

TABLE DES MATIERES

DEDICACE.....	3
REMERCIEMENTS	4
LISTE DES ACRONYMES	5
LISTE DES FIGURES.....	6
LISTE DES TABLEAUX.....	7
AVANT-PROPOS	8
INTRODUCTION.....	9

PARTIE I : PRESENTATION DE LA STRUCTURE D'ACCUEIL..... 10

I- HISTORIQUE	11
II- MISSIONS.....	11
III- ORGANISATION ET ROLES DES SERVICES	11
IV- ORGANIGRAMME.....	14
V- PRESENTATION DU SERVICE D'ACCUEIL	16

PARTIE II : ETUDE PREALABLE..... 18

I- PRESENTATION DU PROJET LASCOPE	19
1- CONTEXTE DU PROJET	19
2- OBJECTIF DU PROJET.....	19
3- CHOIX DE METHODE D'ANALYSE.....	19
3.1- LES CONCEPTS DE BASE DE L'APPROCHE OBJET	19
a) Objet.....	19
b) Classe	19
c) Héritage.....	20
3.2- LES DIAGRAMMES UML	20
A- DIAGRAMME DE CAS D'UTILISATION.....	21
a) Concepts de base.....	21
b) Représentation du diagramme des cas d'utilisation	22
c) Relations entre cas d'utilisation	22
d) Description textuelle des cas d'utilisation	22
B- DIAGRAMME DE CLASSE	23
a) Le concept d'objet	23
b) Le concept de classe.....	23
c) RELATION ENTRE CLASSES	24
C- DIAGRAMME D'ACTIVITE	25
II- ETUDE DE L'EXISTANT	26
III- CRITIQUE DE L'EXISTANT	26
IV- PROPOSITION DE SOLUTION	27

PARTIE III : ETUDE DETAILLEE (APPLICATION DU LANGAGE DE MODELISATION UML A NOTRE ETUDE)..... 27

I- DIAGRAMME DE CAS D'UTILISATION	28
---	----

1-	PRESENTATION DES ACTEURS	28
2-	IDENTIFICATION DES CAS D'UTILISATION	28
3-	DESCRIPTION TEXTUELLE DES CAS D'UTILISATION	29
4-	DIAGRAMME DE CAS D'UTILISATION GLOBAL	33
II-	DIAGRAMME DE CLASSE	34
III-	DIAGRAMME D'ACTIVITE	34
1-	CAS : AUTHENTIFICATION	35
2-	CAS : SAISIE DE DONNEES	35
3-	CAS : CONSULTATION DE FICHE	36
IV-	PRINCIPE DE FONCTIONEMENT DU MODULE D'EXPORTATION	37
1-	PRESENTATION DE LA TABLE LIS_TAB	38
2-	PRESENTATION ET DESCRIPTION DU SCRIPT	39

PARTIE IV : IMPLEMENTATION ET DEPLOIEMENT DE L'APPLICATION..... 44

I-	ETUDE TECHNIQUE	44
1-	ENVIRONNEMENT MATERIEL	44
2-	ENVIRONNEMENT LOGICIEL	44
3-	DESCRIPTION DES LOGICIELS DE DEVELOPPEMENT UTILISES	45
II-	REALISATION DE L'APPLICATION	46
1-	DESCRIPTION DES INTERFACES DE L'APPLICATION	46
a)	Interface Authentification	46
b)	Interface principale	47
c)	Onglet « sélection des sites »	49
d)	Onglet « gestion des sites »	49
e)	Onglet « Exportation des données »	50
f)	Onglet « ajout nouveau site »	51
	CONCLUSION	52
	WEBOGRAPHIE	53
	SOMMAIRE	54

DEDICACE

Nous dédions ce document à toute notre famille.

REMERCIEMENTS

Nous ne saurons entamer la rédaction de ce document sans toutefois exprimer notre profonde gratitude à de toutes ses personnes avisées qui nous ont été d'une grande aide à quelque niveau que ce soit, personnes sans lesquelles l'élaboration de ce projet n'aurait pas eu lieu.

Nos remerciements vont tout d'abord à l'endroit de la Direction Exécutive du Programme PAC-CI, avec à sa tête le Dr Raoul MOH.

Ensuite, à l'endroit de :

- M. Célestin N'CHO, Chef du Service Informatique et Mme Adrienne ABOUA, Responsable des Ressources Humaines pour nous avoir permis d'intégrer l'entreprise et nous accordé leur confiance.
- M. Cyrille YAO, notre encadreur et M. Elvis AMANI, pour la proposition du sujet, leurs disponibilités, leurs recommandations et leurs précieux conseils.
- M. Dominique YAO, Responsable de la Hotline pour sa patience, son orientation et assistance.

Ainsi qu'à tout l'ensemble du Service Informatique pour leur chaleureux accueil et encouragement.

Nous saisissons cette occasion pour adresser notre reconnaissance, à tous nos parents, tout particulièrement à ma maman Kossia Pauline KOBENAN pour son inlassable soutien, à nos différents professeurs pour leurs instructions et à nos amis et frères.

LISTE DES ACRONYMES

- **UML** : Unified Modeling Language
- **HTML** : Hypertext Make-up Language
- **PHP** : Hypertext Preprocessor
- **SGBD** : Système de Gestion de Base de Données
- **BDD** : Base De Données
- **ODBC** : Open Database Connectivity

LISTE DES FIGURE

FIGURE I.1 : ORGANIGRAMME GENERALE DU PROGRAMMA PAC-CI	15
FIGURE I.2 : ORGANIGRAMME DU SERVICE INFORMATIQUE DU PROGRAMME PAC-CI	17
FIGURE II.1 : CAS D'UTILISATION ET INTERACTION	22
FIGURE II.2 : REPRESENTATION DU DIAGRAMME DE CAS D'UTILISATION	22
FIGURE II.3 : FORMALISME D'UNE CLASSE	24
FIGURE II.4 : REPRESENTATION D'UNE ASSOCIATION	25
FIGURE II.5: RELATION SPECIALISATION/GENERALISATION	25
FIGURE III.1 : DIAGRAMME DU CAS D'UTILISATION GLOBAL	33
FIGURE III.2 : DIAGRAMME DE CLASSE	34
FIGURE III.3 : DIAGRAMME D'ACTIVITE « AUTHENTIFICATION »	35
FIGURE III.4 : DIAGRAMME D'ACTIVITE « SAISIE DE DONNEES »	36
.....	37
FIGURE III.5: DIAGRAMME D'ACTIVITE « CONSULTATION DE FICHES »	37
FIGURE III.6 : STRUCTURE DE LA TABLE LIS_TAB	38
FIGURE III.7: DECHARGEMENT DES DONNEES DE LA TABLE LIS_TAB	38
FIGURE III.8 : VUE D'ENSEMBLE DE LA TABLE DANS LA BDD	39
FIGURE III.9 : CODE DE RECUPERATION DU SITE	39
FIGURE III.10 : CODE DE CREATION A LA CONNEXION ODBC	40
FIGURE III.12 : EXTRAIT DU CODE D'EXPORTATION DES DONNEES.	43
FIGURE IV.1: INTERFACE D'AUTHENTIFICATION	47
FIGURE IV.2: PAGE D'ACCUEIL	48
FIGURE IV.3: INTERFACE « SELECTION DES SITES »	49
FIGURE IV.4: INTERFACE « GESTION DES SITES »	50
FIGURE IV.5: INTERFACE « EXPORTATION DES DONNEES »	51
FIGURE IV.6: INTERFACE « AJOUT NOUVEAU SITE »	52

LISTE DES TABLEAUX

TABLEAU III.1 : IDENTISATION DES CAS D'UTILISATION	29
TABLEAU III.2 : DESCRIPTION TEXTUELLE DE CAS D'UTILISATION « AUTHENTIFICATION »	29
TABLEAU III.3 : DESCRIPTION TEXTUELLE DE CAS D'UTILISATION « SAISIE DE DONNEES »	29
TABLEAU III.4 : DESCRIPTION TEXTUELLE DE CAS D'UTILISATION « CONSULTATION DES FICHES »	30
TABLEAU III.5 : DESCRIPTION TEXTUELLE DE CAS D'UTILISATION « EXPORT DE DONNEES »	30
TABLEAU III.6 : DESCRIPTION TEXTUELLE DE CAS D'UTILISATION « AJOUT D'UN UTILISATEUR »	31
TABLEAU III.7 : DESCRIPTION TEXTUELLE DE CAS D'UTILISATION « MODIFICATION D'UN UTILISATEUR	31
TABLEAU III.8 : DESCRIPTION TEXTUELLE DE CAS D'UTILISATION « SUPPRESSION D'UN UTILISATEUR »	32

AVANT-PROPOS

La licence professionnelle Réseau et Génie Logiciel est un diplôme de niveau BAC + 3 qui confère au détenteur une technicité et une aptitude professionnelle. Elle se déroule en deux phases :

- La première phase est la phase théorique. Elle concerne les cours suivis tout le long de l'année académique.
- La seconde, est la phase pratique. Elle permettra à l'étudiant de mettre en application les cours appris pendant la phase théorique. C'est elle qui oriente les étudiants vers les entreprises afin d'y effectuer un stage pratique D'au moins trois (3) mois.

Après ces 3 mois de stage, l'étudiant devra présenter en présence d'un jury un mémoire de fin de cycle relatif à tout ce qu'il aura appris et fait en entreprise. Le jury à son tour se prononcera sur son admission au diplôme de la Licence Professionnelle Réseau et Génie Logiciel.

C'est dans ce cadre que nous avons effectué un stage au Programme PAC-CI sur un projet dont le thème est le suivant :

« Conception et développement d'un module d'exportation d'une base de données MySQL vers Access : Cas du projet LASCOPE »

INTRODUCTION

L'informatique fait partie intégrante de la plupart des entreprises à tel enseigne que le métier d'informaticien s'est diversifié en une multitude de fonctions spécialisées touchant au réseau, à la sécurité des systèmes informatiques, à l'infographie, etc.

On trouve parmi ces métiers la conception et le développement d'applications qui consistent à analyser un besoin, à modéliser la future solution, à concevoir la base de données et à coder la solution informatique retenue. Après cela, il peut être nécessaire de vouloir faire une exportation des données de la base de données dans un format précis en vue de faire des traitements ultérieurs.

Dans le cadre de notre projet de fin de cycle de la licence professionnelle en Réseaux et Génie Logiciel, nous serons amenés à travailler sur le thème suivant :

<<Conception et développement d'un module d'exportation d'une base de données MySQL vers Access : Cas du projet LASCOPE>>.

Pour la mise en place de ce projet, notre étude se présente comme suit :

- ✓ Présentation de la structure d'accueil
- ✓ Etude préalable
- ✓ Etude détaillée
- ✓ Etude technique

PARTIE I : PRESENTATION DE LA STRUCTURE D'ACCUEIL

I- HISTORIQUE

Le programme « PACCI/site ANRS de Côte d'Ivoire » a été ouvert en 1995, puis formalisé en 1996 par convention entre le Ministère ivoirien de la Santé, le Ministère ivoirien de l'Economie et des Finances, le Ministère français de la Coopération et l'ANRS. Cette convention fondatrice fixait deux objectifs :

- La formation du personnel de santé à la recherche médicale sur le VIH/SIDA ;
- La mise en œuvre de recherches médicales sur le VIH/SIDA dont les résultats devraient rapidement utiles aux personnes affectées par la maladie.

En 2010, les partenaires ivoiriens et français, jugent le bilan positif et consolident le programme PACCI par une révision de la convention élargit les missions du programme PACCI, dont l'objectif est maintenant de développer la recherche scientifique et médicale non seulement sur le VIH/SIDA mais aussi sur les autres maladies infectieuses.

En 2013, le développement institutionnel de PAC-CI s'est poursuivi de façon substantielle. Ainsi une convention de coopération a été signée le 6 novembre 2013 entre les différents partenaires suivants pour créer le tout premier Laboratoire International Associé (LIA-INSERM) en Afrique subsaharienne.

II- MISSIONS

Ses tâches sont basées sur trois thèmes prioritaires de recherche que sont :

- La prévention de la transmission du VIH de la mère à l'enfant,
- L'amélioration de la prise en charge de l'adulte infecté par le VIH,
- L'amélioration de la prise en charge de l'enfant infecté par le VIH.

Sur ceux-ci, les outils de recherche ont été d'abord épidémiologiques, cliniques et biologiques. Ensuite des années après, d'autres disciplines sont venues appuyer les recherches sur ces thèmes, notamment la sociologie, l'anthropologie, l'économie et la modélisation.

III- ORGANISATION ET ROLES DES SERVICES

Le Programme PACCI est composé d'une « équipe centrale » inter-projets, et « d'équipes partenaires ».

● EQUIPE CENTRALE

Hébergée dans les locaux du programme PACCI, dans l'enceinte du CHU de Treichville, elle est sous la coresponsabilité de deux Coordonnateurs (un ivoirien, dit Coordonnateur « Sud », et un français, dit Coordonnateur « Nord »), de deux Coordonnateurs adjoints, d'un Directeur Exécutif et d'un Directeur Exécutif adjoint.

Le Directeur est appuyé dans ses tâches par un comité administratif et une équipe de chercheurs :

❖ **LE COMITE ADMINISTRATIF**

- Un responsable administratif et financier
- Une responsable administrative
- Un responsable logistique
- Un responsable informatique

➤ **Le Département Administratif et Financier**

Ce département est par M. Abdoulaye CISSE, il pilote et participe aux missions, de planification, de gestion et de coordination des tâches administratives financières et comptable de la structure. Il est aidé dans ces tâches par Mlle Madikona DOSSO (Chef comptable). Il a pour rôle principal de :

- Contrôler de la comptabilité de l'entreprise ;
- Gérer la trésorerie ;
- Prévoir le financement des investissements et/ou Projets ;
- Assurer une veille sur l'évolution des lois et réglementations ;
- Superviser les relations avec les services fiscaux ;
- Assurer une permanence juridique et sociale ;
- Participer à l'administration du personnel ;
- Participer à la gestion des systèmes d'informations de l'entreprise.

