



Université Catholique de l'Afrique de l'Ouest
Faculté des Sciences de Gestion
INSTITUT SUPERIEUR DE GESTION SAINT MICHEL
Science – Foi – Action



Agrément : N° 05/AG/SAC/MESUCURRS/DES/DFS

Habilitation : N° RepSEN/Ensup-priv/HA/015-2017

SPECIALITE : INFORMATIQUE DE GESTION

MEMOIRE

Présenté par

Mme SALIMATA CISSE

Pour l'obtention du diplôme de

Master en INFORMATIQUE DE GESTION

SUJET

Etude et conception d'une plateforme pour la gestion d'une structure sanitaire généraliste : LE CAS D'UNE CLINIQUE

Soutenu à UCAO/Saint Michel le 22/012020 devant le jury composé de :

Président : Pr Cheikh Ahmadou Bamba GUEYE	Professeur Titulaire en Informatique	UCAD
Directeur de mémoire : Pr SAMBA NDIAYE	Professeur Titulaire en Informatique	UCAD
Co-encadreur : Dr Edouard Ngor Sarr	Maître-Assistant en Informatique	UCAO
Examineur : Dr Reine Marie MARONE	Assistante en Informatique	UCAO

Année 2018-2019

DEDICACES

Je dédie ce mémoire à toute famille et à tous ceux qui ont participé à ma formation. Mention spéciale à :

- Mon cher mari Papa Djibril Diagne ;
- Ma mère Aissatou Niang ;
- Ma sœur Fatou Touré Ndiaye ;
- Mon père Bara Cissé ;
- Ma tante Waré Thialé Niang.

REMERCIEMENTS

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je commencerais par mon encadreur le Dr Edouard Ngor SARR, je tiens à vous exprimer mes plus vifs remerciements car vous avez été un encadreur disponible malgré vos nombreuses charges. Vos compétences, votre rigueur et votre clairvoyance m'ont beaucoup aidé à arriver au bout du tunnel.

J'exprime tous mes remerciements à l'ensemble des membres du jury.

Je remercie tout le personnel de l'Université Catholique de l'Afrique de l'Ouest (UCAO Saint Michel) de Dakar.

Un grand merci à mon Directeur des études Monsieur Remy BASSE et à tous mes professeurs de l'UCAO. Ils m'ont fournies les outils nécessaires à la réussite de mes études universitaires.

A mes amis, à la promotion Informatique de Gestion 2018.

Les mots les plus simples étant les plus forts, j'adresse toute mon affection à ma famille. Je remercie aussi ma belle-famille particulièrement à ma belle-mère, merci pour le soutien et les encouragements.

Et enfin, mon bien-aimé, mon mari, Papa Djibril DIAGNE, merci à toi chéri pour ton amour, ta tendresse, ta présence et ta grande patience. Je ne serai peut-être pas arrivée au bout de cette aventure sans toi.

RÉSUMÉ

En informatique, une plateforme désigne un matériel ou un logiciel qui héberge une application ou un service. Une plateforme applicative, se compose d'un matériel, d'un système d'exploitation et de programmes de coordination qui utilisent le jeu d'instructions d'un processeur ou microprocesseur particulier. Ainsi, la plateforme constitue la base sur laquelle repose la bonne exécution d'un code objet. Notre étude porte sur la conception d'une plateforme pour une clinique généraliste et regroupe un ensemble de modules. Chaque module correspond à un lot de fonctionnalités associées. On aura les différents modules suivants :

1. Module de Gestion des rendez-vous
2. Module de Gestion des consultations
3. Module de Gestion des dossiers médicaux des patients
4. Module de Gestion des ressources humaines
5. Gestion des hospitalisations et de la logistique
6. Module de Gestion financière liée aux consultations et aux hospitalisations

MOTS-CLES : Plateforme, SANTE

ABSTRACT

In IT, a platform designated as hardware or software that hosts an application or service. An application platform, for example, consists of hardware, an operating system, and coordination programs that use the instruction set of a particular processor or microprocessor. In this case, the platform constitutes the basis on which to rest the correct execution of a code object. Thus, study relates to the design of a platform of a general practitioner grouping together a set of modules. Each module corresponds to a set of associated functionalities. We will have the following different modules:

1. Appointment management module
2. Consultation management module
3. Patient medical records management module
4. Human resources management module
5. Hospitalization and logistics management
6. Financial management module linked to consultations and hospitalizations

KEYWORDS: PLATFORM, HEALTH

TABLE DES MATIERES

DEDICACES	1
REMERCIEMENTS	2
RÉSUMÉ	3
ABSTRACT	4
TABLE DES MATIERES	5
LISTE DES FIGURES	7
LISTE DES TABLEAUX	8
LISTE DES ABREVIATIONS :	9
INTRODUCTION GENERALE	10
Partie 1: Etude Préalable	12
CHAPITRE1 : Contexte et clarification des concepts	12
1.1.1. Contexte Général :	12
1.1.2. Clarification des concepts :	12
1.1.2.1. Conception :	12
1.1.2.2. Application :	12
1.1.2.3. Structure sanitaire généraliste :	14
1.1.2.4. La gestion des rendez-vous des patients :	14
1.1.2.5. La gestion des consultations :	15
1.1.2.6. La gestion des dossiers médicaux des patients :	15
1.1.2.7. La gestion des ressources humaines :	15
1.1.2.8. La gestion des hospitalisations et de la logistique :	15
1.1.2.9. La gestion financière :	15
CHAPITRE 2 : Problématique et étude des besoins	16
1.2.1. Problématique	16
1.2.2. Etude des besoins et état de l'art :	17
1.2.2.1 Besoins informationnels :	17
1.2.2.2 Besoins technologiques :	17
1.2.2.3. Etat de l'art :	26
Partie 2 : Etude Pratique	29
CHAPITRE1 : Conception du Système d'information informatique	29
2.1.1. Modélisation:	29
2.1.1.1. Diagramme de classe.....	29
2.1.1.2. Diagramme de cas d'utilisation :	30

2.1.1.3.	Diagramme d'activité :.....	32
2.1.2.	Développement :	33
2.1.2.1.	Téléchargement et installation de Symfony sous Windows :	33
2.1.2.2.	Création d'une application Symfony :	38
2.1.2.3.	Démarrage application Symfony.....	39
2.1.2.4.	Squelette du projet.....	39
2.1.2.5.	Coût du projet.....	41
CHAPITRE 2 :	Présentation de la plateforme.....	42
3.	Conclusion et perspectives :	56
REFERENCES	57
BIBLIOGRAPHIES	58
ANNEXE 1 :	Extrait Modèle hospitalisation.....	59
ANNEXE 2 :	Vue hospitalisation.....	60
ANNEXE 3 :	Controller hospitalisation.....	61

