete inclus,
 * et si c'est le cas, ne l'inclut pas une deuxieme fois,
 * et genere une erreur si le fichier a inclure n'est pas trouve.
 */
?>

<!DOCTYPE html>
<!--
-->
<html>
    <head>
        <title>Accueil</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width">
        <link href="../css/chb.css" rel="stylesheet" />
        <link href="../css/formulaireHorizontal.css"
rel="stylesheet" />
    </head>

    <body>
        <?php
        require_once('../parts/header.inc.php');
        ?>
        <br class="nettoyeur">
        <?php
        require_once('../parts/nav.inc.php');
        ?>

        <?php
        require_once('../parts/formConnexion.inc.php');
        ?>

        <br class="nettoyeur">

        <div id="contenu">
            <h1 id="h1Accueil">ACCUEIL</h1>
            <label id="lblMessage">
```



```
        </label>
        
    </div>

    <?php
    require_once('../../parts/footer.inc.php');
    ?>

</body>
</html>
```

6.3.2 - Exemple de code CSS

```
/*
chb.css
*/

h1{margin:0; padding:0.1em;}
h2{margin:0; padding:0.1em;}
h3{margin:0; padding:0.1em;}

input[type="submit">{text-align: center;}
input[type="reset">{text-align: center;}
input[type="button">{text-align: center;}

a{color: black;}

h1{text-align: center;
}

/* HEADER */
header
{
    font-family: "Trebuchet MS", Arial, sans-serif;

    border-radius: 30px;
    background-color: rgba(255, 255, 255, 0.5);
    box-shadow: 0px 0px 10px white;
    -moz-box-shadow: 0px 0px 10px white;
    -webkit-box-shadow: 0px 0px 10px white;
    -o-box-shadow: 0px 0px 10px white;
}

header h1
{
    text-transform: uppercase;
}

/* MENU */
nav{margin:0; padding:0.3em 0;}
nav ul { list-style: none; margin: 0; padding: 0; }
nav ul li { margin: 0.1em 0; float: left; display:inline-block;
padding-left:1em;}
nav ul li:first-child { margin: 0; padding-left:0em;}
```

6.3.3 - Exemple de code JavaScript

```
/*
 * connexionDeconnexion.js
 */

$("#btConnexion").click(validerSaisiesConnexion);
$("#btDeconnexion").click(deconnexion);

/**
 *
 * @returns {undefined}
 */
function validerSaisiesConnexion() {

    var lsMessage = "";
    var lsPseudo = $("#pseudoConnexion").val().trim();
    var lsMDP = $("#mdpConnexion").val().trim();

    // SI 1 des 2 zones de saisie est vide
    if (lsPseudo === "" || lsMDP === "") {
        lsMessage = "Toutes les saisies sont obligatoires";
        $("#labelMessageConnexion").html(lsMessage);
    }
    else {
        /* A FAIRE */
    }
}

} /// validerSaisiesConnexion
```

6.3.4 - Exemple de code AJAX

```

/**
 * MedecinGerer.js
 */

function initMedecinCRUD() {
    $("#btRechercherMedecin").click(rechercherMedecin);
    $("#btEnregistrerMedecin").click(enregistrerMedecin);
    $("#btSupprimerMedecin").click(supprimerMedecin);

    $
    $("#btOuiConfirmCreationMedecin").click(ouiConfirmCreationMedecin);
    $
    $("#btNonConfirmCreationMedecin").click(nonConfirmCreationMedecin);

    $("#lblMessageGeneral").html("");
} /// initMedecinCRUD

/* Fonction de validation des champs */
function rechercherMedecin() {

    var resultat = true;

    remiseAZeroDesMessages();

    //On verifie que tous les champs ont bien ete remplis
    resultat = verificationSaisiesMedecin("recherche");

    if (resultat) {
        remiseAZeroDesMessages();

        $("#lblMessage").html("Les informations saisies sont
correctes !");

        /*
         * Recuperation des saisies
         */
        var nomMedecin = $("#nomMedecin").val();
        var prenomMedecin = $("#prenomMedecin").val();
        var emailMedecin = $("#emailMedecin").val();
        var rppsMedecin = $("#rppsMedecin").val();

        // Attention passage en mode GET
        var jqXHR = $.get
            (
                "../controls/MedecinCTRL.php",

```

```

        {nomMedecin: nomMedecin, prenomMedecin:
prenomMedecin, emailMedecin: emailMedecin, rppsMedecin:
rppsMedecin, action: "selectOne"},
        function(donnees) {
            // Pourquoi ? D'où vient le Retour chariot ?
            // trim() : seulement depuis JS v ???
            donnees = donnees.trim();
            console.log("Code retour recherche : *" +
donnees + "*");
            if (donnees === "0") {
                $("#lblMessageGeneral").html("Médecin
introuvable !");

                $
                ($("#divConfirmCreationMedecin").css("display", "block");
                }
                else {
                    $("#lblMessageGeneral").html("Le médecin
existe dans la BD");
                }

            },

            "text"
        );
        jqXHR.error(function() {
            var sTexte = jqXHR.status + ":" + jqXHR.statusText;
            $("#lblMessageGeneral").html(sTexte);
        });

    }
    else {
        $("#lblMessage").html("Les informations saisies ne sont pas
correctes !");
    }
} /// rechercherMedecin

function enregistrerMedecin() {

    remiseAZeroDesMessages();

    var resultat = verificationSaisiesMedecin("enregistrement");

    if (resultat) {

        /*
        *
        * Recuperation des saisies
        */
        var nomMedecin = $("#nomMedecin").val();
        var prenomMedecin = $("#prenomMedecin").val();
        var emailMedecin = $("#emailMedecin").val();
    }
}

```