➤ **Le Département Ressources Humaines**

La gestion des ressources humaines est la fonction organisationnelle du Programme PAC-CI. Elle est présidée par Mme Adrienne ABOUA, qui s'occupe du recrutement, de la gestion, du perfectionnement et de la motivation du personnel. Elle fournit de même du soutien, des systèmes fonctionnels et spécialisés pour favoriser la participation des employés ainsi que des systèmes de gestion pour favoriser le respect réglementaire des normes liées à l'emploi et aux droits du personnel.

➤ **Le Département Logistique et Achats**

Cette section est dirigée par M. Romuald KONAN. Il est épaulé dans l'exécution des tâches qui incombent à la section par une assistante Mlle Sylvie KONAN. Son rôle premier est de mettre à la disposition du personnel les moyens nécessaires pour arriver à travailler dans les conditions optimales. En plus de cela, il assure toutes les tâches qui composent la fonction achat dans une entreprise normale depuis la réception de l'expression des besoins jusqu'à satisfaction du demandeur. En effet il établit les demandes d'achat, lance les commandes, sélectionne-le(s) fournisseur(s) après confrontation des offres, réceptionne les commandes, procède à la vérification avant de les transmettre au service concerné.

➤ **Le Département Informatique et Télécommunication**

Cette section dirigée par M. Célestin N'CHOT, constitue l'un des piliers essentiels sur lequel repose la structure. La performance de son système lui permet d'effectuer l'échange des données informatisées entre toutes les structures de recherche rattachées. La maintenance de ce système est assurée par le service informatique. Cette direction a en charge, le suivi, l'évolution et l'adaptation aux solutions informatiques implémentées par le groupe.

❖ **UNE EQUIPE DE CHERCHEUR**

- Deux chercheurs seniors de l'équipe, les chefs de projet appuyer eux aussi par :
- Quatre médecins moniteurs d'étude clinique
- Un biologiste, moniteur d'étude biologique
- Une équipe spécialisée en Gestion des données
- Cinq gestionnaires de bases de données/développeurs d'applications
- Un responsable technicien d'études clinique
- Une traductrice

● **EQUIPE PARTENAIRE**

➤ **En Côte d'Ivoire**

- **Les trois CHU d'Abidjan** : le CHU de Treichville, le CHU de Yopougon et le CHU de Cocody.
- **Les quatre grands centres de soins extrahospitaliers spécialisés dans le VIH** :
 - Le Centre de Prise en charge et de Formation
 - Le Centre Médical de Suivi des Donneurs de Sang
 - Le Centre Intégré de Recherches Biocliniques d'Abidjan
 - L'Unité de Soins Ambulatoires et de Conseil
- **Des structures de soins générales autres que les CHU** : les hôpitaux généraux de Yopougon et d'Abobo Nord.

➤ **En France**

- **A Bordeaux** : Le Centre Inserm, Institut de Santé Publique Epidémiologie et Développement
- **A Montpellier** : L'Université de Montpellier
- **A Nantes** : L'Hôpital Laennec

➤ **En Afrique**

Burkina Faso: CHU Salado Ouédraogo, CHU Souro Sanou Bobo-Dioulasso

Cameroun : Hôpital Centrale de Yaoundé

Guinée : CHU Donka

Mali : CHU du Point G

Sénégal : le Centre Régional de Recherche et Formation

Ouganda: Mbarara University of Science and Technology

Tanzanie : NIMR

Togo : Le Laboratoire BIOLIM, Université de Lomé, CHU Sylvanus Olympio

➤ **En Asie**

Vietnam: Clinical research unit Pham Ngoc Thach Hospital

Cambodge: National Center for HIV/AIDS

➤ **En Amérique Latine**

Brésil: Laboratory on Clinical research on STD/AIDS – IPEC/FIOCRUZ

IV- ORGANIGRAMME

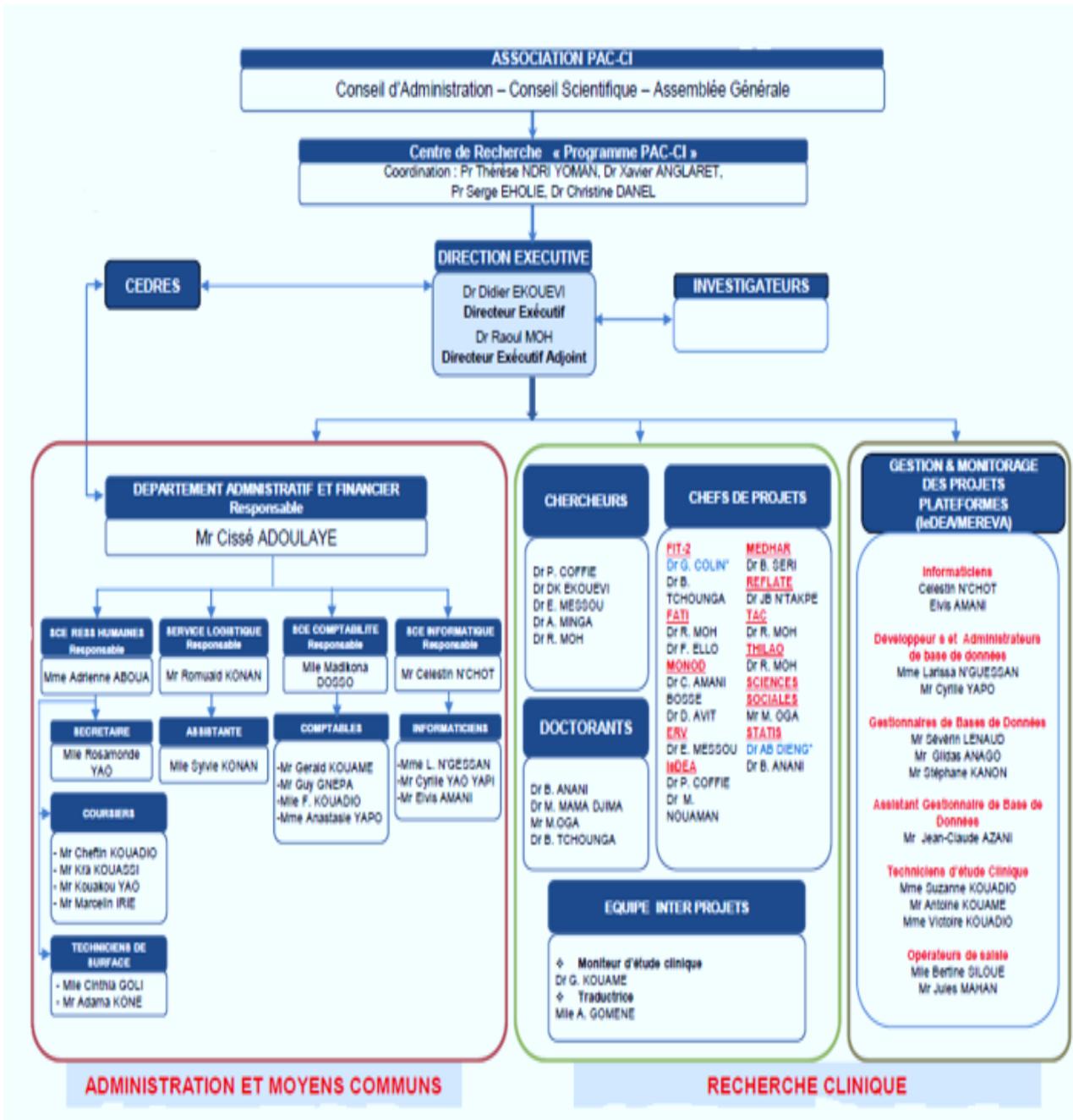


FIGURE I.1 : Organigramme générale du Programme PAC-CI

V- PRESENTATION DU SERVICE D'ACCUEIL

Notre travail a été effectué au sein du service informatique. Ce service est chargé d'automatiser les tâches des différents services pour une gestion plus fine de l'entreprise et aussi permettre aux travailleurs d'être plus efficace dans le travail. Elle est subdivisée comme suite :

- ✓ **Le pole réseaux et sécurité** chargé de :
 - Maintenir le parc informatique, planifier les interventions d'installation, de configuration et de dépannage de matériels,
 - Gérer le réseau informatique et faire évoluer l'infrastructure matérielle dans tous les bâtiments,
 - Gérer la messagerie électronique, le réseau sans fil,
 - Gérer les équipements audiovisuels et les systèmes de visioconférence,
 - Mettre en place les mécanismes concernant la sécurité informatique et assurer la veille sur l'évolution des risques,
 - Administrer et exploiter les serveurs administratifs, etc.

- ✓ **Le pole développement et gestion des bases de données** chargé de :
 - Mettre en place une politique de sauvegarde et d'archivage des données,
 - Maintenir et faire évoluer le système d'information,
 - Mise en place de nouvelles solutions applicatives.

Sa **représentation hiérarchique** est la suivante :

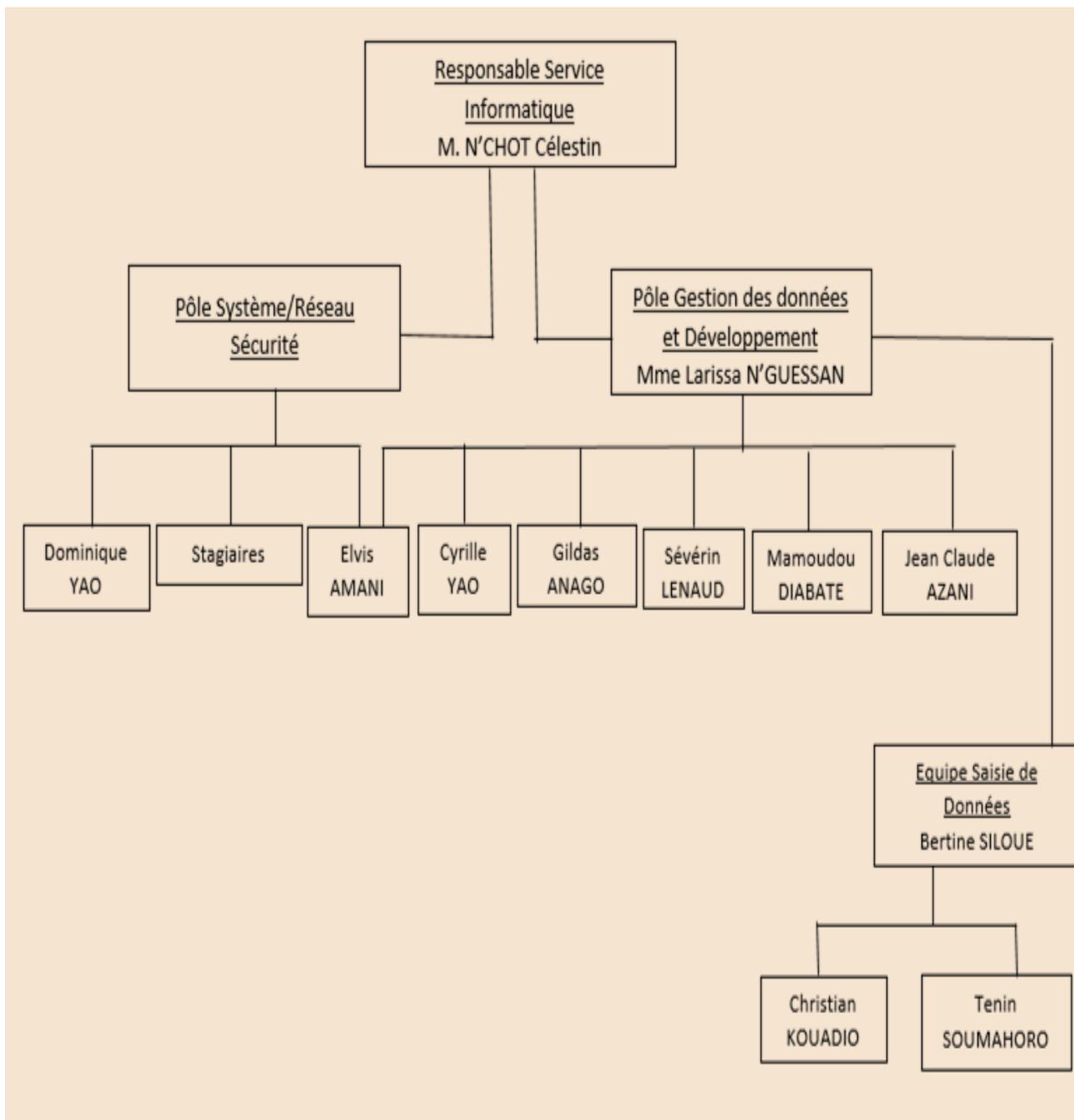


FIGURE I.2 : Organigramme du Service Informatique du Programme PAC-CI

PARTIE II : ETUDE PREALABLE

I- PRESENTATION DU PROJET LASCOPE

1- CONTEXTE DU PROJET

Le projet LASCOPE est une étude prospective observationnelle se déroulant au Nigéria. Cette étude décrit le parcours de la prise en charge des caractéristiques biologiques des patients atteint de la fièvre de Lassa. Compte tenu de l'importance des données à traiter et à analyser dans la cadre de ce projet, la mise en place d'une base de données n'est plus à prouver. Ainsi dans le souci de mener à bien cette activité, une application web PHP/MySQL conçu à partir du Framework Symfony a été développée et déployée afin de permettre la saisie en ligne, le traitement et l'analyse des données pour une bonne organisation du travail. Par ailleurs, cette application devrait aussi permettre de générer des demandes de corrections pour les données mal renseignées ou incohérentes. Toutefois, cette option étant absente à l'heure actuelle dans l'application, il est apparu nécessaire de développer rapidement un module d'exportation de base de données depuis MySQL vers un format ACCESS où il existe un outil déjà développé par le service informatique permettant de faire lesdits demandes de correction.