LISTE DES FIGURES

Figure 1. Diagramme de classe	29
Figure 2. Cas d'utilisation hospitalisation	30
Figure 3. Cas d'utilisation consultation.....	31
Figure 4. Diagramme d'activité	32
Figure 5. Téléchargement Symfony	34
Figure 6.Choix dossier de destination	35
Figure 7. Choix du dossier d'installation	36
Figure 8. Ajout de Symfony aux variables d'environnement.....	37
Figure 9. Fin d'installation	38
Figure 10. Création application	38
Figure 11. Démarrage application	39
Figure 12. Squelette du projet.....	39
Figure 13. Page de connexion.....	42
Figure 14. Page récupération mot de passe	43
Figure 15. Page changement mot de passe.....	44
Figure 16. Page d'accueil.....	45
Figure 17. Page Liste des consultations.....	47
Figure 18. Page détail d'une consultation	47
Figure 19. Page Liste Hospitalisations	48
Figure 20. Page Modifier une Hospitalisation	48
Figure 21. Page Dossier Patients.....	49
Figure 22. Page Patient	50
Figure 23. Page Création Facture.....	51
Figure 24. Page Modification/Suppression facture	52
Figure 25. Page Personnel	53
Figure 26. Page Rendez-vous	54
Figure 27. Page Création Utilisateur.....	55
Figure 28. Page Modification/Suppression Utilisateur	55
Figure 29. Modèle hospitalisation (extrait).....	59
Figure 30. Vue hospitalisation.....	60
Figure 31. Controller hospitalisation.....	61

LISTE DES TABLEAUX

Tableau 1. Tableau comparatif PHP/PHYTON	21
Tableau 2 . Tableau comparatif PHP/JAVA	22
Tableau 3. Tableau comparatif de 3 solutions de gestion hospitalière.....	27

LISTE DES ABREVIATIONS :

- OS : Operating System
- GRH : Gestion des ressources humaines
- HTML : HyperText Markup Language
- CSS : Cascading Style Sheets
- MVC : Modèle-vue-contrôleur
- PHP : Hypertext Preprocessor
- MIT : Massachusetts Institute of Technology
- ORM : object-relational mapping
- MTV : Model - Template - View
- XML : Extensible Markup Language
- URL : Uniform Resource Locator
- API : Application Programming Interface
- CRUD : create, read, update, delete
- SQL : Structured Query Language
- SGBD : Système de Gestion de Bases de Données
- GPL : GNU General Public License
- SGBDR : Système de Gestion de Bases de Données Relationnelles
- UML : Unified Modeling Language
- VSCode : Visual Studio Code
- FTP: File Transfer Protocol
- PNG : Portable Network Graphics
- IPM : Institution de prévoyance maladie

INTRODUCTION GENERALE

1. Contexte Général :

Avec l'accroissement des structures sanitaires et la prolifération des maladies, les organisations sanitaires doivent être dotées d'un système de gestion numérique de leurs activités et d'une base de données centralisée afin de pouvoir piloter au mieux le processus de suivi des patients.

C'est dans ce contexte qu'on est amené à se poser la question à savoir est-ce qu'il existe une application performante pour la gestion d'une structure de santé généraliste telle qu'une clinique ?

En effet les organismes sanitaires rencontrent d'énormes difficultés dans la gestion de leur fil d'attente dû au manque d'organisation et de suivi des patients qui y sont traités. Aujourd'hui rare sont les structures qui peuvent vous fournir vos historiques de consultations, car généralement tout est répertorié derrière une énorme paperasse. Ce qui génère une panoplie de données difficilement utilisables.

Cet état actuel peut se justifier par la lente transition vers les systèmes modernes (numériques), à un refus de modernisation, à une crainte d'un nouveau système qui changera leur méthode de travail ainsi qu'à des employés non-qualifiés pour un tel changement.

Par ailleurs le système actuel bien que non optimal pourrait être amélioré en intégrant des archivistes documentaristes capables de mettre en ordre toute cette paperasse et des assistantes qualifiées pour l'utilisation des ordinateurs. Mais la solution idéale est une application qui permettra aux organismes sanitaires d'avoir une bonne gestion de leur structure.

2. Problématique de la recherche :

Notre étude portera sur la conception d'un système adapté au besoin des structures de santé. Comme évoqué dans le contexte général, la plupart des structures sanitaires font face à un gap énorme par rapport aux solutions modernes.

Ainsi on pourrait se demander comment pallier à ce gap ? Comment optimiser le mode de travail actuel ? Comment mettre l'information au cœur du système ? La philosophie du « Paperless » ne devrait-il pas être adoptée ? Le partage d'information entre les structures de santé via des plateformes numériques ne serait-il pas un bond en avant ? L'investigation des symptômes via des cas similaires ne pourrait-il pas être basée sur l'historique des centres

sanitaires ? La liste est non-exhaustive. Bon nombre de questionnement qui mérite qu'on s'y penche. Notre analyse sera basée sur tous ces problèmes dont la solution se trouve derrière une étude approfondie de l'existant, de ce que l'on peut améliorer et de ce qui mérite une profonde transformation.

3. Objectif et intérêt :

L'objectif de notre étude est la conception d'une application qui embarque les 6 modules ci-dessous qui permettent de :

- ✓ Gérer les rendez-vous des patients.
- ✓ Gérer les consultations.
- ✓ Gérer les dossiers médicaux des patients.
- ✓ Gérer les ressources humaines.
- ✓ Gérer les hospitalisations et la logistique.
- ✓ Gérer les factures.

L'intérêt de notre étude est de permettre aux structures de santé d'avoir :

- ✓ Un système d'information sécurisé.
- ✓ Un temps de recherche rapide comparé au système actuel.
- ✓ Une méthode de travail plus organisée.

4. Annonce du plan :

Notre mémoire comporte deux parties. La première est une étude préalable constituée de deux chapitres dans lesquels nous allons expliquer les concepts, faire une étude des besoins et parler de la problématique de notre sujet. Dans la seconde partie nous ferons une étude pratique constituée également de deux chapitres qui traiteront la conception de notre système d'information.