```

var rppsMedecin = $("#rppsMedecin").val();
var idSpecialite = $("#idSpecialite").val();

// Attention passage en mode POST
var jqXHR = $.post
(
    // "../controls/MedecinCTRLSimplifie.php",
    "../controls/MedecinCTRL.php",
    {nomMedecin: nomMedecin, prenomMedecin:
prenomMedecin, emailMedecin: emailMedecin, rppsMedecin:
rppsMedecin, idSpecialite: idSpecialite, action: "insert"},
    function(donnees) {
        donnees = donnees.trim();
        console.log("Code retour ajout : *" + donnees +
"*");

        var lsMessage = "";
        if (donnees == 1) {
            lsMessage = "Un médecin a été ajouté";
            $("#btEnregistrerMedecin").css("display",
"none");

        } else {
            if (donnees == 0) {
                lsMessage = "Aucun médecin n'a été
ajouté";

            }
            else {
                if (donnees == -1) {
                    lsMessage = "Problème d'ajout dans
la BD";

                }
            }
        }
        $("#lblMessageGeneral").html(lsMessage);

    },
    "text"
);
jqXHR.error(function() {
    var sTexte = jqXHR.status + ":" + jqXHR.statusText;
    $("#lblMessageGeneral").html(sTexte);
});
}
} /// enregistrerMedecin

function supprimerMedecin() {

    remiseAZeroDesMessages();

    var resultat = verificationSaisiesMedecin("suppression");

```

```

    if (resultat) {
        /*
         * Recuperation des saisies
         */
        var nomMedecin = $("#nomMedecin").val();
        var prenomMedecin = $("#prenomMedecin").val();
        var emailMedecin = $("#emailMedecin").val();
        var rppsMedecin = $("#rppsMedecin").val();
        var idSpecialite = $("#idSpecialite").val();

        console.log("idSpecialite : " + idSpecialite);

        // Attention passage en mode POST
        var jqXHR = $.post
        (
            "../controls/MedecinCTRL.php",
            {nomMedecin: nomMedecin, prenomMedecin:
prenomMedecin, emailMedecin: emailMedecin, rppsMedecin:
rppsMedecin, idSpecialite: idSpecialite, action: "delete"},
            function(donnees) {
                donnees = donnees.trim();
                console.log("Code retour suppression : *" +
donnees + "*");
                $("#lblMessageGeneral").html(donnees);
            },
            "text"
        );
        jqXHR.error(function() {
            var sTexte = jqXHR.status + ":" + jqXHR.statusText;
            $("#lblMessageGeneral").html(sTexte);
        });
    }
} /// supprimerMedecin

function verificationSaisiesMedecin(action) {
    /*
     * Dans tous les cas (recherche, enregistrement, suppression)
     il faut : EMAIL et RPPS
     * Dans les cas de l'ajout et de la recherche il faut aussi NOM
     et PRENOM
     */
    /*
     */

    $("#lblMessageGeneral").html("");

    // Init du resultat
    var resultat = true;

    /*
     * Expressions regulieres
     */

```

```

// 1 ou 2 chiffres
var motifId = "[0-9]{1,2}$";
var erId = new RegExp(motifId);
// Alpha
var motifAlpha = "[A-Za-zàâäéèëïîôöùû -']{2,}$";
var erAlpha = new RegExp(motifAlpha);

// var reTelephone = /^0[1-9][0-9]{8}$/;
var motifEmail = "[a-z0-9._-]+@[a-z0-9._-]{2,}\\.[a-z]{2,4}$";
var erEmail = new RegExp(motifEmail);

// 6 a 10 caracteres, au moins 1 chiffre, au moins 1 minus, au
moins 1 majus
var motifMDP = "^(?=.*\\d)(?=.*[a-z])(?=.*[A-Z]).{6,10}$";
var erMDP = new RegExp(motifMDP);

// RPPS : 11 chiffres
var motifRPPS = "[0-9]{11}$";
var erRPPS = new RegExp(motifRPPS);

// Ajout ou recherche
// Tout sauf ID
if (action === "enregistrement" || action === "recherche") {

    /*
     * On verifie que tous les champs ont correctement ete
remplis
    */
    if ($("#nomMedecin").val().trim().length < 2 && !
erAlpha.test($("#nomMedecin").val())) {
        $("#messageNomMedecin").html("Le nom doit être saisi (2
caractères au minimum)");
        $("#spanEtoileNomMedecin").css("color", "red");
        resultat = false;
    }
    if ($("#prenomMedecin").val().trim().length < 2 && !
erAlpha.test($("#prenomMedecin").val())) {
        $("#messagePrenomMedecin").html("Le prénom doit être
saisi (2 caractères au minimum)");
        $("#spanEtoilePrenomMedecin").css("color", "red");
        resultat = false;
    }
}

/*
 * DANS TOUS LES CAS : EMAIL + RPPS
 */
/*
 * On verifie que tous les champs ont correctement ete remplis
 */
// On verifie l'expression reguliere

```



```

        if (!erEmail.test($("#emailMedecin").val()) || $
("#emailMedecin").val().trim().length === 0) {
            $('#spanEtoileEmailMedecin').css("color", "red");
            $("#messageEmailMedecin").html("L'e-mail doit être
valide");
            resultat = false;
        }

        if (!erRPPS.test($("#rppsMedecin").val()) || $
("#rppsMedecin").val().trim().length === 0) {
            $('#spanEtoileRppsMedecin').css("color", "red");
            $("#messageRppsMedecin").html("Le RPPS doit être valide");
            resultat = false;
        }

        return resultat;
    } /// verificationSaisiesMedecin

function remiseAZeroDesMessages() {
    /*
     * Remise "a zero"
     */
    $('#spanEtoileNomMedecin').css("color", "black");
    $('#spanEtoilePrenomMedecin').css("color", "black");
    $('#spanEtoileEmailMedecin').css("color", "black");
    $('#spanEtoileRPPSMedecin').css("color", "black");
    $('#spanEtoileIdSpecialite').css("color", "black");