2- OBJECTIF DU PROJET

L'objectif de ce projet est de mettre en place un module d'exportation de la base de données du projet LASCOPE de MySQL vers ACCESS. Cette base au format ACCESS, servira à :

- générer des demandes de correction afin de procéder aux modifications des données manquantes ou aberrantes.
- réaliser des analyses statistiques

3- CHOIX DE METHODE D'ANALYSE

La méthode qui est utilisée pour la modélisation de l'application du projet LASCOPE est UML (Unified Modeling Language), en français (Langage de Modélisation Unifié). Il s'agit d'un ensemble de notations graphiques s'appuyant sur des diagrammes et permettant de spécifier, visualiser et de documenter les systèmes logiciels orientés-objet. Comme UML est un langage de modélisation objet, nous allons pour la suite voir ce qu'est l'approche orienté objet et ce à quoi elle sert.

3.1- LES CONCEPTS DE BASE DE L'APPROCHE OBJET

a) Objet

Un objet est une abstraction d'un élément du monde réel. Il possède des informations, par exemple nom, prénom, adresse etc., et se comporte suivant un ensemble d'opérations qui lui sont applicables.

De plus, un ensemble d'attributs caractérisent l'état d'un objet, et l'on dispose d'un ensemble d'opérations (les méthodes) qui permettent d'agir sur le comportement de l'objet. Ce dernier est le cœur de cette approche. Tout objet donné possède deux caractéristiques : Son état courant (attributs) et Son comportement (méthodes).

b) Classe

On appelle classe la structure d'un objet, c'est-à-dire la déclaration de l'ensemble des entités qui composeront un objet. Les objets de même nature ont en général la même structure et le même comportement. La classe factorise les caractéristiques communes de ses objets et permet de les classifier. Un objet est l'instance d'une classe, et une classe est un type de données abstrait, caractérisé par des propriétés (ses attributs et ses méthodes) communes à ses objets et un mécanisme permettant de créer des objets ayant ces propriétés. Le concept de classe permet donc de regrouper des objets de même nature.

c) Héritage

L'héritage est un principe propre à la programmation orientée objet, permettant de créer une nouvelle classe à partir d'une classe existante. Le nom d'"héritage" (ou parfois dérivation de classe) provient du fait que la classe dérivée (la classe nouvellement créée) contient les attributs et les méthodes de sa superclasse (la classe dont elle dérive).

L'intérêt majeur de l'héritage est de pouvoir définir de nouveaux attributs et de nouvelles méthodes pour la classe dérivée, qui viennent s'ajouter à ceux et celles héritées. Par ce moyen on crée une hiérarchie de classes de plus en plus spécialisées.

Pour conclure notons que :

- L'approche orientée objet permet de modéliser une application sous la forme d'interactions entre objets.
- Les objets ont des propriétés et peuvent faire des actions.
- Ils masquent la complexité d'une implémentation grâce à l'encapsulation.
- Les objets peuvent hériter de fonctionnalités d'autres objets s'il y a une relation d'héritage entre eux.

3.2- LES DIAGRAMMES UML

- **Les Diagrammes Structurels** : Ils ont comme vocation de représenter l'aspect statique d'un système. Ils permettent d'identifier les objets constituant le programme, leurs attributs, leurs opérations et les méthodes qui leurs sont associés. Ils sont au nombre de six à savoir :
 - Diagramme de Classe ;
 - Diagramme d'objet ;
 - Diagramme de composant ;
 - Diagramme de déploiement ;
 - Diagramme de Paquetage ;
 - Diagramme de structure composite.
- **Les Diagrammes De Comportement** : Ces diagrammes représentent la partie dynamique d'un système réagissant aux événements et permettant de produire les résultats attendus par les utilisateurs. Sept diagrammes sont proposés :
 - Diagramme des cas d'utilisation ;
 - Diagramme d'état-transition ;
 - Diagramme d'activités ;
 - Diagramme de séquence ;
 - Diagramme de communication ;
 - Diagramme global d'interaction ;

- Diagramme de temps.

Dans le cadre de la modélisation, il conviendra pour nous d'utiliser les diagrammes de cas d'utilisation, de classes et d'activités. Ces diagrammes sont utiles pour la maîtrise d'œuvre à qui ils permettent de formaliser les contraintes de la réalisation et la solution technique.

A- DIAGRAMME DE CAS D'UTILISATION

Le diagramme de cas d'utilisation est un moyen permettant de recueillir, d'analyser et d'organiser les besoins des acteurs d'un système, et de recenser les grandes fonctionnalités d'un système. Il s'agit donc de la première étape UML d'analyse d'un système. Un cas d'utilisation permet de décrire l'interaction entre les acteurs (utilisateurs du cas) et le système. La description de l'interaction est réalisée suivant le point de vue de l'utilisateur.

a) Concepts de base

La représentation d'un cas d'utilisation met en jeu trois concepts : l'acteur, le cas d'utilisation et l'interaction entre l'acteur et le cas d'utilisation.

- **Acteur**

Un acteur est un utilisateur type qui a toujours le même comportement vis-à-vis d'un cas d'utilisation. Ainsi les utilisateurs d'un système appartiennent à une ou plusieurs classes d'acteurs selon les rôles qu'ils tiennent par rapport au système. Une même personne physique peut se comporter en autant d'acteurs différents que le nombre de rôles qu'elle joue vis-à-vis du système.

- **Cas d'utilisation et interaction**

- Un cas d'utilisation correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur. Un cas d'utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou parties prenantes du système.
- Une interaction permet de décrire les échanges entre un acteur et un cas d'utilisation.

Un cas d'utilisation se représente par un ovale dans lequel figure son intitulé. L'interaction entre un acteur et un cas d'utilisation se représente comme une association.

Le formalisme de base de représentation d'un cas d'utilisation est donné à la figure ci-dessous :

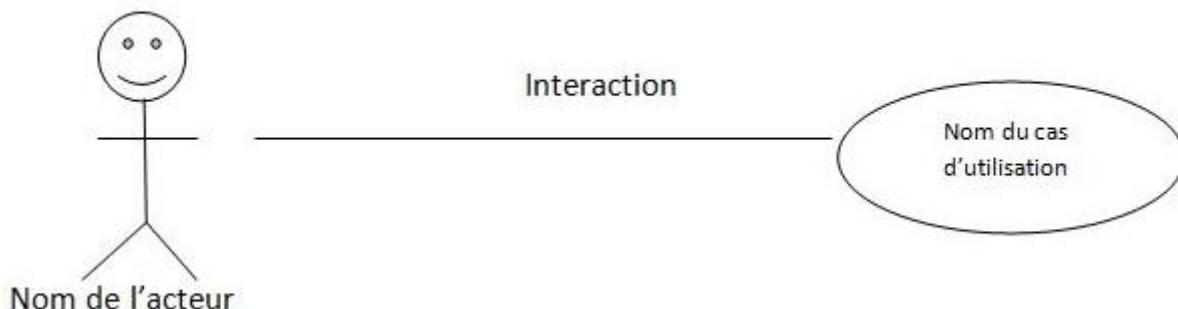


FIGURE II.1 : cas d'utilisation et interaction

b) Représentation du diagramme des cas d'utilisation

Tout système peut être décrit par un certain nombre de cas d'utilisation correspondant aux besoins exprimés par l'ensemble des utilisateurs. À chaque utilisateur, vu comme acteur, correspondra un certain nombre de cas d'utilisation du système. L'ensemble de ces cas d'utilisation se représente sous forme d'un diagramme. On pourrait représenter ce diagramme de la manière suivante :

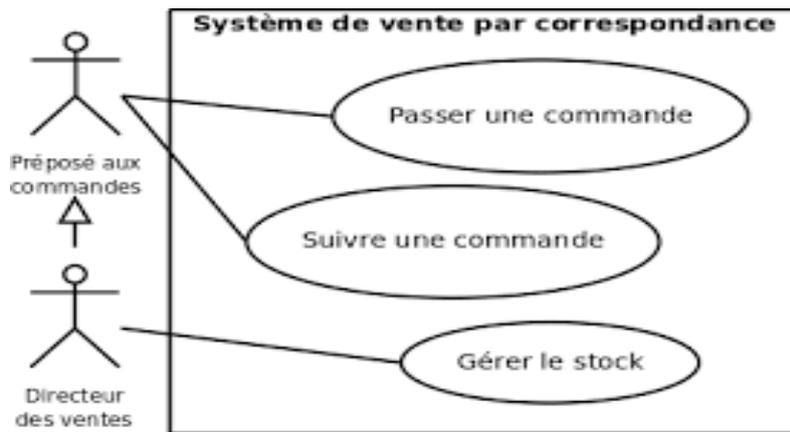


FIGURE II.2 : Représentation du diagramme de cas d'utilisation

c) Relations entre cas d'utilisation

UML définit trois types de relations standardisées entre cas d'utilisation :

- **Une relation d'inclusion** : le cas de base en incorpore explicitement un autre, à un endroit spécifié dans ses enchaînements. Le cas d'utilisation inclus n'est jamais exécuté seul, mais seulement en tant que partie d'un cas de base plus vaste. Elle est formalisée par un mot-clé <<include>> ;

Exemple : Avant de créer une commande, il faut d'abord s'authentifier.

- **Une relation d'extension** : formalisée par un mot-clé <<extend>>, une relation d'extension d'un cas d'utilisation A par un cas d'utilisation B signifie qu'une instance de A peut être étendue par le comportement décrit dans B ;

Exemple : Après la création d'une commande, on peut payer ou ne pas payer avant la finition.

- **Relation de généralisation/spécialisation** : consiste à hiérarchiser les cas d'utilisations

Exemple : Pour la gestion de la commande on peut spécifier la création, la modification.

d) Description textuelle des cas d'utilisation

À chaque cas d'utilisation doit être associée une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs. La description textuelle d'un cas d'utilisation est articulée en six points :

- **Objectif** : décrire succinctement le contexte et les résultats attendus du cas d'utilisation.
- **Acteurs concernés** : le ou les acteurs concernés par le cas doivent être identifiés en précisant globalement leur rôle.
- **Pré conditions** : si certaines conditions particulières sont requises avant l'exécution du cas, elles sont à exprimer à ce niveau.
- **Post conditions** : par symétrie, si certaines conditions particulières doivent être réunies après l'exécution du cas, elles sont à exprimer à ce niveau.
- **Scénario nominal** : il s'agit là du scénario principal qui doit se dérouler sans incident et qui permet d'aboutir au résultat souhaité.
- **Scénarios alternatifs** : les autres scénarios, secondaires ou correspondants à la résolution d'anomalies, sont à décrire à ce niveau. Le lien avec le scénario principal se fait à l'aide d'une numérotation hiérarchisée rappelant le numéro de l'action concernée.

B- DIAGRAMME DE CLASSE

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations. Le détail des traitements n'est pas représenté directement dans le diagramme de classe ; seul l'algorithme général et le pseudocode correspondant peuvent être associés à la modélisation.

La description du diagramme de classe est fondée sur :

a) Le concept d'objet

Un objet est un concept, une abstraction ou une chose qui a un sens dans le contexte du système à modéliser. Chaque objet a une identité et peut être distingué des autres sans considérer a priori les valeurs de ses propriétés.

Un objet est caractérisé par les valeurs de ses propriétés qui lui confèrent des états significatifs suivant les instants considérés.

b) Le concept de classe

Une classe décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition).

Un objet est une instance d'une classe. La classe représente l'abstraction de ses objets. Au niveau de l'implémentation, c'est-à-dire au cours de l'exécution d'un programme, l'identificateur d'un objet correspond une adresse mémoire.

Une classe se représente à l'aide d'un rectangle comportant plusieurs compartiments.

Les trois compartiments de base sont :

- La désignation de la classe ;
- La description des attributs ;
- La description des opérations.

Deux autres compartiments peuvent être aussi indiqués :

- La description des responsabilités de la classe ;
- La description des exceptions traitées par la classe.

Cependant Il est possible de manipuler les classes en limitant le niveau de description à un nombre réduit de compartiments selon les objectifs poursuivis par le modélisateur.

Ainsi les situations suivantes sont possibles pour la manipulation d'une description restreinte de classe :

- Description uniquement du nom et des caractéristiques générales de la classe ;
- Description du nom de la classe et de la liste d'attributs.

Formalisme d'une classe

Nom de la classe
Attributs
Opérations
Responsabilités et/ou exception

Exemple de classe

Personne
Nom Prénom
<u>Afficher()</u> ;

FIGURE II.3 : Formalisme d'une classe

c) RELATION ENTRE CLASSES

➤ UNE ASSOCIATION

Une association exprime une connexion sémantique bidirectionnelle entre classe. C'est une abstraction des liens qui existent entre les objets instances des classes associées. Elle est représentée par un trait plein, pouvant être orienté. La multiplicité est notée du côté du rôle cible de l'association. Elle spécifie le nombre d'instances pouvant être associées (liées) avec une seule instance source de la relation.