Partie 1: Etude Préalable

CHAPITRE1 : Contexte et clarification des concepts

1.1.1. Contexte Général :

Aujourd'hui, nous remarquons que les structures sanitaires sont envahies d'un nombre considérable de patients chaque jour. Cette situation entraîne de longues queues au niveau des différents services, un personnel débordé de travail et une organisation du travail qui devient une tâche fastidieuse.

C'est dans ce contexte que nous avons songé à créer une plateforme de gestion de l'ensemble des services d'une clinique pluridisciplinaire. Nous développerons d'abord un module pour la gestion des ressources humaines c'est-à-dire de l'ensemble du personnel de la clinique ; toute personne travaillant dans la clinique aura ses informations stockées au niveau d'une base de données centralisée pour une bonne traçabilité. Nous aurons ensuite à gérer les patients, leur dossier médical, à générer leurs factures en cas d'hospitalisation ou de consultation.

Pour un gain de temps précieux, une prise de rendez-vous à distance pourra se faire ce qui permettra au patient de choisir le jour ainsi que l'heure de son rendez-vous. Le médecin de son côté pourra savoir combien de patients il recevra quotidiennement.

Cette plateforme représente une valeur ajoutée considérable dans la gestion de l'organisation des structures de santé, l'historisation et le suivi des consultations ou des hospitalisations. Elle permet d'avoir une vision macro de toute l'activité.

1.1.2. Clarification des concepts :

Dans ce chapitre, nous allons clarifier les différents concepts énumérés dans notre document

1.1.2.1.Conception :

La conception est la phase créative d'un projet d'ingénierie. Le but premier de la conception est de permettre de créer un système ou un processus répondant à un besoin en tenant compte des contraintes. Le système doit être suffisamment défini pour pouvoir être installé, fabriqué, construit et être fonctionnel pour répondre aux besoins du client.

1.1.2.2.Application :

Une « application » est un programme directement utilisé pour réaliser une tâche et/ou assurer un ensemble de fonctions précises. Notons qu'un programme est tout simplement un ensemble

d'instructions destiné à être exécuter par un Ordinateur. Une instruction peut être par exemple une information communiquée. Cette information peut être à la fois une commande et/ou une explication pour décrire une action, un comportement, ou une tâche qui devra s'exécuter.

Au fil du temps, 3 grandes familles d'application ont vu le jour avec chacun sa particularité et son domaine d'application.

❖ **Une application mobile :**

Une application mobile est un logiciel applicatif développé pour un appareil électronique mobile, tel qu'un assistant personnel, un téléphone portable, un Smartphone, un baladeur numérique, une tablette tactile, ou encore certains ordinateurs fonctionnant avec le système d'exploitation Windows Phone ou Chrome OS. Elles sont pour la plupart distribuées depuis des plateformes de téléchargement (parfois elles-mêmes contrôlées par les fabricants de Smartphones) telles que l'App Store (plateforme d'Apple), le Google Play (plateforme de Google / Androïdes), ou encore le Microsoft Store (plateforme de Microsoft pour Windows 10 Mobile). Mais des applications peuvent aussi être installées sur un ordinateur, grâce par exemple au logiciel iTunes distribué par Apple pour ses appareils. Les applications distribuées à partir des magasins d'applications sont soit payantes, soit gratuites, mais généralement avec des publicités.

❖ **Une application web :**

En informatique, une application web (aussi appelée web application, de l'anglais) est une application manipulable directement en ligne grâce à un navigateur web et qui ne nécessite donc pas d'installation sur les machines clientes, contrairement aux applications mobiles. De la même manière que les sites web, une application web est généralement installée sur un serveur et se manipule en actionnant des widgets à l'aide d'un navigateur web, via un réseau informatique (Internet, intranet, réseau local, etc.). Exemples: Les messageries web, les systèmes de gestion de contenu, les wikis et les blogs sont des applications web.

Les moteurs de recherches, les logiciels de commerce électronique, les jeux en ligne, les logiciels de forum, peuvent être sous forme d'application

❖ **Une application desktop :**

Une application desktop est une application qui s'exécute complètement sur un seul PC et donc n'a pas besoin d'un accès Web pour fonctionner.

L'un de ses avantages est qu'elle est plus rapide et plus stable car ne dépend pas des performances d'une autre application. En revanche, son inconvénient c'est sa portabilité car elle a besoin d'être installée sur chaque machine pour être accessible. De plus, elles font face aux problèmes de compatibilité de système d'exploitation.

1.1.2.3. Structure sanitaire généraliste :

C'est un établissement de santé qui ne se limite pas à un domaine spécifique. En quelque sorte, c'est un établissement pluridisciplinaire où l'on retrouve entre autres une pédiatrie, une dermatologie etc... Il ne se limite donc pas à un seul domaine de la santé.

1.1.2.4. La gestion des rendez-vous des patients :

Une planification des rendez-vous pour les médecins et les différents services de la clinique sera développée dans ce module pour permettre une bonne gestion de la file d'attente car chaque patient devra venir à une heure qui a été choisie lors de la prise de rendez-vous pour une meilleure organisation du quotidien des médecins.

1.1.2.5.La gestion des consultations :

La consultation est un entretien d'un médecin avec son client, au cours duquel il le questionne, l'ausculte en vue de diagnostiquer une maladie pour vue de lui prescrire un traitement.

Ainsi, sur la plateforme, le médecin traitant pourra indiquer le type de pathologie du patient, les informations de ce dernier (nom, prénom, adresse, date de naissance, tension artérielle...), le montant de la consultation, si le patient dispose d'une prise en charge ...

1.1.2.6.La gestion des dossiers médicaux des patients :

« Le dossier médical est un ensemble de documents (physiques ou informatisés) qui retrace des épisodes ayant affecté la santé de cette personne : lettres, notes, compte rendu, résultats de laboratoire, film radiologique etc... », le médecin pourra bien suivre ses patients.

1.1.2.7.La gestion des ressources humaines :

« C'est l'ensemble des pratiques mises en œuvre pour administrer, mobiliser et développer les ressources humaines impliquées dans l'activité d'une organisation. », c'est pour cela que nous avons jugé nécessaire de mettre en place un module de GRH qui se chargera de la gestion du personnel de la clinique.

1.1.2.8.La gestion des hospitalisations et de la logistique :

Elle permet de gérer les besoins hospitaliers tels que la chambre d'hospitalisation en fonction de la disponibilité et du coût, le régime alimentaire, le numéro d'admission du patient, les décharges, les transferts... En cas d'accident ou d'accouchement on pourra vérifier la disponibilité des chambres avant d'accepter des patients.