    $('#messageNomMedecin').html("");
    $('#messagePrenomMedecin').html("");
    $('#messageTelephoneMedecin').html("");
    $('#messageEmailMedecin').html("");
    $('#messageRPPSMedecin').html("");
    $('#messageIdSpecialite').html("");
} /// remiseAZeroDesMessages

function ouiConfirmCreationMedecin() {
    $('#lblMessageGeneral').html("");
    $("#divConfirmCreationMedecin").css("display", "none");
    $("#btEnregistrerMedecin").css("display", "block");
}

function nonConfirmCreationMedecin() {
    $("#divConfirmCreationMedecin").css("display", "none");
}

/*

```

```
| *  
| */  
| $(document).ready(initMedecinCRUD);
```

6.3.5 - Exemples de code PHP

6.3.5.1 - Boundary

```

<!DOCTYPE html>
<!--
MedecinLister.php
-->
<html>
    <head>
        <meta charset="UTF-8">
        <link rel="stylesheet"
href="../DataTables/media/css/jquery.dataTables.css">
        <link rel="stylesheet" href="../css/chb.css">
        <title>MedecinLister</title>
    </head>

    <body>
        <?php
        //
        require_once '../daos/Connexion.php';
        require_once '../entities/Medecin.class.php';
        require_once '../daos/MedecinDAO.php';

        $lcnx =
Connexion::getConnexion("../ressources/bd.properties");
        $tMedecins = MedecinDAO::selectAll($lcnx);
        Connexion::seDeconnecter($lcnx);
        ?>
        <table id="tableDesMedecins">
            <thead>
                <tr>
                    <th>ID</th>
                    <th>Nom</th>
                    <th>Prénom</th>
                    <th>E-mail</th>
                    <th>RPPS</th>
                </tr>
            </thead>
            <tbody>
                <?php foreach ($tMedecins as $medecin) { ?>
                    <tr>
                        <td><?php echo $medecin->getIdMedecin(); ?
></td>
                        <td><?php echo $medecin->getNomMedecin(); ?
></td>
                        <td><?php echo $medecin-
>getPrenomMedecin(); ?></td>

```

```

        <td><?php echo $medecin->getEmailMedecin();
?></td>
        <td><?php echo $medecin-
>getRppsMedecin(); ?></td>
    </tr>
    <?php } ?>
</tbody>
</table>

<br>
<a href="../../../controls/NavCTRL.php?
redirection=MedecinRecherche">Retour au CUD !</a>
<br>
<a href="../../../controls/NavCTRL.php?
redirection=Accueil">Retour à l'accueil !</a>

<script src="../../../jquery/jquery.js"></script>
<script
src="../../../DataTables/media/js/jquery.dataTables.js"></script>
<script>
    $(document).ready(function() {
        $('#tableDesMedecins').dataTable();
    });
</script>
</body>
</html>

```

6.3.5.2 - Contrôleurs

```
<?php

/*
 * DeconnexionCTRL.php
 *
 * Pas une classe !!!
 *
 * Script pour la deconnexion via AJAX
 */
session_start();

// On met a "0" la variable de session
$_SESSION["connecte"] = "0";
$_SESSION["profil"] = "0";
echo "Déconnecté(e)";

header("location: ../boundaries/Accueil.php");
?>
```

```

<?php

/*
 * ConnexionCTRL.php
 *
 * Script pour l'authentification via AJAX
 *
 * Du procedural !!!
 *
 */
session_start();

require_once '../daos/Connexion.php';
require_once '../entities/Utilisateur.class.php';
require_once '../daos/UtilisateurDAO.php';

/*
 * SELECTONE BY pseudo, mdp
 */

/*
 * CONNEXION
 */

$lcnx = Connexion::getConnexion("../ressources/bd.properties");

// Recuperation des valeurs transmises
$pseudo = filter_input(INPUT_POST, "pseudoConnexion");
$mdp = filter_input(INPUT_POST, "mdpConnexion");

$ut = UtilisateurDAO::selectOneByPseudoMdp($lcnx, $pseudo, $mdp);

/*
 * SI connexion KO
 * On met a "0" la variable de session nommee "connecte"
 * On met a "NUL" la variable de session nommee "profil"
 */
if ($ut->getNomUtilisateur() === null) {
    $lsMessage = "Connexion ratée";
    $_SESSION["connecte"] = "0";
    $_SESSION["profil"] = "NUL";
} else {
    $lsMessage = "Connecté(e)";
    // A modifier en faisant une requete SQL vers profil
    utilisateur ?
    // 1 : ADM
    // 2 : USER
    // 3 : Anonymous
    $_SESSION["connecte"] = "1";
    if ($ut->getIdProfilUtilisateur() == 1) {
        $_SESSION["profil"] = "ADM";
    } elseif ($ut->getIdProfilUtilisateur() == 2) {

```

```
        $_SESSION["profil"] = "USER";
    }
}
echo $lsMessage;
?>
```

|

6.3.5.3 - Entity

```
<?php

/*
 * Specialite.php
 */

class Specialite {

    private $idSpecialite;
    private $libelleSpecialite;

    function __construct($idSpecialite = "", $libelleSpecialite =
    "") {
        $this->idSpecialite = $idSpecialite;
        $this->libelleSpecialite = $libelleSpecialite;
    }

    public function getIdSpecialite() {
        return $this->idSpecialite;
    }

    public function getLibelleSpecialite() {
        return $this->libelleSpecialite;
    }

    public function setIdSpecialite($idSpecialite) {
        $this->idSpecialite = $idSpecialite;
    }

    public function setLibelleSpecialite($libelleSpecialite) {
        $this->libelleSpecialite = $libelleSpecialite;
    }