Une association comporte des cardinalités et peut se présenter comme suit :

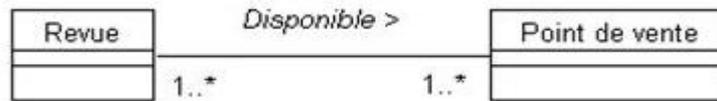


FIGURE II.4 : Représentation d’une association

Interprétation :

Une revue est disponible dans un ou plusieurs points de vente. Un point de vente distribue de une à plusieurs revues.

➤ **RELATION DE SPECIALISATION /GENERALISATION**

La généralisation décrit une relation entre une classe générale (classe parent) et une classe spécialisée (sous-classe). La classe spécialisée est intégralement cohérente avec la classe de base, mais comporte des informations supplémentaires (attributs, opérations, associations). Un objet de la classe spécialisée peut être utilisé partout où un objet de la classe de base est autorisé. On parle également de relation d’héritage.

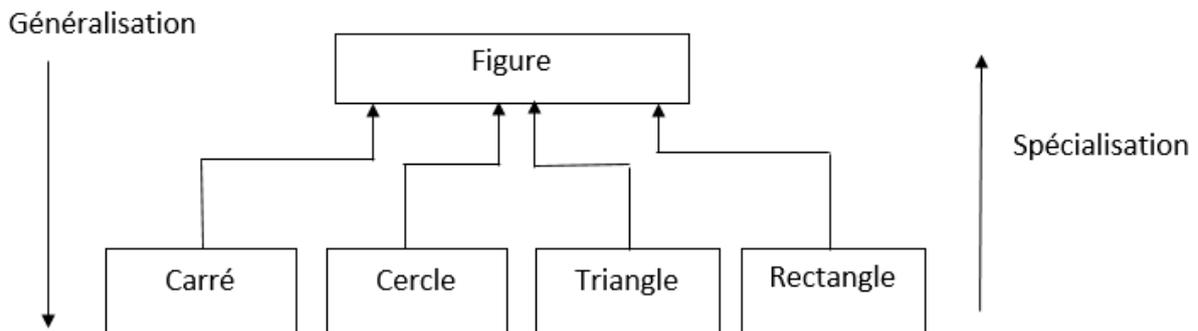


FIGURE II.5: Relation spécialisation/généralisation

➤ **RELATION DE DEPENDANCE**

Une dépendance est une relation unidirectionnelle exprimant une dépendance sémantique entre des éléments du modèle. Autrement dit, c’est une relation entre deux éléments dans laquelle une modification à un élément provoque des modifications sur l’autre élément. Elle est représentée par un trait discontinu orienté vers l’élément indépendant.

C- DIAGRAMME D’ACTIVITE

Le diagramme d'activité est un diagramme agissant sur les traitements, Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Les concepts spécifiques au diagramme d'activités sont les suivants :

- **Les nœuds de bifurcation** : ils permettent de créer plusieurs flots concurrents en sortie de la barre de synchronisation à partir du flot unique.
- **Les nœuds de jonction** : ils permettent de produire un seul flot unique sortant à partir de plusieurs flots concurrents en entrant de la synchronisation.
- **Le nœud de fusion-test** : permet d'avoir plusieurs flots entrants et un seul flot sortant. Celui-ci est exécuté dès que le flot sortant est activé.
- **Un pin d'entrée et de sortie** : représente un paramètre que l'on peut spécifier en entrée ou en sortie d'une action.
- **Un nœud d'objet** : il permet de représenter le flot de données véhiculé entre les actions. Les objets peuvent se représenter de deux manières différentes.
- **Partition** : souvent appelées couloirs ou lignes d'eau du fait de leur notation, permettent d'organiser les nœuds d'activités dans un diagramme d'activités en opérant des regroupements.

II- ETUDE DE L'EXISTANT

Comme mentionné plus haut une application web conçue avec le Framework Symfony a été mise en place et est utilisé actuellement dans le cas du Projet LASCOPE. Cette application permet aux opérateurs de saisir, de rentrer de manière électronique les données des patients renseignées sur des fiches papiers par des agents de santé sur les sites de suivies au Nigéria. Ces fiches papiers sont scannées et téléchargées sur la plateforme collaborative web de stockage et partage de fichiers du Programme PACCI ou les opérateurs peuvent les récupérer pour la saisie.

III- CRITIQUE DE L'EXISTANT

Un volet très important de ce genre de projet est le monitoring de données. Il consiste à surveiller les données recueillies et saisies afin de s'assurer de leurs complétudes. Ensuite, vient l'analyse des données en continue et en fin de projet, pour tirer les conclusions de l'études. L'outil présentement utilisé ne dispose pas de module de monitoring incorporé qui permettrait de faire des demandes de corrections. Il existe toutefois un autre outil de monitoring développé par le service informatique sous le système de base de données ACCESS.

IV- PROPOSITION DE SOLUTION

Vu l'urgence du calendrier du projet et le temps de développement conséquent pour la mise en place complète d'un module de monitoring, la solution retenue est de développer rapidement un module d'export de données d'une base MySQL vers ACCESS. Le format ACCESS généré pourra être utilisée dans l'outils de monitoring déjà existant dont nous avons parlé plus haut.

PARTIE III : ETUDE DETAILLEE (APPLICATION DU LANGAGE DE MODELISATION UML A NOTRE ETUDE)

I- DIAGRAMME DE CAS D'UTILISATION

1- PRESENTATION DES ACTEURS

Un acteur représente une entité (personnes morales ou physiques) pouvant interagir directement avec le système étudié. Il peut consulter ou modifier l'état du système, en émettant et/ou en recevant des messages susceptibles d'être porteurs de données.

Dans notre cas quatre types d'acteurs seront définis pour utiliser notre application :

- **Opératrice de saisie** : c'est l'acteur chargé de la saisie des données dans la base via l'interface utilisateur ;
- **Administrateur** : il s'agit de l'acteur principal, il fait des mises à jour de la base de données, assure la gestion des comptes utilisateur, gère les droits des utilisateurs ;
- **Gestionnaire de base de données** : chargé de faire le contrôle et la vérification de la qualité des données en base de données ;
- **Médecin** : chargé de consulter les différentes fiches de suivi des patients saisies dans la base.

2- IDENTIFICATION DES CAS D'UTILISATION

Un cas d'utilisation est une séquence d'activités ou d'actions organisées en étapes distinctes, et qu'un système effectue pour répondre à une sollicitation extérieure.

Dans le système à développer, nous avons identifié les cas d'utilisations suivantes :

ACTEUR	CAS D'UTILISATION
<ul style="list-style-type: none"> ➤ Administrateur ➤ Operateur de saisie ➤ Médecin 	Authentification
Administrateur	Gestion des comptes utilisateur <ul style="list-style-type: none"> ➤ Création de compte ➤ Ajout d'un utilisateur ➤ Modification d'utilisateur ➤ Suppression d'un compte Gestion des droits d'utilisateur <ul style="list-style-type: none"> ➤ Attribution de permission à un utilisateur ou groupe d'utilisateur (droit de lecture, écriture...) Maintenance du site <ul style="list-style-type: none"> ➤ Modification ou mise à jour de l'application
Gestionnaire de base de données	Export des données
Opératrice de saisie	Saisie des données

➤ Médecin ➤ Opératrice de saisie	Consultation de fiches

TABLEAU III.1 : Identisation des cas d'utilisation

3- DESCRIPTION TEXTUELLE DES CAS D'UTILISATION

Cette partie consiste à décrire chaque cas d'utilisation :

- **Cas d'utilisation : « authentification »**

Objectif	Avoir la possibilité d'accéder aux différents comptes
Acteur	Administration, Opératrice de saisie, médecin, gestionnaire de base de données
Pré-condition	L'utilisateur doit saisir l'adresse web de l'application, le système affiche le formulaire d'authentification
Post-condition	
Scénario nominal	<ul style="list-style-type: none"> ➤ L'utilisateur saisit ses paramètres de connexion et soumet le formulaire ➤ Le système récupère les données saisies puis vérifie si celles-ci sont conformes à la base de données ➤ Si les données sont correctes, alors l'utilisateur a accès à l'application
Scénario alternatif	<ul style="list-style-type: none"> ➤ Sinon l'accès est refusé et le système réaffiche le formulaire de connexion

TABLEAU III.2 : Description textuelle de cas d'utilisation « Authentification »

- **Cas d'utilisation « saisie des données »**

Objectif	Renseigner les différents formulaires de la base
Acteur	Opératrice de saisie
Pré-condition	L'utilisateur doit être authentifié
Post-condition	
Scénario nominal	<ul style="list-style-type: none"> ➤ L'opératrice choisit le site ➤ Le système affiche la liste des patients du site et Donne le choix à l'utilisateur de sélectionner un patient existant ou de créer une nouvelle fiche admission ➤ Si l'opératrice sélectionne un patient existant ➤ Le système affiche les données du patient ➤ Si l'opératrice choisit de créer une nouvelle fiche admission ➤ Le système affiche le nouveau formulaire ➤ L'opératrice saisit les données, ➤ Le système les stocke dans la base de données.
Scénario alternatif	<ul style="list-style-type: none"> ➤ Coupure d'électricité ➤ Déconnexion du réseau

TABLEAU III.3: Description textuelle de cas d'utilisation « Saisie de données »

• **Cas d'utilisation « consultation des fiches »**

Objectif	Consulter les fiches déjà enregistrées
Acteur	Opératrice de saisie, médecin
Précondition	L'utilisateur doit pouvoir s'authentifier
Post-condition	
Scénario nominal	<ul style="list-style-type: none"> ➤ L'utilisateur choisit le formulaire à visualiser ➤ Le système affiche le formulaire
Scénario alternatif	<ul style="list-style-type: none"> ➤ Coupure d'électricité ➤ Déconnexion du réseau

TABLEAU III.4: Description textuelle de cas d'utilisation « Consultation des fiches »

• **Cas d'utilisation « export de données »**

Objectif	Exporter les données vers un site
Acteur	Gestionnaire de base de données
Précondition	L'utilisateur doit pouvoir s'authentifier
Post-condition	
Scénario nominal	<ul style="list-style-type: none"> ➤ Le gestionnaire choisit la rubrique « exportation de données » ➤ Le système affiche le formulaire d'exportation ➤ Le gestionnaire choisit le site vers lequel les données doivent être exportées ➤ Le système exécute la tâche .
Scénario alternatif	<ul style="list-style-type: none"> ➤ Coupure d'électricité ➤ Déconnexion du réseau

TABLEAU III.5: Description textuelle de cas d'utilisation « Export de données »

• **Cas d'utilisation « gestion des comptes utilisateur »**

➤ **Ajout d'un utilisateur**

Titre	Ajout d'un utilisateur
Objectif	Donner la possibilité d'ajouter un utilisateur
Acteur	Administrateur
Pré-condition	L'utilisateur doit pouvoir s'authentifier
Post-condition	
Scénario nominal	<ul style="list-style-type: none"> ➤ L'administrateur demande au système le formulaire d'ajout d'un utilisateur ➤ Le système affiche le formulaire ➤ L'administrateur saisit les données de l'utilisateur à ajouter ➤ Le système les envoie dans la base de données et effectue une vérification ➤ Le système confirme l'ajout si les données sont bien correctes
Scénario alternatif	<ul style="list-style-type: none"> ➤ Sinon il affiche un message d'erreur ➤ Coupure d'électricité ➤ Déconnexion du réseau

TABLEAU III.6: Description textuelle de cas d'utilisation « Ajout d'un utilisateur »

➤ **Modification d'un utilisateur**

Titre	Modification d'un utilisateur
Objectif	Donner la possibilité de modifier un utilisateur
Acteur	Administrateur
Pré-condition	L'utilisateur doit pouvoir s'authentifier
Post-condition	
Scénario nominal	<ul style="list-style-type: none"> ➤ L'administrateur demande au système le formulaire de modification d'un utilisateur ➤ Le système affiche le formulaire ➤ L'administrateur saisit le numéro matricule de l'utilisateur à modifier ➤ Le système envoie la requête à la base de données ➤ La base de données recharge la requête au système ➤ Le système affiche les données de l'utilisateur ➤ L'administrateur saisit les nouvelles données de l'utilisateur ➤ Le système les envoie à la base de données pour le stockage
Scénario alternatif	<ul style="list-style-type: none"> ➤ Si l'utilisateur n'existe pas, le système affiche un message d'erreur ➤ Coupure d'électricité ➤ Déconnexion du réseau

TABLEAU III.7 : Description textuelle de cas d'utilisation « Modification d'un utilisateur

➤ **Suppression d'un utilisateur**

Titre	Suppression d'un utilisateur
Objectif	Donner la possibilité de supprimer un utilisateur
Acteur	Administrateur
Pré-condition	L'utilisateur doit pouvoir s'authentifier
Post-condition	
Scénario nominal	<ul style="list-style-type: none"> ➤ L'administrateur demande au système le formulaire de suppression d'un utilisateur ➤ Le système affiche le formulaire ➤ L'administrateur saisit le numéro matricule de l'utilisateur à supprimer ➤ Le système envoie la requête à la base de données ➤ La base de donnée recharge la requête ➤ Le système affiche les données de l'utilisateur ➤ L'administrateur demande au système la suppression d'un utilisateur ➤ Le système affiche un message de confirmation ➤ L'administrateur confirme la suppression ➤ Le système fait le chargement
Scénario alternatif	<ul style="list-style-type: none"> ➤ Si l'utilisateur n'existe pas, le système un message d'erreur ➤ Coupure d'électricité ➤ Déconnexion du réseau

TABLEAU III.8 : Description textuelle de cas d'utilisation « Suppression d'un utilisateur »