1.1.2.9.La gestion financière :

Elle permet une gestion du paiement relatif à l'hospitalisation, à la consultation, aux soins infirmiers etc. Ainsi si le patient dispose d'une prise en charge elle ne payera qu'un pourcentage et le reste sera verser par son assurance au cas échéant il paie le montant fixé par la clinique.

CHAPITRE 2 : Problématique et étude des besoins

1.2.1. Problématique

Comme évoqué dans l'introduction, nous devons voir si notre solution digitale permet de pallier le gap constaté et d'optimiser le mode de travail actuel. En effet l'idée est de mettre l'information au cœur du système. Une information fiable et hautement disponible, un gain de temps et d'énergie. La solution digitale devra non seulement régler le problème de la masse énorme de paperasse mais aussi doit être un outil d'aide à la prise de décision.

Avec le mode de travail actuel, pour prendre un rendez-vous téléphonique, l'assistant vous signale au bout du fil que la réception se fera par ordre d'arrivée. Du coup une perte de temps énorme est déjà à prévoir en faisant la queue. Une fois la personne arrivée au niveau de la clinique, l'assistant vous inscrit sur la fiche des consultations ou sur la fiche des rendez-vous, une autre perte de temps qui aurait pu être évité.

D'où l'importance une fois de plus de mettre en œuvre une application web centralisée accessible à tout le personnel de l'établissement.

Dans certains cas le dossier médical peut être perdu. Avec notre solution, ce genre de situation ne se produira pas car la base de données sera hautement sécurisée et sauvegardée, aucune perte d'informations ne sera permise.

Autre point essentiel, est qu'on aura toutes les informations relatives au patient en un clic pas besoins de chercher derrière une longue paperasse.

De nos jours, l'informatisation du système d'information d'une clinique ou de toute autre structure de santé devient incontournable.

1.2.2. Etude des besoins et état de l'art :

1.2.2.1 Besoins informationnels :

Dans notre étude on aura besoin de côtoyer des médecins, des employés chargés de l'accueil au sein de la clinique pour qu'ils nous expliquent leur mode de travail en interne. Ainsi on aura toutes les informations nécessaires pour une bonne conception de notre application de gestion d'une clinique généraliste, une autorisation d'aller au sein de la structure pour voir un peu comment l'activité se déroule réellement ce qui constituera une réelle valeur ajoutée à notre étude.

1.2.2.2 Besoins technologiques :

Pour les besoins technologiques, il faudra :

1.2.2.2.1. Un langage de programmation pour le Back-end :

Un langage de programmation est un langage informatique (Turing-complet), permettant à un être humain d'écrire un programme informatique (le code source) destiné à être exécuté par une machine, généralement un ordinateur. Le code source subit une transformation ou une évaluation dans une forme exploitable par la machine, ce qui permet d'obtenir un programme exécutable. Les langages permettent souvent de faire abstraction des mécanismes de bas niveau de la machine, de sorte que le code source puisse représenter une solution telle que comprise ou conçue par un être humain.

Pourquoi l'utilisation d'un Framework ?

Un Framework est un regroupement de briques logicielles qui servent de fondations à un nouveau projet.

Le but de la mise à disposition d'un Framework peut être multiple. En général, il s'agit de fournir une base des fonctionnalités courantes d'un langage de programmation afin de ne pas avoir à réinventer la roue lors de son utilisation.

Si on parle d'un langage côté serveur, le Framework embarquera des modules qui permettront de gérer les connexions aux bases de données, la validation des données entrées par les

utilisateurs et bien d'autres choses encore. Cela permet au développeur de s'appuyer sur les fonctionnalités souvent solidement approuvées par la communauté gravitant autour du développement et de l'utilisation du Framework et d'ainsi économiser un temps précieux à ne pas réinventer les fonctions de bases communes à un très grand nombre de projets.

Si on parle de HTML, de JavaScript ou de CSS, le Framework propose en règle générale toute une panoplie de composants permettant là encore de ne pas avoir à tout programmer de zéro. Le plus souvent, ils sont accompagnés de composants prêts à être utilisés directement dans le projet et qui permettront d'obtenir un style visuel cohérent (bouton, menu, barre de navigation ...)

Il existe un très grand nombre de Framework, chacun ayant ses spécificités d'abord gouvernées par le ou les langages de programmation.

❖ **JAVA avec le Framework Spring MVC :**

Spring est un Framework libre pour construire et définir l'infrastructure d'une application Java, dont il facilite le développement et les tests.

Spring MVC est l'un des plus anciens Framework Web Java, mais c'est également l'un des meilleurs. À ce jour, il est utilisé, car il s'adapte et améliore constamment des changements et des développements spécifiques avec Java.

Pour les ingénieurs en logiciel de tous types, il offre un Framework pour développer des applications Web et les configurer, ainsi que pour développer les fonctionnalités de sécurité qui vont avec. Il s'agit d'un Framework Web large et expansif, plus que capable d'assumer toute tâche ou projet potentiel sur lequel vous souhaitez travailler.

Grâce à la modularité des outils eux-mêmes, cela vous permet d'écrire du code très propre et accessible. Il existe une énorme quantité d'excellents documents et une communauté florissante qui vous aidera si vous avez des questions ou des préoccupations concernant la manière de faire certaines choses ou comment certaines choses fonctionnent ou quelque chose du genre.

Le plus gros inconvénient, et le seul véritable inconvénient, est le fait qu'il s'agit d'un Framework Web complexe et que, si vous débutez en programmation Java, ce n'est probablement pas le meilleur choix, tout simplement parce qu'il nécessite beaucoup de connaissances à l'avance.

❖ **PHP avec le Framework Symfony :**

Symfony est un Framework très populaire, français, et très utilisé dans le milieu des entreprises. Développé en PHP, sous licence MIT, Symfony est reconnu comme le Framework PHP open source leader sur le marché.

Lancé il y a plus d'une dizaine d'années par l'éditeur français SensioLabs, il a aujourd'hui acquis une très large popularité et bénéficie d'une communauté internationale très active.

À force de toujours recréer les mêmes fonctionnalités de gestion d'utilisateurs, gestion ORM, etc., SensioLabs a développé ce Framework pour ses propres besoins.

Les projets les plus connus utilisant Symfony sont eZ Publish (utilisation de Symfony 3 en full stack), Drupal depuis la version 8 (utilisation de composants Symfony), phpBB, Laravel, Joomla!, Composer, Magento, Piwik11, PrestaShop depuis la version 1.7 (utilisation de composants Symfony).