}
```

6.3.5.4 - Connexion

```

<?php

class Connexion {

    /**
     *
     * @param type $psCheminParametresConnexion
     * @return type
     */
    public static function
    getConnexion($psCheminParametresConnexion) {

        $tProprietes =
        parse_ini_file($psCheminParametresConnexion);

        $lsProtocole = $tProprietes["protocole"];
        $lsServeur = $tProprietes["serveur"];
        $lsPort = $tProprietes["port"];
        $lsUT = $tProprietes["ut"];
        $lsMDP = $tProprietes["mdp"];
        $lsBD = $tProprietes["bd"];

        /*
         * Connexion
         */
        $lcnx = null;
        try {
            $lcnx = new
            PDO("$lsProtocole:host=$lsServeur;port=$lsPort;dbname=$lsBD;",
            $lsUT, $lsMDP);
            $lcnx->setAttribute(PDO::ATTR_ERRMODE,
            PDO::ERRMODE_EXCEPTION);
            $lcnx->setAttribute(PDO::ATTR_AUTOCOMMIT, FALSE);
            $lcnx->exec("SET NAMES 'UTF8'");
        } catch (Exception $ex) {
            $lcnx = null;
            echo $ex->getMessage();
        }

        return $lcnx;
    }

    /**
     *
     * @param PDO $pcnx
     */
    public static function seDeconnecter(PDO &$pcnx) {
        $pcnx = null;
    }
}

```

```
    }

    /**
     *
     * @param PDO $pcnx
     */
    public static function initialiserTransaction(PDO &$pcnx) {
        if ($pcnx != null) {
            // Si aucune transaction n'est active on initialise une
transaction
            if (!$pcnx->inTransaction()) {
                $pcnx->beginTransaction();
            }
        }
    }

    /**
     *
     * @param PDO $pcnx
     */
    public static function validerTransaction(PDO &$pcnx) {
        // Il faudrait un try/catch et une levee d'exception !!!
        if ($pcnx != null) {
            $pcnx->commit();
        }
    }

    /**
     *
     * @param PDO $pcnx
     */
    public static function annulerTransaction(PDO &$pcnx) {
        if ($pcnx != null) {
            $pcnx->rollBack();
        }
    }
}
```

6.3.5.5 - Properties

```
-- bd.properties
[section_connexion]
protocole=mysql
serveur=127.0.0.1
port=3306
bd=annuaire_medecins_pb
ut=root
mdp=
```

6.3.5.6 - DAO

```

<?php

/**
 * SpecialiteDAO.php
 */
require_once '../entities/Specialite.class.php';

class SpecialiteDAO {

    /**
     *
     * @param PDO $pcnx
     * @param type $id
     * @return \Specialite
     */
    public static function selectOne(PDO $pcnx, $id) {
        $lbOK = true;
        $specialite = new Specialite();

        try {
            /*
             * Execution de la requete
             */
            $lsSelectOne = "SELECT * FROM specialite WHERE
id_specialite=?";
            $lrs = $pcnx->prepare($lsSelectOne);

            $lrs->execute(array($id));
            /*
             * Le resultat
             */
            $lrs->setFetchMode(PDO::FETCH_NUM);
            $enr = $lrs->fetch();

            if ($enr[1] === null) {
                $lbOK = false;
                $specialite->setIdSpecialite(trim("0"));
                $specialite->setLibelleSpecialite("INTROUVABLE");
            } else {
                $specialite->setIdSpecialite($enr[0]);
                $specialite->setLibelleSpecialite($enr[1]);
                ;
            }

            $lrs->closeCursor();
        } catch (PDOException $e) {
            $lbOK = false;
            $specialite->setIdSpecialite("-1");
        }
    }
}

```

```

        $specialite->setLibelleSpecialite("Erreur PDO");
    }

    return $specialite;
}

/**
 *
 * @param PDO $pcnx
 * @return \Specialite
 */
public static function selectAll(PDO $pcnx) {
    $lbOK = true;
    $tSpecialites = array();

    try {
        /*
         * Execution de la requete
         */
        $lsSelect = "SELECT * FROM specialite";
        // $lsSelect = "CALL medecinSelectAll()";
        $lrs = $pcnx->prepare($lsSelect);

        $lrs->execute();
        /*
         * On fait une boucle sur le resultat
         */
        $lrs->setFetchMode(PDO::FETCH_NUM);

        while ($enr = $lrs->fetch()) {
            $specialite = new Specialite();

            $specialite->setIdSpecialite($enr[0]);
            $specialite->setLibelleSpecialite($enr[1]);

            $tSpecialites[] = $specialite;
        }

        $lrs->closeCursor();
    } catch (PDOException $e) {
        $lbOK = false;

        $specialite = new Specialite();

        $specialite->setIdSpecialite(-1);
        $specialite->setLibelleSpecialite($e->getMessage());

        $tSpecialites[] = $specialite;
    }
    return $tSpecialites;
}

/**

```

```

*
* @param PDO $pcnx
* @param Specialite $objet
* @return boolean
*/
public static function insert(PDO $pcnx, Specialite $objet) {
    $lbOK = true;

    try {
        if (!$pcnx->inTransaction()) {
            $pcnx->beginTransaction();
        }
        $lsSQL = "INSERT INTO specialite(libelle_specialite)
VALUES(?) ";
        $lcmd = $pcnx->prepare($lsSQL);

        $t = array();
        array_push($t, $objet->getLibelleSpecialite());

        $lbOK = $lcmd->execute($t);