4- DIAGRAMME DE CAS D'UTILISATION GLOBAL

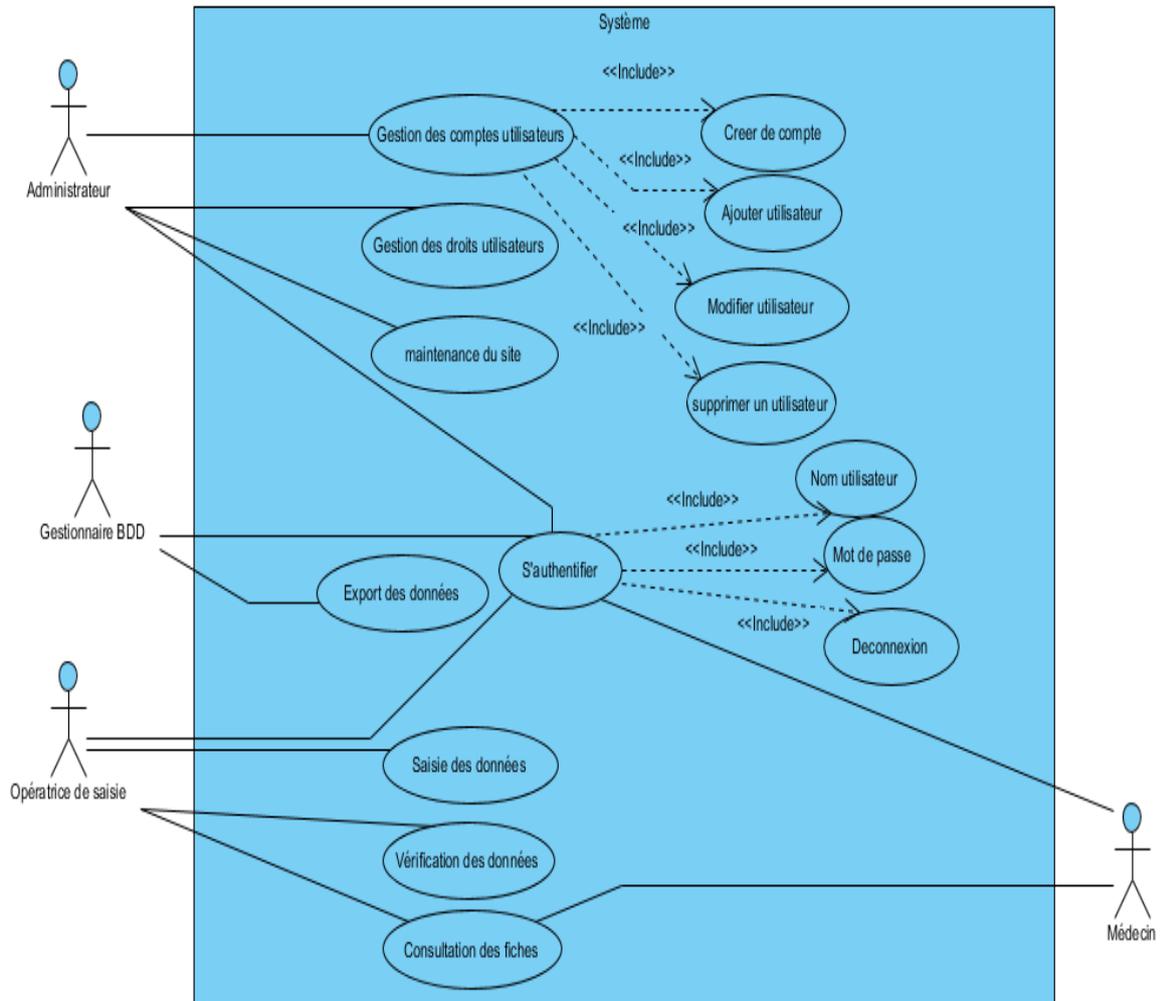


FIGURE III.1 : Diagramme du cas d'utilisation global

1- Cas : authentification

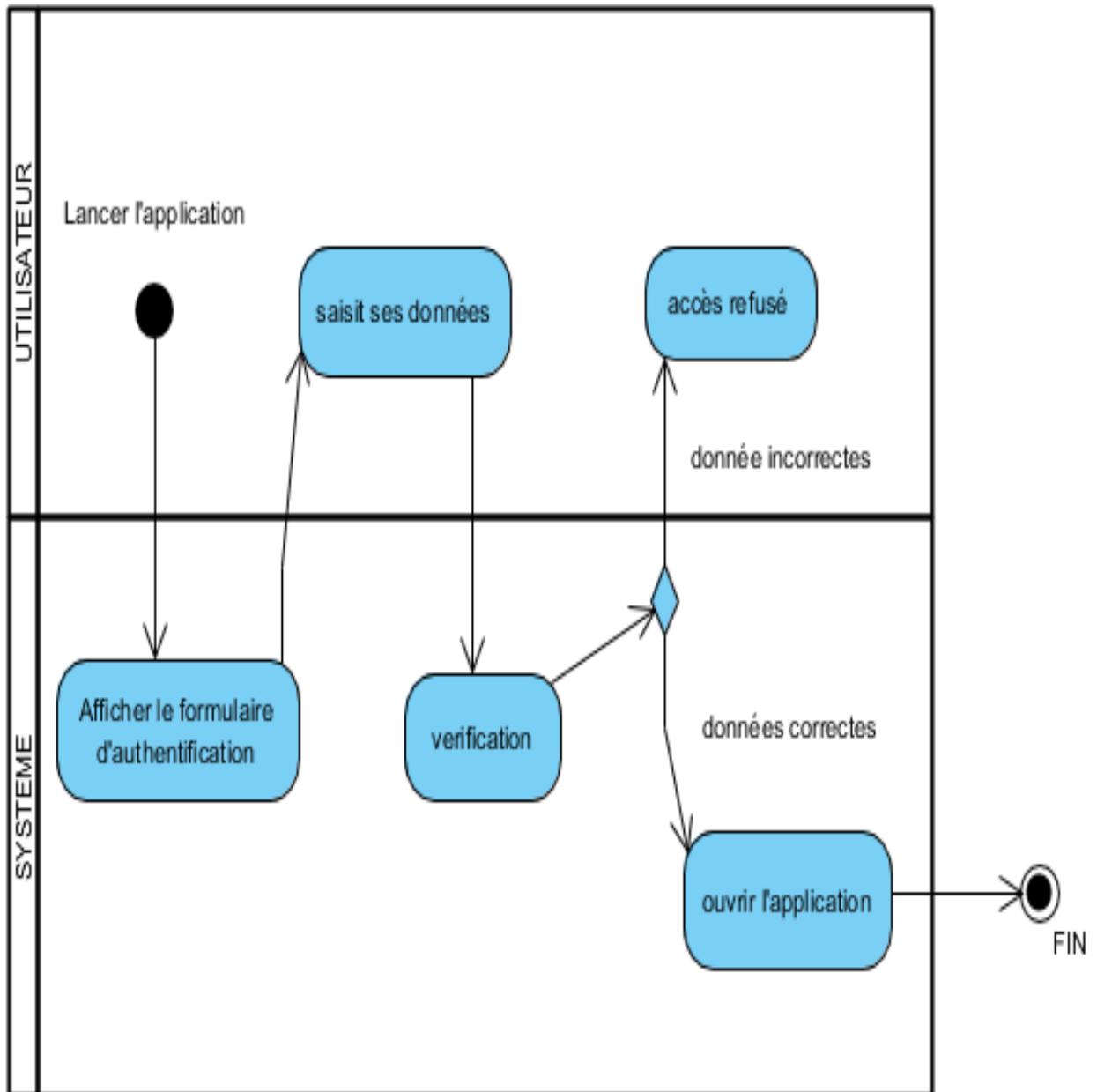


FIGURE III.3 : Diagramme d'activité « authentification »

2- Cas : saisie de données

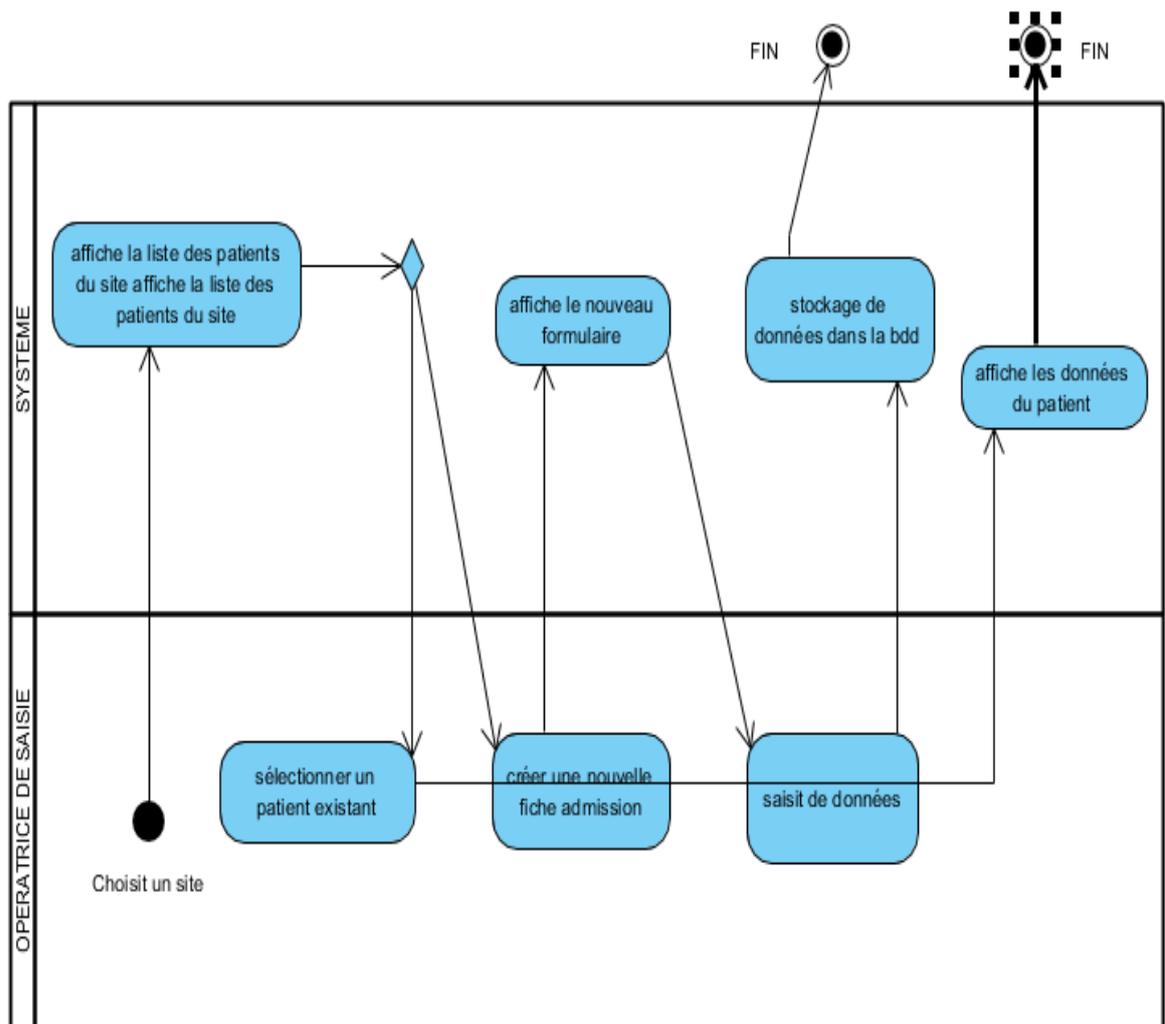


FIGURE III.4 : Diagramme d'activité « saisie de données »

3- Cas : consultation de fiche

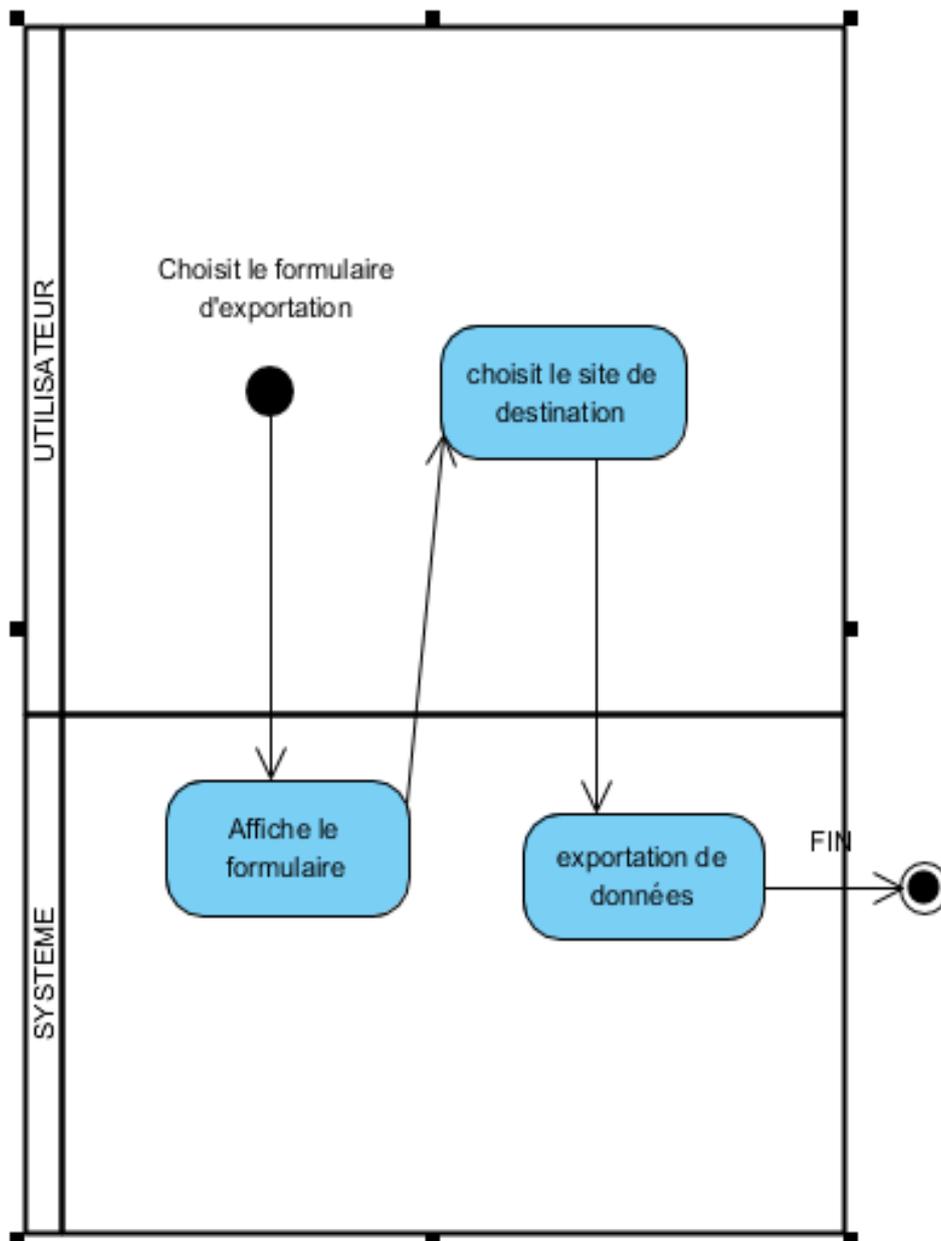


FIGURE III.5: Diagramme d'activité « consultation de fiches »

IV- PRINCIPE DE FONCTIONEMENT DU MODULE D'EXPORTATION

Le module d'exportation est un script qui permettra le transfert de la base de données MySQL vers la base ACCESS. L'exportation se fera par la soumission d'un formulaire. L'utilisateur pourra faire un choix d'exporter en fonction de ses droits soit toutes les données de l'étude ou les données d'un site bien précis.