Symfony utilise la version 5 de PHP (PHP \geq 5.2.4 pour Symfony 1.x, PHP \geq 5.3.3 pour Symfony 2.x et PHP \geq 5.5.9 pour Symfony 3.x).

Symfony 4.0 requiert la version 7.1.3 ou supérieur afin de fonctionner.

Symfony (1.x) propose entre autres :

- une séparation du code en trois couches, selon le modèle MVC, pour une plus grande maintenabilité et évolutivité ;
- des performances optimisées et un système de cache afin d'assurer des temps de réponse optimaux ;
- une gestion des URL parlante, permettant à une page d'avoir une URL distincte de sa position dans l'arborescence ;
- un système de configuration en cascade utilisant pleinement le langage de description YAML ;
- un générateur de back-office et un lanceur de module (scaffolding) ;
- l'internationalisation native ;
- le support d'AJAX ;
- une architecture extensible permettant créations et utilisations de plugins.

Symfony fournit une interface en ligne de commande pour améliorer la productivité en créant un code de base modifiable à volonté.

Symfony est très modulaire. C'est un ensemble de composants, faiblement couplés, réunis sous la forme d'une distribution. L'édition standard offre un Framework full-stack incluant un ORM (Doctrine), le moteur de templating Twig, la gestion des emails avec SwiftMailer ou encore un composant de sécurité pour la gestion de l'authentification utilisateur et des permissions.

Le respect du protocole HTTP et du paradigme MVC sont au cœur du Framework. Le tout pour fournir aux développeurs un contexte de développement exhaustif avec un outil complet de débogage. Accès uniformisé à la base de données, aux contrôleurs, aux vues... le Framework fournit également une large bibliothèque de fonctions utilitaires.

La version 5.0 est sortie en novembre 2019.

❖ PYTHON avec le Framework Django

Django est un cadre de développement web open source en Python. Il a pour but de rendre le développement web 2.0 simple et rapide. Pour cette raison, le projet a pour slogan « Le Framework pour les perfectionnistes avec des deadlines. ».

Django est un Framework qui s'inspire du principe MVC ou MTV (la vue est gérée par un Template) composé de 3 parties distinctes :

- Un langage de Template flexible qui permet de générer du HTML, XML ou tout autre format texte ;
- Un contrôleur fourni sous la forme d'un "remapping" d'URL à base d'expressions rationnelles ;
- Une API d'accès aux données est automatiquement générée par le Framework compatible CRUD. Inutile d'écrire des requêtes SQL associées à des formulaires, les requêtes SQL sont générées automatiquement par l'ORM.
- En plus de l'API d'accès aux données, une interface d'administration fonctionnelle est générée depuis le modèle de données. Un système de validation des données entrées par l'utilisateur est également disponible et permet d'afficher des messages d'erreurs automatiques.

Tableau 1. Tableau comparatif PHP/PYTHON

Paramètres	PHP	PYTHON
Facilité d'apprentissage	Temps apprentissage long	Beaucoup plus facile à apprendre
Soutien communautaire	Excellent support communautaire	Excellent support communautaire
Tarifcation	Entièrement gratuit et open source	Entièrement gratuit et open source
Vitesse	PHP 7.x est extrêmement rapide	PHP est dominant pour ce paramètre

Tableau 2 . Tableau comparatif PHP/JAVA

Paramètres	PHP	JAVA
EXÉCUTION DU CODE	Le code PHP peut être exécuté de plusieurs façons, car il peut incorporer du code PHP dans du code HTML à l'aide de la balise de script ou exécuter le code PHP à l'aide du système de gestion de contenu .	D'un autre côté, le code source Java est compilé en bytecode qui est exécuté sur la machine virtuelle Java. JVM est exécutée par le processeur sur lequel le bytecode Java fonctionne correctement.
SÉCURITE	Bien que les deux langages offrent autant de sécurité à son utilisateur, PHP manque de certaines des fonctionnalités de sécurité intégrées par rapport à un autre langage de programmation. Par conséquent, les programmeurs doivent utiliser les fonctionnalités de sécurité fournies par les différents cadres de PHP pour protéger leur application d'entreprise.	Alors que Java est livré avec les fonctionnalités de sécurité avancées, il offre une sécurité complète à son utilisateur. Il permet aux développeurs de faire échanger les données client et serveur via les protocoles de transfert sécurisés. Pour développer une application de site Web ou une application d'entreprise, de nombreux développeurs préfèrent Java à PHP.
COÛT DE DÉVELOPPEMENT	PHP est un langage de programmation open source qui permet à l'utilisateur de développer n'importe quel code sans payer de frais.	D'un autre côté, Java n'est pas un langage de programmation open source. Il utilise un JDK ouvert qui est utilisé comme pour l'implémentation du langage de programmation Java. Donc, cela coûte plus cher qu'un projet PHP.

En définitive, notre choix porte sur PHP avec le Framework Symfony la version 4 pour le développement de notre application web. Parce que Symfony a une large communauté réactive, une riche documentation pour nous accompagner sur la réalisation de notre plateforme.

Ainsi, voulant évoluer dans le développement web avec nos projets de Drupal 8 que nous avons prévu de faire ultérieurement, Symfony reste un challenge étant donné que Drupal utilise des composants de Symfony ce projet sera bénéfique pour nous.

2.1.2.1.1. Une base de données MYSQL.

Une base de données est un « conteneur » stockant des données telles que des chiffres, des dates ou des mots, pouvant être retraités par des moyens informatiques pour produire une information; par exemple, des chiffres et des noms assemblés et triés pour former un annuaire. Les retraitements sont typiquement une combinaison d'opérations de recherches, de choix, de tri, de regroupement, et de concaténation.

C'est la pièce centrale d'un système d'information ou d'un système de base de données (ou base de données tout court), qui régit la collecte, le stockage, le retraitement et l'utilisation de données. Ce dispositif comporte souvent un logiciel moteur, des logiciels applicatifs, et un ensemble de règles relatives à l'accès et l'utilisation des informations.

Le système de gestion de base de données est une suite de programmes qui manipule la structure de la base de données et dirige l'accès aux données qui y sont stockées. Une base de données est composée d'une collection de fichiers ; on y accède par le SGBD qui reçoit des demandes de manipulation du contenu et effectue les opérations nécessaires sur les fichiers. Il cache la complexité des opérations et offre une vue synthétique sur le contenu. Le SGBD permet à plusieurs usagers de manipuler simultanément le contenu, et peut offrir différentes vues sur un même ensemble de données.