        $liAffectes = $lcmd->rowCount();
        if ($liAffectes === 1) {
            $lbOK = 1;
        } else {
            $lbOK = -1;
        }

        $pcnx->commit();
    } catch (PDOException $e) {
        $lbOK = false;
    }
    return $lbOK;
}

/**
*
* @param PDO $pcnx
* @param Specialite $objet
* @return boolean
*/
public static function delete(PDO $pcnx, Specialite $objet) {
    $lbOK = true;

    try {
        if (!$pcnx->inTransaction()) {
            $pcnx->beginTransaction();
        }
        /*
        * Execution de la requete
        */
        $lsSQL = "DELETE FROM specialite WHERE id_specialite
= ?";

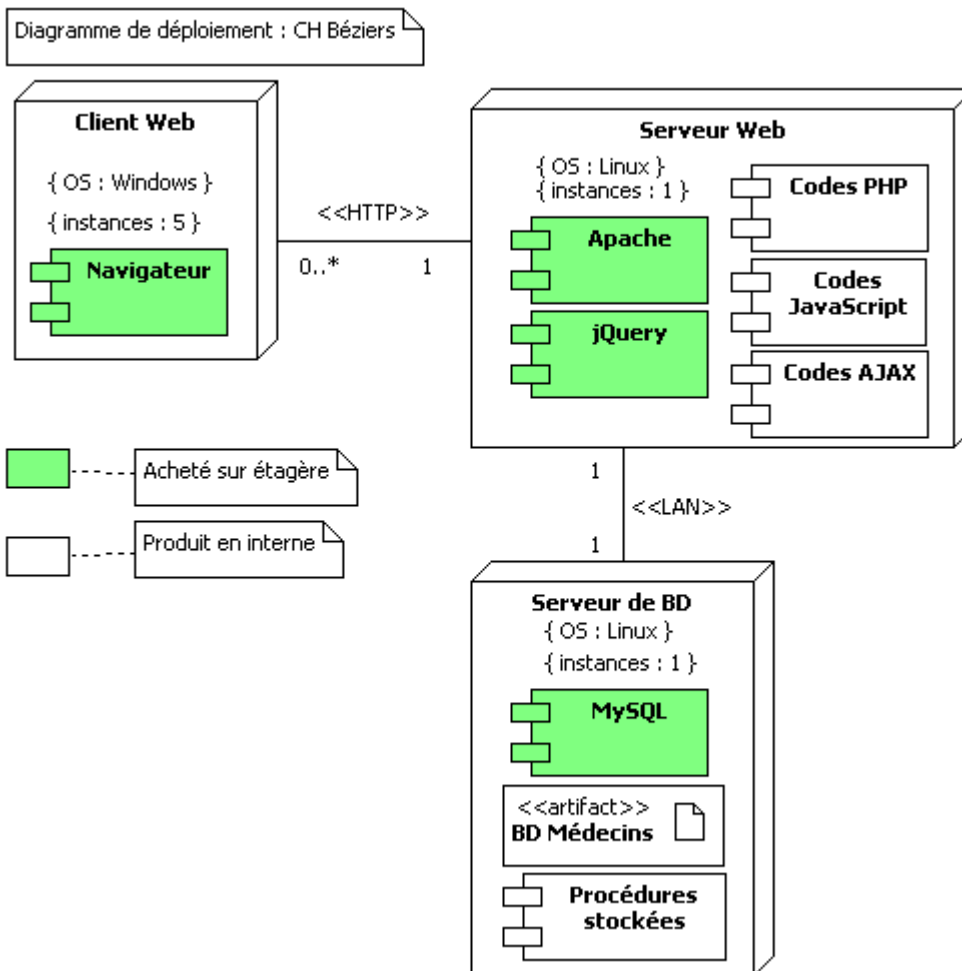
```

```
        $lcmd = $pcnx->prepare($lsSQL);