Pour se faire, nous avons ajouté une nouvelle table de travail « **LIS_TAB** » dont les données sont celle concernées par l'exportation.

1- Présentation de la table LIS_TAB

- Structure de la nouvelle table LIS_TAB

```
-- Structure de la table `lis_tab`
--
DROP TABLE IF EXISTS `lis_tab`;
CREATE TABLE IF NOT EXISTS `lis_tab` (
  `TAB_IDE` int(11) NOT NULL AUTO_INCREMENT COMMENT 'Cl? primaire',
  `tab_lib` varchar(255) COLLATE latin1_general_ci NOT NULL,
  `tab_ref` varchar(255) COLLATE latin1_general_ci NOT NULL,
  PRIMARY KEY (`TAB_IDE`)
) ENGINE=InnoDB AUTO_INCREMENT=10 DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
--
```

FIGURE III.6 : Structure de la table LIS_TAB

- Déchargement des données de la table LIS_TAB

```
-- Déchargement des données de la table `lis_tab`
--
INSERT INTO `lis_tab` (`TAB_IDE`, `tab_lib`, `tab_ref`) VALUES
(1, 'crf_adm', 'ADM_IDE'),
(2, 'crf_bio', 'BIO_IDE'),
(3, 'crf_bio_exa', 'BIO_EXA_IDE'),
(4, 'crf_bio_det', 'BIO_DET_IDE'),
(5, 'crf_con', 'CON_IDE'),
(6, 'crf_dia', 'DIA_IDE'),
(7, 'crf_dia_ses', 'DIA_SES_IDE'),
(8, 'crf_fol', 'FOL_IDE'),
(9, 'crf_pre', 'PRE_IDE');
```

FIGURE III.7: Déchargement des données de la table LIS_TAB

- Vue de la table depuis la base de données

✓ Affichage des lignes 0 - 8 (total de 9, traitement en 0,0000 seconde(s).)

`SELECT * FROM `lis_tab``

Profilage [Éditer en ligne]

Tout afficher | Nombre de lignes : 25 | Filtrer les lignes : Chercher dans cette tabl | Trier sur l'index: Aucun(e)

- Options

	TAB_IDE <small>Clé primaire</small>	tab_lib	tab_ref
<input type="checkbox"/> Éditer Copier Supprimer	1	crf_adm	ADM_IDE
<input type="checkbox"/> Éditer Copier Supprimer	2	crf_bio	BIO_IDE
<input type="checkbox"/> Éditer Copier Supprimer	3	crf_bio_exa	BIO_EXA_IDE
<input type="checkbox"/> Éditer Copier Supprimer	4	crf_bio_det	BIO_DET_IDE
<input type="checkbox"/> Éditer Copier Supprimer	5	crf_con	CON_IDE
<input type="checkbox"/> Éditer Copier Supprimer	6	crf_dia	DIA_IDE
<input type="checkbox"/> Éditer Copier Supprimer	7	crf_dia_ses	DIA_SES_IDE
<input type="checkbox"/> Éditer Copier Supprimer	8	crf_fol	FOL_IDE
<input type="checkbox"/> Éditer Copier Supprimer	9	crf_pre	PRE_IDE

↑ Tout cocher Avec la sélection : Éditer Copier Supprimer Exporter

FIGURE III.8 : vue d'ensemble de la table dans la BDD

2- Présentation et description du script

- Récupération du code du site

```
public function index()
{
    $em = $this->getDoctrine()->getManager() ;
    $siteRepository = $em->getRepository('App:Site');
    $sites = $siteRepository->findAll();
    return $this->render('export/index.html.twig', [
        'controller_name' => 'ExportController',
        'sites' => $sites,
    ]);
}
```

FIGURE III.9 : code de récupération du site

- Récupération des objets de la BDD
- Recherche de l'entité 'site'

- Récupération de toutes les données contenue dans 'site'
- Affichage du site

CREATION DE LA BASE ACCESS

- Création à la connexion ODBC

```

$VAR_TBL_TBL = "LIS_TAB";
    $projectRoot = $this->get('kernel')->getProjectDir();
    if (PHP_OS == "WINNT") {
        $VAR_CHEMIN = "$projectRoot\databases\XXX_BDS.accdb";
        //dd($VAR_CHEMIN);
        if (!file_exists($VAR_CHEMIN)) {
            die('Error finding access database');
        }
    }

    $em = $this->getDoctrine()->getManager();

    $qb = $em->getRepository("App:LisTab");
    $RES_REQ_LST_TAB = $qb->findAll();

    if (PHP_OS == "WINNT") {
        try {
            $VAR_DBH_ODBC = new PDO("odbc:Driver={Microsoft Access
Driver (*.mdb, *.accdb)};Dbq=$VAR_CHEMIN;Uid=Admin");
        } catch (PDOException $VAR_EXP_ODBC) {
            echo "<p>Erreur de connexion à la BD avec PDO ODBC => Voir
fichier 'import.ctr.php' </p>";
            echo "<p>Erreur ! : " . $VAR_EXP_ODBC->getMessage() .
"</p>";
            die();
        }
    } else $VAR_DBH_ODBC = "";

    // Parcourir le resultat et créer les tables dans la base access
    foreach ($RES_REQ_LST_TAB as $TAB_ROW) {

        // On supprime l'ancienne table dans access

        $REQ_SUP_TAB = "DROP TABLE " . $TAB_ROW->getTabLib();
        if (PHP_OS == "WINNT") {
            $VAR_DBH_ODBC->exec($REQ_SUP_TAB);
        }
    }

```

FIGURE III.10 : code de création à la connexion ODBC

- Récupération de la table LIS_TAB
- Récupération du répertoire racine du projet

- Indexation du chemin d'accès à la BDD ACCESS
- Si chemin inexistant, affichage d'un message d'erreur
- Récupération des objets de la BDD
- Recherche de l'entité 'LIS_TAB'
- Récupération de toutes les données contenue dans 'LIS_TAB' (\$RES_REQ_LST_TAB)
- Création de la connexion au pilote ODBC
- Parcours de chaque élément de LIS_TAB
- Création des tables correspondant à chaque élément dans la base ACCESS
- Suppression des anciennes tables dans ACCESS

- **Parcours des résultats de la description**

```
// On récupère la description de la table dans MySQL

$STM_REQ_DESC_TAB = $qb->describe Table($TAB_ROW->getTabLib());
$RES_REQ_DESC_TAB = $STM_REQ_DESC_TAB;
$NBR_COL = count($RES_REQ_DESC_TAB);
$VAR_CHAMP = "";

// On parcourt le resultat de la description

foreach ($RES_REQ_DESC_TAB as $TAB_DESC_ROW) {
    switch ($TAB_DESC_ROW->Type) {
        case 'int(11)':
            $VAR_TYP = 'int';
            break;

        case 'decimal(11,1)':
            $VAR_TYP = 'double';
            Break ;

        case 'float':
            $VAR_TYP = 'double';
            break;

        case 'tinyint(1)':
            $VAR_TYP = 'int';
            break;

        case 'tinytext':
            $VAR_TYP = 'varchar(50)';
            break;

        case 'longtext':
            $VAR_TYP = 'varchar(255)';
            break;

        default:
```

```

        $VAR_TYP = $TAB_DESC_ROW->Type;
        break;
    }

    if ($TAB_DESC_ROW->Key === 'PRI') {
        $VAR_CLE = ' PRIMARY KEY(' . $TAB_DESC_ROW->Field .)';
    }

    $VAR_CHAMP .= $TAB_DESC_ROW->Field . ' ' . $VAR_TYP . ' ,';
}

// On constitue l'ordre SQL de création des tables dans access et on exécute

$REQ_CREAT_TAB = "CREATE TABLE " . $TAB_ROW->getTabLib() . "
(" . $VAR_CHAMP . $VAR_CLE .)";
$script .= $REQ_CREAT_TAB . ";" . PHP_EOL;
if (PHP_OS == "WINNT") {
    $VAR_DBH_ODBC->exec($REQ_CREAT_TAB);
}
}

```

FIGURE III.11 : parcours des résultats de la description.

- Renvoi d'informations sur les colonnes de LIS_TAB
- Parcours de chaque colonne de la LIS_TAB
- Comparaison des types de donnée de chaque colonne
- Constitution de l'ordre SQL de création des tables dans ACCESS
- Exécution des tables

EXPORTATION DES DONNEES

```

//-----Exportation des données-----//
// On parcourt à nouveau le résultat de la liste des tables dans MySQL

//Récupération de la liste des patients par site

$lispat = $qb->lisPat($SIT_COD);
$bio = null;
$bioexa = null;
$biodet = null;

```

```

foreach ($lispat as $pat) {
    foreach ($RES_REQ_LST_TAB as $TAB_ROW) {

        // Description de la table

        $RES_REQ_DESC_TAB = $qb->describeTable($TAB_ROW->getTabLib());
        $NBR_COL = count($RES_REQ_DESC_TAB);

        $ARR_CHAMP = $RES_REQ_DESC_TAB[0]->Field;
        $ARR_TYP_CHAMP = $RES_REQ_DESC_TAB[0]->Type;

        for ($i = 1; $i < $NBR_COL; $i++) {
            $ARR_CHAMP = $ARR_CHAMP . "-" . $RES_REQ_DESC_TAB[$i]->Field;
            $ARR_TYP_CHAMP = $ARR_TYP_CHAMP . "-" . $RES_REQ_DESC_TAB[$i]->Type;
        }

        $ARR_CHAMP = explode('-', $ARR_CHAMP);

        $ARR_TYP_CHAMP = explode('-', $ARR_TYP_CHAMP);

        // Requête récupération des données MySQL
        $sentName = $inflector->camelize(downcase($TAB_ROW->getTabLib ()),
Inflector::UPCASE_FIRST_LETTER);

        $repository = $em->getRepository("App:$sentName");
        if ($sentName === 'CrfBio') {
            $bio = $qb->findByMe('CRF_BIO', 'pat', $pat['PAT']);
            $table = $bio; }
        elseif ($sentName === 'CrfBioExa' && !empty($bio[0])) {

            $bioexa = $qb->findByMe('CRF_BIO_EXA', 'BIO_IDE', $bio[0]["BIO_IDE"]);
            $table = $bioexa;}
        elseif ($sentName === 'CrfBioDet' && !empty($bioexa[0])) {
            $logger->error('<pre>' . print_r($bioexa, true) . '</pre>');
            foreach ($bioexa as $bioexact) {
                $biodet0[0] = $qb->findByMe('CRF_BIO_DET', 'BIO_EXA_IDE',
$bioexact['BIO_EXA_IDE']);
                foreach ($biodet0[0] as $biodet1) {
                    $biodet[] = $biodet1;
                }
            }
        }
    }
}

```

FIGURE III.12 : Extrait du code d'exportation des données

PARTIE IV : IMPLEMENTATION ET DEPLOIEMENT DE L'APPLICATION

I- ETUDE TECHNIQUE

1- Environnement matériel

Les caractéristiques de la machine de travail sont les suivantes :

Ordinateur HP

- Mémoire ram 4 Go
- Processeur Intel® Pentium® Dual CPU E2200 @2.2GHz
- Disque dur 147.9 Go
- Système d'exploitation 64bits Windows 10 Entreprise

2- Environnement logiciel

- Le système de gestion de base de données (SGBD) utilisé est MySQL

- Le logiciel de développement : Visual Studio Code
- Framework : Symfony 4.1, PHP 7.1.16
- Langage de programmation : PHP, HTML5, CSS
- Langage de modélisation : Visual Paradigm
- Logiciel de rédaction du mémoire : Microsoft Word 2016

3- Description des logiciels de développement utilisés

- **SYMFONY**

Symfony est un ensemble de composants PHP ainsi qu'un Framework MVC libre écrit en PHP. Il fournit des fonctionnalités modulables et adaptables qui permettent de faciliter et d'accélérer le développement d'un site web. Symfony est un système fragmenté et chaque composant peut être utilisé individuellement et intégré dans un site internet.