MySQL est SGBDR. Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

2.1.2.1.2. Pour le front-end, on utilisera :

Bootstrap est une infrastructure frontale gratuite pour un développement Web plus rapide et plus facile. Il comprend des modèles de conception basés sur HTML et CSS pour la typographie, les formulaires, les boutons, les tableaux, la navigation, les modaux, les carrousels d'images et bien d'autres, ainsi que des plugins JavaScript en option. Bootstrap vous donne également la possibilité de créer facilement des conceptions réactives.

2.1.2.1.3. Une application de modélisation en UML (StarUML) :

Le Langage de Modélisation Unifié est un langage de modélisation graphique à base de pictogrammes conçu pour fournir une méthode normalisée pour visualiser la conception d'un système. Il est couramment utilisé en développement logiciel et en conception orientée objet.

UML est utilisé pour spécifier, visualiser, modifier et construire les documents nécessaires au bon développement d'un logiciel orienté objet. UML offre un standard de modélisation, pour représenter l'architecture logicielle. Les différents éléments représentables sont :

- ✓ Activité d'un objet/logiciel
- ✓ Acteurs
- ✓ Processus
- ✓ Schéma de base de données
- ✓ Composants logiciels
- ✓ Réutilisation de composants

Grâce aux outils de modélisation UML comme (StarUML), il est également possible de générer automatiquement tout ou partie du code d'une application logicielle, par exemple en langage Java, à partir des divers documents réalisés.

2.1.2.1.4. Un éditeur (VS Code) :

VS Code est présenté lors de la conférence des développeurs Build d'avril 2015 comme un éditeur de code multi-plateforme, open source et gratuit, supportant une dizaine de langages.

Il est basé sur Electron, une structure utilisée pour déployer des applications Node.js pour le bureau exécuté sur le moteur Blink. Bien qu'il utilise le Framework Electron, le logiciel n'utilise pas Atom mais utilise le même composant éditeur (nommé "Monaco") utilisé dans Azure DevOps (anciennement appelé Visual Studio Online et Visual Studio Team Services).

Le code source est fourni sous la licence libre MIT (plus précisément la licence MIT) sur le site du projet sur GitHub. En revanche, l'exécutable est proposé sur le site officiel de Microsoft sous une licence propriétaire³.

2.1.2.1.5. Un logiciel de gestion de versions : Git

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.

Les logiciels de gestion de versions sont utilisés principalement par les développeurs ; ce sont donc bel et bien des outils pour geeks. En effet, ils sont quasi exclusivement utilisés pour gérer des codes sources, car ils sont capables de suivre l'évolution d'un fichier texte ligne de code par ligne de code.

Ces logiciels sont fortement conseillés pour gérer un projet informatique.

2.1.2.1.6. XAMPP

XAMPP est un ensemble de logiciels permettant de mettre en place un serveur Web local, un serveur FTP et un serveur de messagerie électronique. Il s'agit d'une distribution de logiciels libres (X (cross) Apache MariaDB Perl PHP) offrant une bonne souplesse d'utilisation, réputée pour son installation simple et rapide. Ainsi, il est à la portée d'un grand nombre de personnes puisqu'il ne requiert pas de connaissances particulières et fonctionne, de plus, sur les systèmes d'exploitation les plus répandus.

Il est distribué avec différentes bibliothèques logicielles qui élargissent la palette des services de façon notable : Open SSL, Expat (parseur XML) PNG, SQLite, zlib... ainsi que différents modules Perl et Tomcat. Nombre de ces extensions étant inutiles aux débutants, une version allégée — version lite — est en conséquence aussi proposée.

1.2.2.3. Etat de l'art :

Un logiciel de gestion hospitalière automatise les processus, y compris la facturation, les rendez-vous, la planification, la conformité réglementaire et les audits financiers au sein des structures sanitaires.

Après quelques recherches, nous avons trouvé des applications à peu près similaires à ce qu'on veut réaliser, nous pouvons citer : PetalMD, Smartsheet, athenaClinicals, Waystar, Availity, TheWorxHub... Parmi ces solutions, nous allons faire une étude comparative des 3 premiers.

1.2.2.3.1. athenaClinicals :

athenaClinicals aide les médecins à prodiguer des soins de haute qualité en organisant leur emploi du temps de soins selon une productivité clinique optimisée ; le logiciel se charge des services administratifs et de qualité pour le médecin afin d'assurer un paiement intégral, tout en faisant suivre toutes les ordonnances de médecins pour traiter l'ensemble du processus, des résultats et des soins de suivi. Les médecins profitent ainsi d'un accès intégré et constant aux bonnes pratiques du plus grand réseau clinique des États-Unis, aux directives du secteur et aux règles de qualité les plus à jour et rigoureuses.

1.2.2.3.2 Petal On-Call (PetalMD)

La plateforme PetalMD est une solution d'optimisation des performances pour les hôpitaux. La liste de disponibilités et les tâches cliniques sont consolidées, mises à jour et distribuées à tout le personnel hospitalier en temps réel de manière radicalement nouvelle. PetalMD assure la confidentialité des échanges entre les parties prenantes, l'archivage des données.

1.2.2.3.3. Smartsheet

Smartsheet est une plateforme d'exécution du travail en ligne qui permet aux organisations de soins de santé de suivre, de gérer et de rendre compte des informations en améliorant la collaboration, en assurant la sécurité des données et en augmentant la visibilité. Avec son format de feuille de calcul familial, elle aide les équipes à se lancer facilement et à collaborer efficacement. De plus, accédez aux renseignements personnels sur la santé, partagez-les et stockez-les en toute sécurité.