        $lbOK = $lcmd->execute(array($objet->getIdSpecialite()));
    /*
     * Exploitation du resultat
     */
    $liAffectes = $lcmd->rowCount();
    if ($liAffectes === 1) {
        $lbOK = 1;
    } else {
        $lbOK = -1;
    }
    $pcnx->commit();
} catch (PDOException $e) {
    $lbOK = false;
}
return $lbOK;
}
}
```


Chapitre 7 - *DÉPLOIEMENT*

7.1 - LE DIAGRAMME DE DÉPLOIEMENT



7.2 - LE DÉPLOIEMENT

De la BD :

exécution du code LDD (création de la BD, création des tables, création des procédures stockées),
création des profils.

De l'application :

copie des codes du site Web (HTML, CSS, JavaScript, AJAX, PHP, fichier .properties, ...) sur le serveur,
installation de jQuery.

Chapitre 8 - *FORMATION DES UTILISATEURS*

Chapitre 9 - *CONCLUSION*

Chapitre 10 - *ANNEXES*

10.1 - CORRESPONDANCES PROJET / REAC

Activité	Compétence	Correspondance
Développer des composants d'interface		
	Maquetter une application	UML (DCU, Maquettes, DSEQ, DNAV, DAC, DET) AGL StarUML
	Développer une interface utilisateur	
	Développer des composants d'accès aux données	Langage serveur (PHP)
	Développer des pages web en lien avec une base de données	HTML, CSS, JavaScript, AJAX. Langage serveur (PHP)
Développer la persistance des données		
	Concevoir une base de données (Schéma entité-Association, modèle physique normalisé)	UML (DCL) et/ou Merise (MCD, MLD, MPD) AGL jMerise
	Mettre en place une base de données (intégrité des données, ...)	SQL côté administrateur Création de la BD, création des tables, mise en place des contraintes (FK)
	Développer des composants dans le langage d'une base de données	Procédures stockées avec P/SQL le langage procédural de MySQL
	Utiliser l'anglais dans son activité professionnelle en informatique (Ecrit et parlé)	Anglais
Développer une application n-tiers		
	Concevoir une application	UML
	Collaborer à la gestion d'un projet informatique	Gestion de projet.
	Développer des composants métier	Classes « Entities »
	Construire une application	ECB (Diagrammes de

	organisée en couches	séquence détaillés)
	Développer une application de mobilité numérique	
	Préparer et exécuter les plans de tests d'une application	
	Préparer et exécuter le déploiement d'une application	Diagramme de déploiement (UML) Déploiement. Installation de la BD. Installation des codes HTML, CSS et JavaScript. Installation des codes PHP.

10.2 - OUTILS UTILISÉS

JMerise

Logiciel dédié permettant la modélisation des modèles conceptuels de données, la généralisation et la spécialisation des entités. Il génère le MLD (Modèle Logique de Données) ainsi que le script MySQL à partir du MCD (Modèle Conceptuel de Données)

StarUML

Logiciel de modélisation UML (Unified Modeling Language), open source, qui génère des diagrammes spécifiés dans la norme UML2.0.

XAMPP

Environnement de développement, open source, de distribution Apache, contenant un ensemble complet de services (phpMyAdmin, MySQL, PHP et PERL,...) pour faire tourner un serveur Web ou créer un serveur de développement.

Netbeans

Environnement de Développement Intégré (EDI), open source. Il supporte plusieurs langages (PHP, JAVA, JavaScript, XML, HTML, CSS,...) et outils de collaboration.

NotePad++

Editeur léger de texte générique qui intègre la coloration syntaxique de code sources pour différents langages (PHP, JavaScript, PHP, HTML,...).

10.3 - WEBOGRAPHIE ET BIBLIOGRAPHIE

WEBOGRAPHIE

w3schools.com
w3.org
php.net
tutoelephorm.com
video2brain.com
starkoverflow.com
jquery.com
sql.sh
toutjavascript.com
api.jquery.com
html5-css3.fr
datatables.net
developpez.net
openclassrooms.com

BIBLIOGRAPHIE

UML2 « Modéliser une application Web » - Editions Eyrolles - Pascal Roques
PHP5 - Editions Eyrolles - Stéphane Mariel
PHP/MySQL et JavaScript - Editions Eyrolles - Philippe Chaléat
JAVASCRIPT - Editions ENI – Emmanuel Gutierrez
JAVASCRIPT - Editions Eyrolles – Thierry Templier

10.4 - GLOSSAIRE/LEXIQUE OU/ET LISTE DE MOTS-CLÉS ET SIGLES

10.5 - CODE DE CRÉATION DE LA BD

```
--  
-- Structure de la table `medecin`  
--  
  
CREATE TABLE IF NOT EXISTS `medecin` (  
  `id_medecin` int(11) NOT NULL,  
  `nom_medecin` varchar(50) DEFAULT NULL,  
  `prenom_medecin` varchar(50) DEFAULT NULL,  
  `email_medecin` varchar(100) DEFAULT NULL,  
  `rpps_medecin` bigint(20) DEFAULT NULL,  
  `id_specialite` int(11) DEFAULT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=7 DEFAULT CHARSET=utf8;  
  
--  
-- Structure de la table `profil`  
--  
  
CREATE TABLE IF NOT EXISTS `profil` (  
  `id_profil` int(11) NOT NULL,  
  `code_profil` int(11) DEFAULT NULL,  
  `libelle_profil` varchar(15) DEFAULT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=3 DEFAULT CHARSET=utf8;  
  
--  
-- Structure de la table `specialite`  
--  
  
CREATE TABLE IF NOT EXISTS `specialite` (  
  `id_specialite` int(11) NOT NULL,  
  `libelle_specialite` varchar(100) DEFAULT NULL  
) ENGINE=InnoDB AUTO_INCREMENT=4 DEFAULT CHARSET=utf8;  
  
--  
-- Structure de la table `utilisateur`  
--  
  
CREATE TABLE IF NOT EXISTS `utilisateur` (  
  `id_utilisateur` int(11) NOT NULL,  
  `nom_utilisateur` varchar(50) DEFAULT NULL,  
  `prenom_utilisateur` varchar(50) DEFAULT NULL,  
  `pseudo_utilisateur` varchar(50) DEFAULT NULL,  
  `mdp_utilisateur` varchar(32) DEFAULT NULL,  
  `email_utilisateur` varchar(100) DEFAULT NULL,  
  `id_profil` int(11) DEFAULT NULL  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
-- Index pour les tables exportées
--
--
-- Index pour la table `medecin`
--
ALTER TABLE `medecin`
  ADD PRIMARY KEY (`id_medecin`), ADD KEY `FK_medecin_id_specialite`
  (`id_specialite`);

--
-- Index pour la table `profil`
--
ALTER TABLE `profil`
  ADD PRIMARY KEY (`id_profil`);

--
-- Index pour la table `specialite`
--
ALTER TABLE `specialite`
  ADD PRIMARY KEY (`id_specialite`);

--
-- Index pour la table `utilisateur`
--
ALTER TABLE `utilisateur`
  ADD PRIMARY KEY (`id_utilisateur`), ADD KEY `FK_utilisateur_id_profil`
  (`id_profil`);