- **Les différentes versions Symfony et PHP**

Symfony utilise la version 5 de PHP :

- PHP >= 5.2.4 pour Symfony 1.x
- PHP >= 5.3.3 pour Symfony 2.x
- PHP >= 5.5.9 pour Symfony 3.x

La dernière version Symfony 4.1 requiert la version 7.1.13 de PHP ou plus afin de pouvoir fonctionner. C'est la version utilisée dans le cas du Projet LASCOPE.

- **Comment installer et configurer Symfony ?**

Pour créer une application Symfony, il faut commencer par installer Composer globalement sur votre système. Il faut s'assurer également de la version de PHP en fonction de la version de Symfony que vous devez utiliser. Ensuite exécutez la ligne de code suivante :

Composer create-project symfony/website-skeleton « nom du projet »

- **Lancer une application Symfony**

En production, il faudrait utiliser un serveur web tel que Nginx ou Apache. Mais pour le développement, il est encore plus simple d'utiliser le serveur web Symfony PHP.

Pour lancer une application Symfony, il faut accéder au projet et démarrer le serveur :

```
PS C:\wamp64\www\lascope> php bin/console server:run
```

Puis ouvrir le navigateur et accéder à : <http://localhost:8000/>

- **VISUAL STUDIO CODE**

Visual Studio Code est un éditeur de code source léger mais puissant qui s'exécute sur le bureau et est disponible sur Windows, MacOS et Linux. Il est livré avec un support intégré pour JavaScript et possède un écosystème riche d'extensions pour d'autres langages (tels que C++, C #, java, Python, PHP, Etc.) et les environnements d'exécution tels que .NET.

- **WAMPSEVEUR**

WampServer (anciennement appelé WAMP5) est une plateforme de développement Web, permettant de faire fonctionner localement des scripts PHP (sans avoir à se connecter à un serveur externe). C'est un environnement comprenant deux serveurs (Apache et MySQL), un interpréteur de script (PHP), ainsi que phpMyAdmin pour l'administration Web des bases MySQL.

- **VISUAL PARADIGM**

Visual Paradigm est un outil UML CASE prenant en charge UML 2. Il permet aux programmeurs de mettre en places des diagrammes UML et fournit des fonctionnalités de génération de rapports de code aux format PDF, Word ou Html. Il est possible de générer un diagramme et ainsi produire du code source dans de nombreux langages tel que JAVA, C++, ou faire l'inverse c'est-à-dire générer des diagrammes à partir de code déjà existant.

- **MySQL**

MySQL est un système de gestion de base de données qui peut vous aider à concevoir des applications de base de données évolutives et hautement performantes.

II- REALISATION DE L'APPLICATION

1- Description des interfaces de l'application

a) Interface Authentification

Au lancement de l'application, un formulaire s'affiche à l'écran ; C'est le formulaire d'authentification. C'est lui qui nous permettra d'avoir accès à l'interface principale.

The screenshot shows the LASCOPE login page. At the top, the title 'LASCOPE' is displayed in a large, bold, black font, with the subtitle 'Connexion' below it in a smaller, regular black font. Below the title, there are two input fields: the first is labeled 'Nom d'utilisateur' and contains the placeholder text 'Nom d'utilisateur'; the second is labeled 'Mot de passe' and contains the placeholder text 'Mot de passe'. Below these fields is a blue button with the text 'Connexion' in white. The entire form is set against a light gray background.

Développé par  Logo PACCI  Logo MEREVA | V1.0. 2018

Lorsque l'utilisateur aura renseigné de mauvaises coordonnées, un message d'erreur apparaîtra comme ci-dessous.

This screenshot shows the same LASCOPE login page as above, but with an error message. The 'Nom d'utilisateur' field now contains the text 'ager'. Below the password field, a light red rectangular box contains the text 'Nom d'utilisateur ou mot de passe incorrect' in a dark red font. The 'Connexion' button remains visible at the bottom.

Développé par  Logo PACCI  Logo MEREVA | V1.0. 2018

FIGURE IV.1: Interface d'authentification

b) Interface principale

Cette interface comporte le menu principal, où l'utilisateur pourra sélectionner la tâche à effectuer selon les droits qui lui ont été attribués.

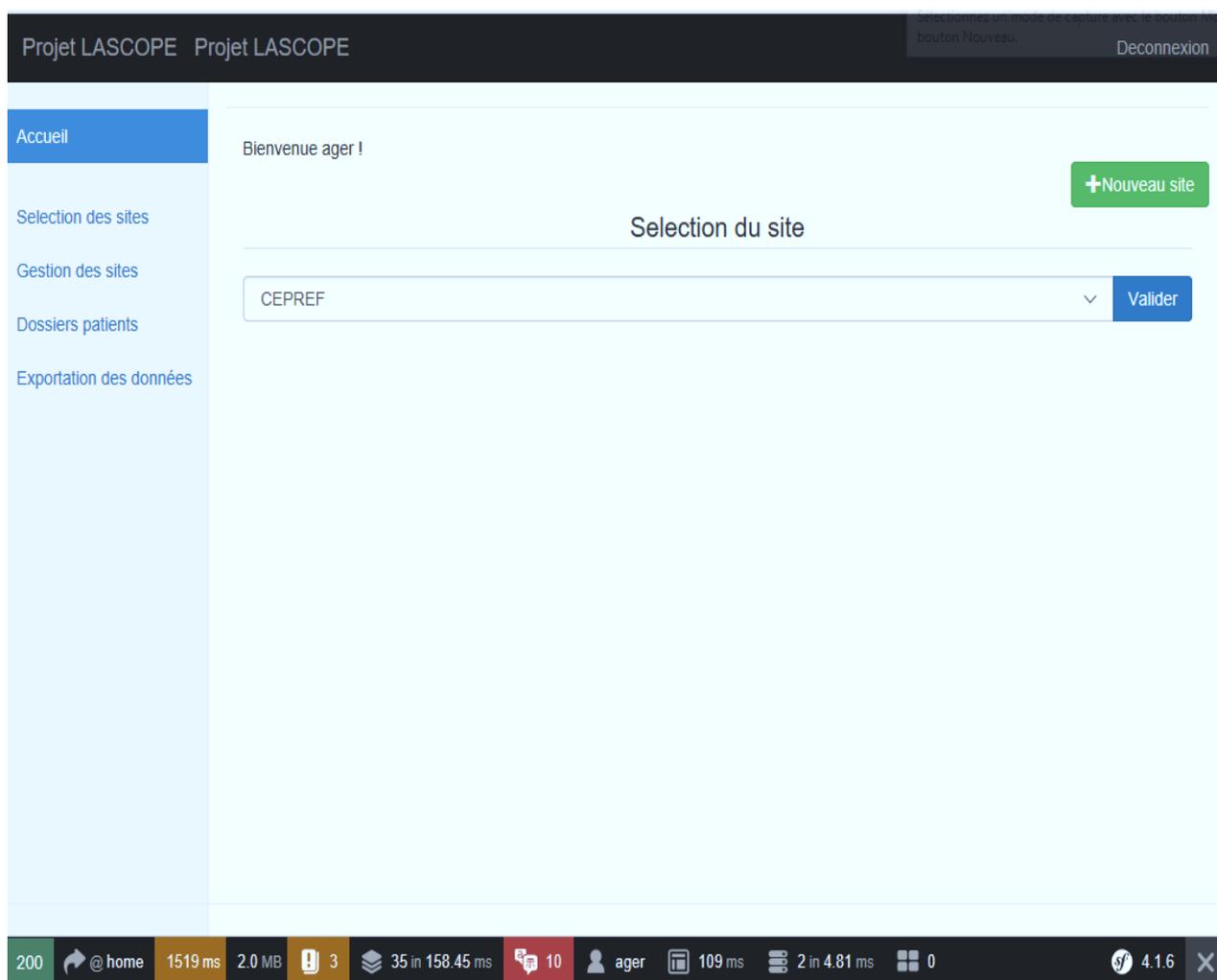


FIGURE IV.2: page d'accueil

Ce menu contient 6 onglets qui sont :

- **La sélection des sites** : permet de sélectionner un site.
- **La gestion des sites** : affiche la liste de tous les sites enregistrés et donne la possibilité de voir, de modifier ou d'ajouter un nouveau site.
- **Les dossiers patients** : affiche la liste des patients de tous les sites.
- **L'exportation des données** : permet au gestionnaire de base de données de sélectionner le site dont les données doivent être exportées.
- **Ajout de nouveau site** : permet d'enregistrer un nouveau site.
- **Déconnexion** : pour se déconnecter de l'application.

c) Onglet « sélection des sites »

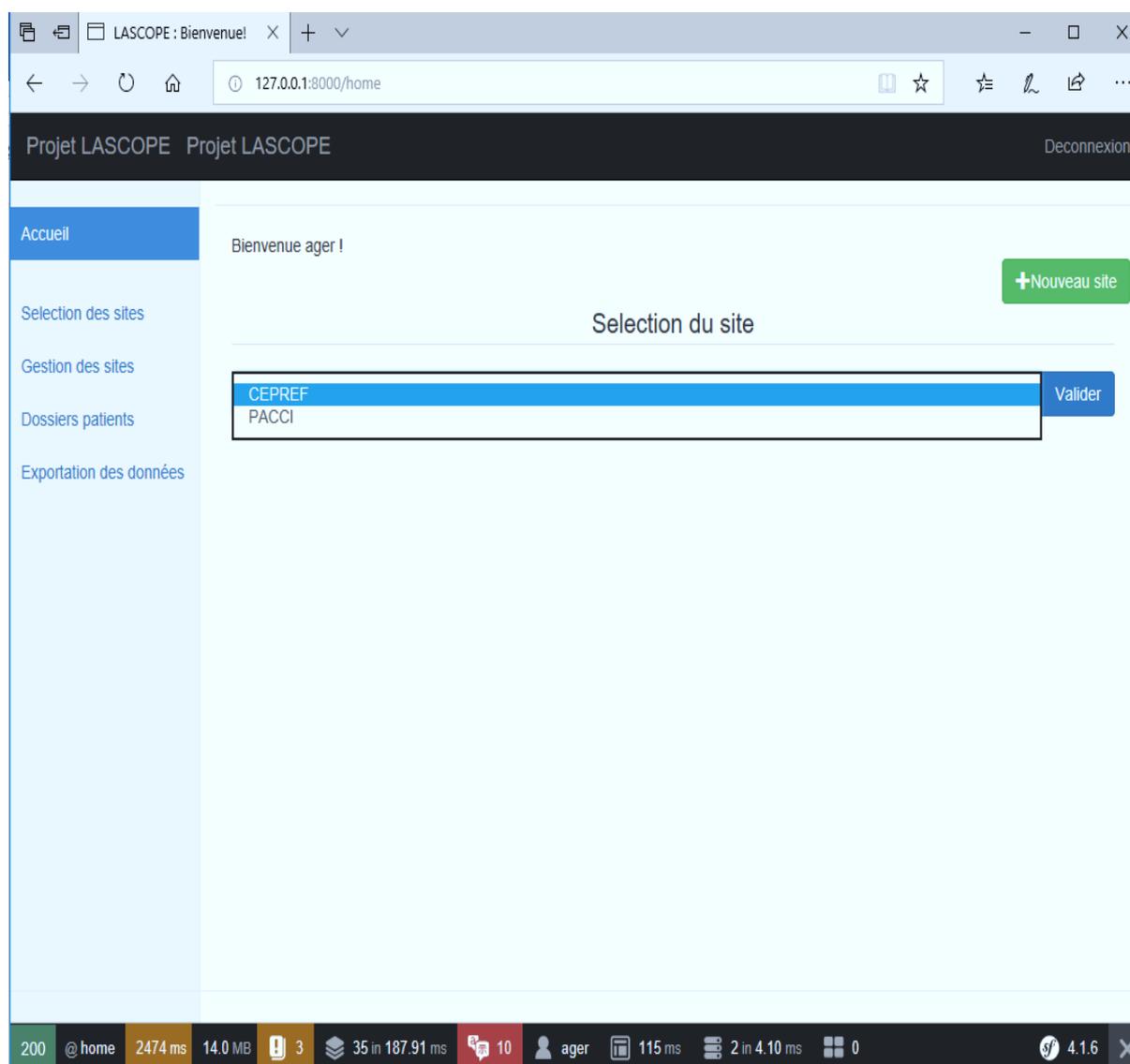


FIGURE IV.3: interface « sélection des sites »

d) Onglet « gestion des sites »