Tableau 3. Tableau comparatif de 3 solutions de gestion hospitalière

	athenaClinicals	Petal On-Call (PetalMD)	Smartsheet
Cibles	médecins, médecins-chefs de service, responsables cliniques, responsables de cabinet, administrateurs de bureau, directeurs des soins, médecins en chef, administrateurs d'hôpital, infirmiers praticiens, assistants-médecins	Convient aux gestionnaires travaillant dans des hôpitaux (chefs du personnel, directeurs informatiques, etc.) qui souhaitent centraliser les horaires des médecins, consolider les listes de disponibilités, sécuriser les communications et améliorer les opérations avec des tableaux de bord analytiques.	L'application de gestion de projet en ligne intuitive de Smartsheet est largement utilisée au sein des organisations pour suivre et gérer divers types de travail pour les équipes grandes et petites, depuis les grandes entreprises jusqu'aux PME.
Fonctionnalités	<ul style="list-style-type: none"> • Facturation médicale • Gestion des dossiers patients • Gestion des lits • Gestion des médecins • Gestion des patients hospitalisés • Gestion des réclamations • Gestion des soins ambulatoires • Gestion des stocks • Gestion des stratégies • Planification des rendez-vous 	<ul style="list-style-type: none"> • Gestion des dossiers patients • Gestion des médecins • Gestion des patients hospitalisés • Gestion des soins ambulatoires • Gestion des stratégies • Planification des rendez-vous 	<ul style="list-style-type: none"> • Gestion des dossiers patients • Gestion des lits • Gestion des médecins • Gestion des réclamations • Gestion des stocks • Gestion des stratégies
Essai gratuit	NON	NON	OUI

1.2.2.3.4. Qu'est-ce que notre plateforme apporte-elle de plus ?

Notre solution ne gère pas uniquement les médecins, elle permet une gestion de l'ensemble du personnel de la clinique, une gestion des consultations chose que ces 3 solutions énumérées précédemment ne proposent pas, ainsi qu'un tableau de bord qui permettra d'avoir une vision macro et une meilleure suivi données clé de performance, un système hautement sécurisé.

De plus, notre plateforme est multiplateforme, et adopte le Responsive Design, elle peut être accéder à partir d'un smartphone, d'une tablette, d'un PC... Une bonne bande passante nous permet aussi de gérer la connexion simultanée des utilisateurs, de ce fait n'y aura pas de lenteur.

Cependant, une formation et une documentation seront disponibles, pour une prise en main rapide.

Notre plateforme dans sa version finale permettra un système de rappel de rendez-vous aux patients 12h avant ainsi que beaucoup d'autres fonctionnalités.