--
-- AUTO_INCREMENT pour les tables exportées
--
--
-- AUTO_INCREMENT pour la table `medecin`
--
ALTER TABLE `medecin`
  MODIFY `id_medecin` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=7;
--
-- AUTO_INCREMENT pour la table `profil`
--
ALTER TABLE `profil`
  MODIFY `id_profil` int(11) NOT NULL AUTO_INCREMENT,AUTO_INCREMENT=3;
--
-- AUTO_INCREMENT pour la table `specialite`
--
ALTER TABLE `specialite`
  MODIFY `id_specialite` int(11) NOT NULL
  AUTO_INCREMENT,AUTO_INCREMENT=4;
--
-- AUTO_INCREMENT pour la table `utilisateur`
--
ALTER TABLE `utilisateur`
  MODIFY `id_utilisateur` int(11) NOT NULL AUTO_INCREMENT;
```

```
--  
-- Contraintes pour les tables exportées  
--  
  
--  
-- Contraintes pour la table `medecin`  
--  
ALTER TABLE `medecin`  
ADD CONSTRAINT `FK_medecin_id_specialite` FOREIGN KEY (`id_specialite`)  
REFERENCES `specialite` (`id_specialite`);  
  
--  
-- Contraintes pour la table `utilisateur`  
--  
ALTER TABLE `utilisateur`  
ADD CONSTRAINT `FK_utilisateur_id_profil` FOREIGN KEY (`id_profil`)  
REFERENCES `profil` (`id_profil`);
```

10.6 - CODE DES PROCÉDURES STOCKÉES

DELIMITER \$\$

USE annuaire_medecins \$\$

--
-- LES DROPS
--

-- DROPS POUR medecin --

DROP PROCEDURE IF EXISTS annuaire_medecins.medecinSelectAll \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinSelectOne \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinInsert \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinUpdate \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.medecinDelete \$\$

-- DROPS POUR profil --

DROP PROCEDURE IF EXISTS annuaire_medecins.profilSelectAll \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.profilSelectOne \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.profilInsert \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.profilUpdate \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.profilDelete \$\$

-- DROPS POUR specialite --

DROP PROCEDURE IF EXISTS annuaire_medecins.specialiteSelectAll \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.specialiteSelectOne \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.specialiteInsert \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.specialiteUpdate \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.specialiteDelete \$\$

-- DROPS POUR utilisateur --

DROP PROCEDURE IF EXISTS annuaire_medecins.utilisateurSelectAll \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.utilisateurSelectOne \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.utilisateurInsert \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.utilisateurUpdate \$\$
DROP PROCEDURE IF EXISTS annuaire_medecins.utilisateurDelete \$\$

DELIMITER ;

--
-- LES CREATEs
--

DELIMITER \$\$

-- CREATEs POUR medecin --

-- CREATE SelectAll POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinSelectAll()
BEGIN
 SELECT *

```

        FROM annuaire_medecins.medecin ;
END $$

-- CREATE SelectOne POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinSelectOne(paramID_MEDECIN
INT(10) )
BEGIN
    SELECT *
    FROM annuaire_medecins.medecin
    WHERE id_medecin = paramID_MEDECIN ;
END $$

-- CREATE Insert POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinInsert(paramID_MEDECIN
INT(10),paramNOM_MEDECIN VARCHAR(50),paramPRENOM_MEDECIN
VARCHAR(50),paramEMAIL_MEDECIN VARCHAR(100),paramRPPS_MEDECIN
BIGINT(19),paramID_SPECIALITE INT(10))
BEGIN
    INSERT INTO
annuaire_medecins.medecin(id_medecin,nom_medecin,prenom_medecin,email_
medecin,rpps_medecin,id_specialite)

VALUES(paramID_MEDECIN,paramNOM_MEDECIN,paramPRENOM_MEDECIN,para
mEMAIL_MEDECIN,paramRPPS_MEDECIN,paramID_SPECIALITE) ;
END $$

-- CREATE Update POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinUpdate(paramID_MEDECIN
INT(10),paramNOM_MEDECIN VARCHAR(50),paramPRENOM_MEDECIN
VARCHAR(50),paramEMAIL_MEDECIN VARCHAR(100),paramRPPS_MEDECIN
BIGINT(19),paramID_SPECIALITE INT(10))
BEGIN
    UPDATE annuaire_medecins.medecin
    SET nom_medecin = paramNOM_MEDECIN,prenom_medecin =
paramPRENOM_MEDECIN,email_medecin = paramEMAIL_MEDECIN,rpps_medecin
= paramRPPS_MEDECIN,id_specialite = paramID_SPECIALITE
    WHERE id_medecin = paramID_MEDECIN ;
END $$

-- CREATE Delete POUR annuaire_medecins.medecin
CREATE PROCEDURE annuaire_medecins.medecinDelete(paramID_MEDECIN
INT(10) )
BEGIN
    DELETE
    FROM annuaire_medecins.medecin
    WHERE id_medecin = paramID_MEDECIN ;
END $$

-- CREATES POUR profil --

-- CREATE SelectAll POUR annuaire_medecins.profil
CREATE PROCEDURE annuaire_medecins.profilSelectAll()
```



```

BEGIN
    SELECT *
    FROM annuaire_medecins.profil ;
END $$

-- CREATE SelectOne POUR annuaire_medecins.profil
CREATE PROCEDURE annuaire_medecins.profilSelectOne(paramID_PROFIL INT(10)
)
BEGIN
    SELECT *
    FROM annuaire_medecins.profil
    WHERE id_profil = paramID_PROFIL ;
END $$

-- CREATE Insert POUR annuaire_medecins.profil
CREATE PROCEDURE annuaire_medecins.profilInsert(paramID_PROFIL
INT(10),paramCODE_PROFIL INT(10),paramLIBELLE_PROFIL VARCHAR(15))
BEGIN
    INSERT INTO annuaire_medecins.profil(id_profil,code_profil,libelle_profil)
    VALUES(paramID_PROFIL,paramCODE_PROFIL,paramLIBELLE_PROFIL) ;
END $$