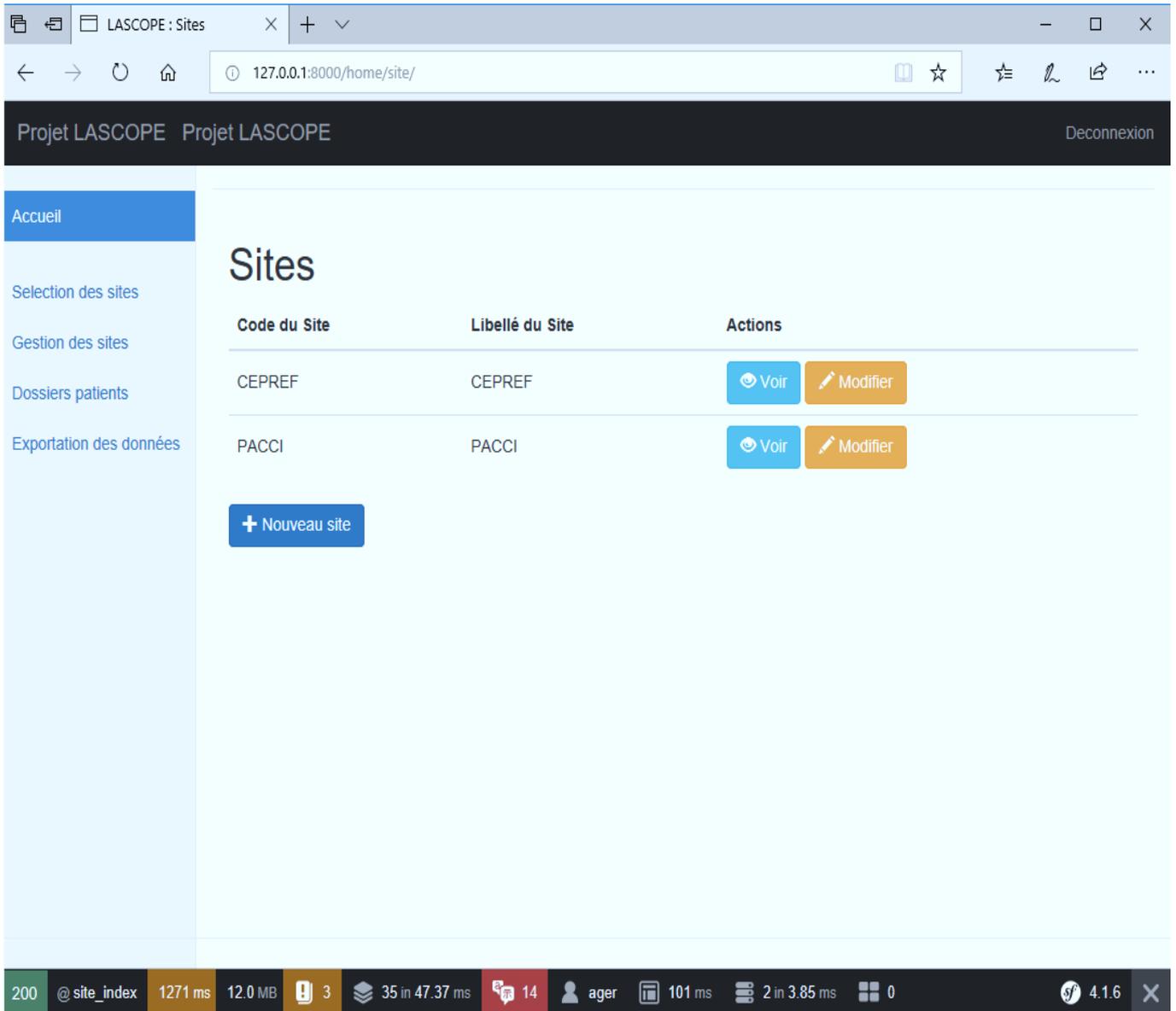


FIGURE IV.4: Interface « gestion des sites »

e) Onglet « Exportation des données »

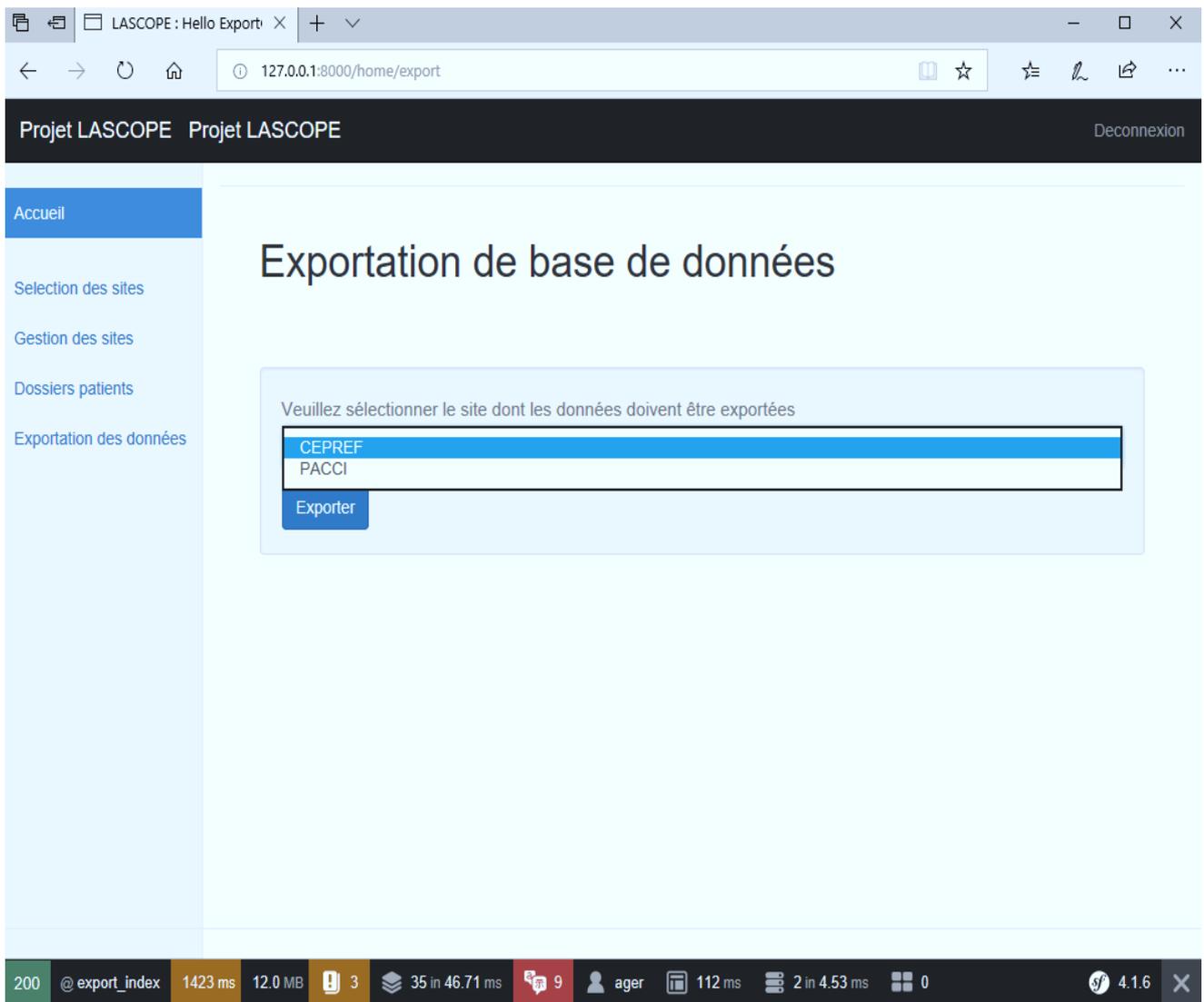


FIGURE IV.5: Interface « exportation des données »

f) Onglet « ajout nouveau site »

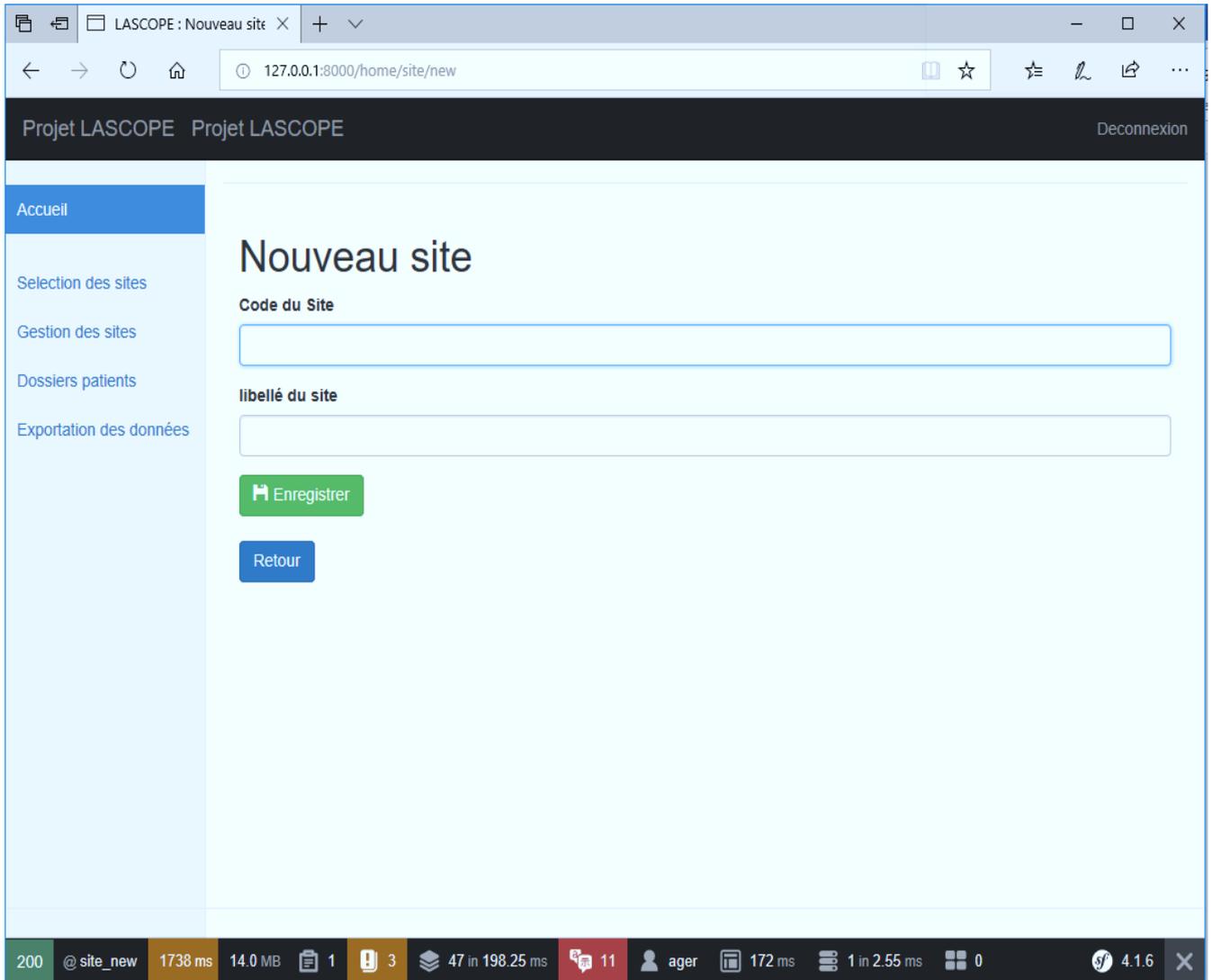


FIGURE IV.6: Interface « ajout nouveau site »

CONCLUSION

Au terme de ce projet, notons qu'il a été pour nous d'un grand avantage, parce qu'il nous a permis non seulement de participer à la vie de l'entreprise mais aussi d'acquérir une certaine expérience professionnelle.

La phase de conception a été le cœur du développement. Cette étape nous a permis de mettre en œuvre dans un cadre réel les connaissances acquises en méthodes de conception et méthodologie de développement. Elle nous a aussi permis d'approfondir nos connaissances en la matière à travers l'utilisation de la méthodologie UML qui est devenue quasiment incontournable. De plus, nous nous sommes familiarisés avec le Framework Symfony à partir duquel nous avons conçu et développé le module d'importation. Ainsi que le système de gestion de base de données MySQL qui a servi à sauvegarder la Base de données.

La réalisation de ce projet a fait l'objet d'une expérience intéressante d'une part mais parsemée de petites difficultés d'autre part. outre cela, nous en tirons grande satisfaction car elle nous a permis d'appréhender les difficultés rencontrées en entreprise et de faire la différence entre la théorie et la pratique.

L'application étant actuellement en production, d'autres modules sont en train d'être conçus tel que le gestionnaire des utilisateurs et consort.

~~Un autre objectif de ce projet est de proposer une version anglaise de l'interface de l'application, actuellement en français seulement.~~

WEBOGRAPHIE

SOMMAIRE

TABLE DES MATIERES.....	1
DEDICACE.....	3
REMERCIEMENTS	4
LISTE DES ACRONYMES	5
LISTE DES FIGURES	6
LISTE DES TABLEAUX.....	7
AVANT-PROPOS	8
INTRODUCTION.....	9

<u>PARTIE I : PRESENTATION DE LA STRUCTURE D'ACCUEIL.....</u>	<u>10</u>
I- HISTORIQUE	11
II- MISSIONS.....	11
III- ORGANISATION ET ROLES DES SERVICES	11
IV- ORGANIGRAMME.....	14
V- PRESENTATION DU SERVICE D'ACCUEIL	16
<u>PARTIE II : ETUDE PREALABLE.....</u>	<u>18</u>
I- PRESENTATION DU PROJET LASCOPE	19
II- ETUDE DE L'EXISTANT	26
III- CRITIQUE DE L'EXISTANT	26
IV- PROPOSITION DE SOLUTION	27
<u>PARTIE III : ETUDE DETAILLEE (APPLICATION DU LANGAGE DE MODELISATION UML A NOTRE ETUDE).....</u>	<u>27</u>
I- DIAGRAMME DE CAS D'UTILISATION	28
II- DIAGRAMME DE CLASSE	34
III- DIAGRAMME D'ACTIVITE	34
IV- PRINCIPE DE FONCTIONEMENT DU MODULE D'EXPORTATION.....	37
<u>PARTIE IV : IMPLEMENTATION ET DEPLOIEMENT DE L'APPLICATION.....</u>	<u>44</u>
I- ETUDE TECHNIQUE	44
II- REALISATION DE L'APPLICATION	46
CONCLUSION.....	52
WEBOGRAPHIE.....	53
SOMMAIRE	54