-- CREATE Update POUR annuaire_medecins.profil
CREATE PROCEDURE annuaire_medecins.profilUpdate(paramID_PROFIL
INT(10),paramCODE_PROFIL INT(10),paramLIBELLE_PROFIL VARCHAR(15))
BEGIN
    UPDATE annuaire_medecins.profil
    SET code_profil = paramCODE_PROFIL,libelle_profil =
paramLIBELLE_PROFIL
    WHERE id_profil = paramID_PROFIL ;
END $$

-- CREATE Delete POUR annuaire_medecins.profil
CREATE PROCEDURE annuaire_medecins.profilDelete(paramID_PROFIL INT(10) )
BEGIN
    DELETE
    FROM annuaire_medecins.profil
    WHERE id_profil = paramID_PROFIL ;
END $$

-- CREATes POUR specialite --

-- CREATE SelectAll POUR annuaire_medecins.specialite
CREATE PROCEDURE annuaire_medecins.specialiteSelectAll()
BEGIN
    SELECT *
    FROM annuaire_medecins.specialite ;
END $$

-- CREATE SelectOne POUR annuaire_medecins.specialite
CREATE PROCEDURE
annuaire_medecins.specialiteSelectOne(paramID_SPECIALITE INT(10) )

```

```

BEGIN
    SELECT *
    FROM annuaire_medecins.specialite
    WHERE id_specialite = paramID_SPECIALITE ;
END $$

-- CREATE Insert POUR annuaire_medecins.specialite
CREATE PROCEDURE annuaire_medecins.specialiteInsert(paramID_SPECIALITE
INT(10),paramLIBELLE_SPECIALITE VARCHAR(100))
BEGIN
    INSERT INTO annuaire_medecins.specialite(id_specialite,libelle_specialite)
    VALUES(paramID_SPECIALITE,paramLIBELLE_SPECIALITE) ;
END $$

-- CREATE Update POUR annuaire_medecins.specialite
CREATE PROCEDURE annuaire_medecins.specialiteUpdate(paramID_SPECIALITE
INT(10),paramLIBELLE_SPECIALITE VARCHAR(100))
BEGIN
    UPDATE annuaire_medecins.specialite
    SET libelle_specialite = paramLIBELLE_SPECIALITE
    WHERE id_specialite = paramID_SPECIALITE ;
END $$

-- CREATE Delete POUR annuaire_medecins.specialite
CREATE PROCEDURE annuaire_medecins.specialiteDelete(paramID_SPECIALITE
INT(10) )
BEGIN
    DELETE
    FROM annuaire_medecins.specialite
    WHERE id_specialite = paramID_SPECIALITE ;
END $$

-- CREATes POUR utilisateur --

-- CREATE SelectAll POUR annuaire_medecins.utilisateur
CREATE PROCEDURE annuaire_medecins.utilisateurSelectAll()
BEGIN
    SELECT *
    FROM annuaire_medecins.utilisateur ;
END $$

-- CREATE SelectOne POUR annuaire_medecins.utilisateur
CREATE PROCEDURE
annuaire_medecins.utilisateurSelectOne(paramID_UTILISATEUR INT(10) )
BEGIN
    SELECT *
    FROM annuaire_medecins.utilisateur
    WHERE id_utilisateur = paramID_UTILISATEUR ;
END $$

-- CREATE Insert POUR annuaire_medecins.utilisateur

```

```
CREATE PROCEDURE annuaire_medecins.utilisateurInsert(paramID_UTILISATEUR
INT(10),paramNOM_UTILISATEUR VARCHAR(50),paramPRENOM_UTILISATEUR
VARCHAR(50),paramPSEUDO_UTILISATEUR
VARCHAR(32),paramMDP_UTILISATEUR VARCHAR(32),paramEMAIL_UTILISATEUR
VARCHAR(100),paramID_PROFIL INT(10))
BEGIN
    INSERT INTO
annuaire_medecins.utilisateur(id_utilisateur,nom_utilisateur,prenom_utilisateur,p
seudo_utilisateur,mdp_utilisateur,email_utilisateur,id_profil)

VALUES(paramID_UTILISATEUR,paramNOM_UTILISATEUR,paramPRENOM_UTILISAT
EUR,paramPSEUDO_UTILISATEUR,paramMDP_UTILISATEUR,paramEMAIL_UTILISAT
EUR,paramID_PROFIL) ;
END $$

-- CREATE Update POUR annuaire_medecins.utilisateur
CREATE PROCEDURE annuaire_medecins.utilisateurUpdate(paramID_UTILISATEUR
INT(10),paramNOM_UTILISATEUR VARCHAR(50),paramPRENOM_UTILISATEUR
VARCHAR(50),paramPSEUDO_UTILISATEUR
VARCHAR(32),paramMDP_UTILISATEUR VARCHAR(32),paramEMAIL_UTILISATEUR
VARCHAR(100),paramID_PROFIL INT(10))
BEGIN
    UPDATE annuaire_medecins.utilisateur
    SET nom_utilisateur = paramNOM_UTILISATEUR,prenom_utilisateur =
paramPRENOM_UTILISATEUR,pseudo_utilisateur =
paramPSEUDO_UTILISATEUR,mdp_utilisateur =
paramMDP_UTILISATEUR,email_utilisateur = paramEMAIL_UTILISATEUR,id_profil =
paramID_PROFIL
    WHERE id_utilisateur = paramID_UTILISATEUR ;
END $$

-- CREATE Delete POUR annuaire_medecins.utilisateur
CREATE PROCEDURE annuaire_medecins.utilisateurDelete(paramID_UTILISATEUR
INT(10) )
BEGIN
    DELETE
    FROM annuaire_medecins.utilisateur
    WHERE id_utilisateur = paramID_UTILISATEUR ;
END $$

DELIMITER ;
```