Partie 2 : Etude Pratique

CHAPITRE1 : Conception du Système d'information informatique

2.1.1. Modélisation:

2.1.1.1. Diagramme de classe

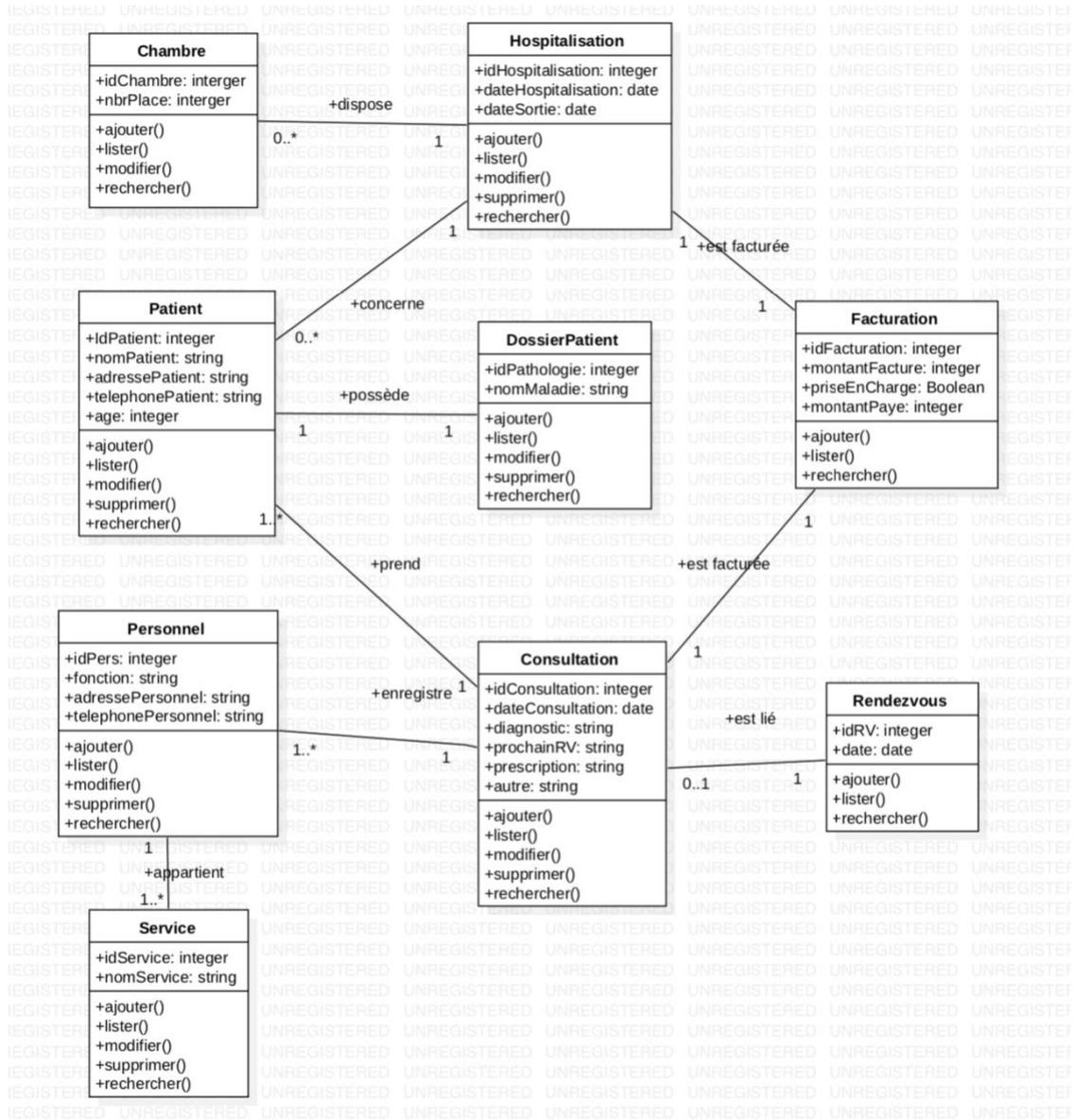


Figure 1. Diagramme de classe

2.1.1.2. Diagramme de cas d'utilisation :

2.1.1.2.1. Hospitalisation :

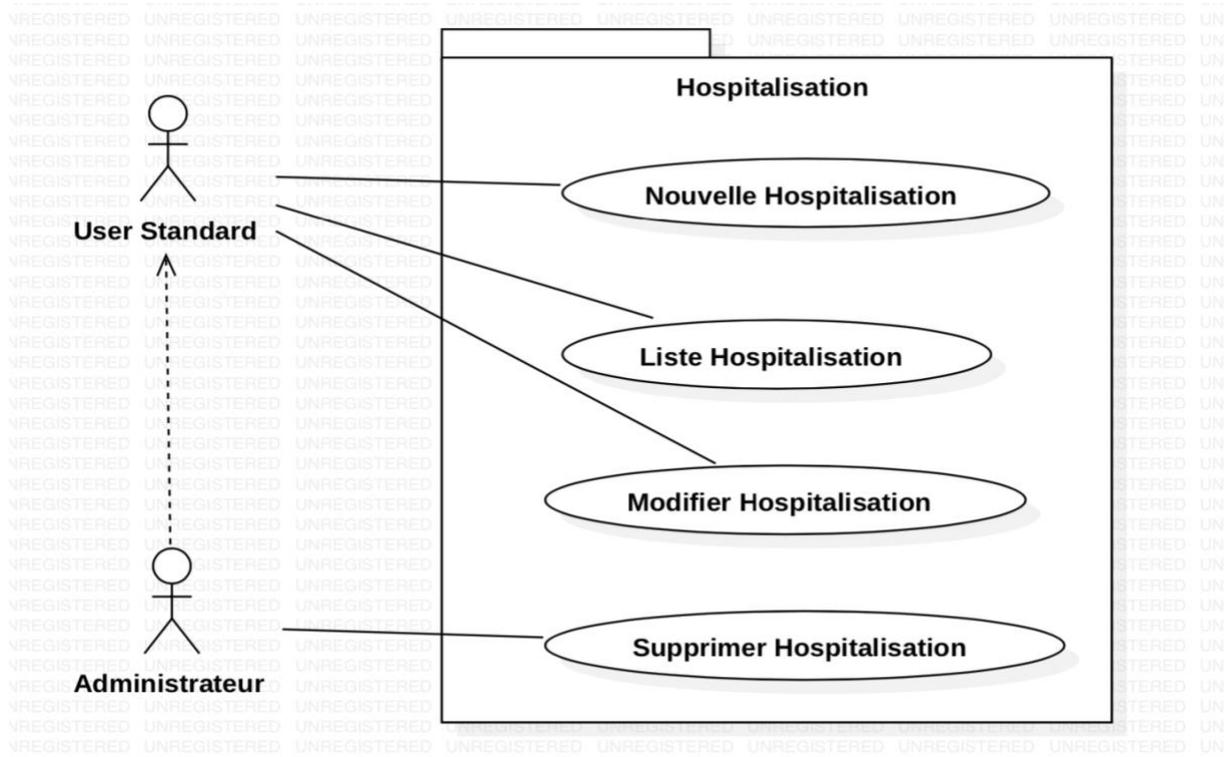


Figure 2. Cas d'utilisation hospitalisation

Un administrateur est un utilisateur standard mais tous les utilisateurs standards ne sont pas des administrateurs.

Un utilisateur standard pourra faire les opérations suivantes :

- Enregistrer une nouvelle consultation.
- Afficher la liste des hospitalisations.
- Modifier une hospitalisation.

En revanche, seul l'administrateur aura le privilège de supprimer une hospitalisation.

2.1.1.2.2. Consultation :

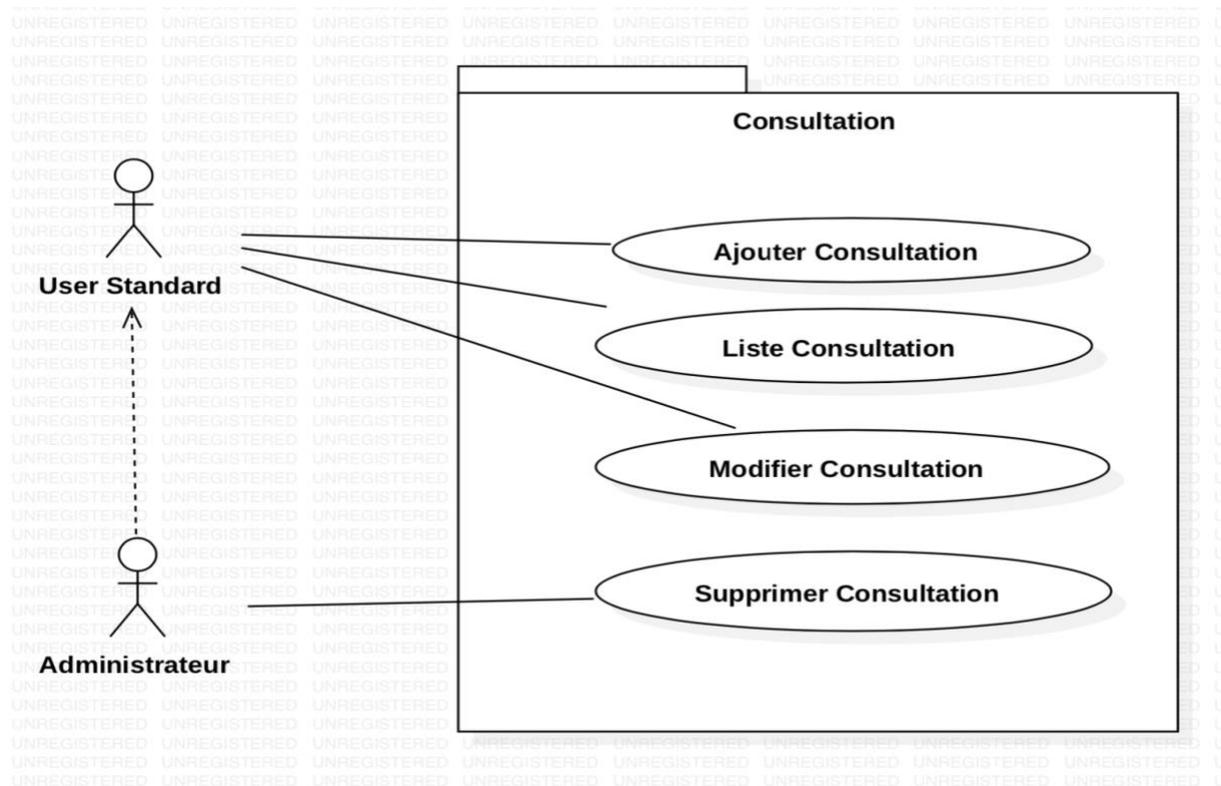


Figure 3. Cas d'utilisation consultation

De même que pour la consultation un utilisateur standard pourra :

- Ajouter une consultation.
- Lister les consultations.
- Modifier une consultation.

La suppression d'une consultation ne se fera que par l'administrateur.

2.1.1.3. Diagramme d'activité :

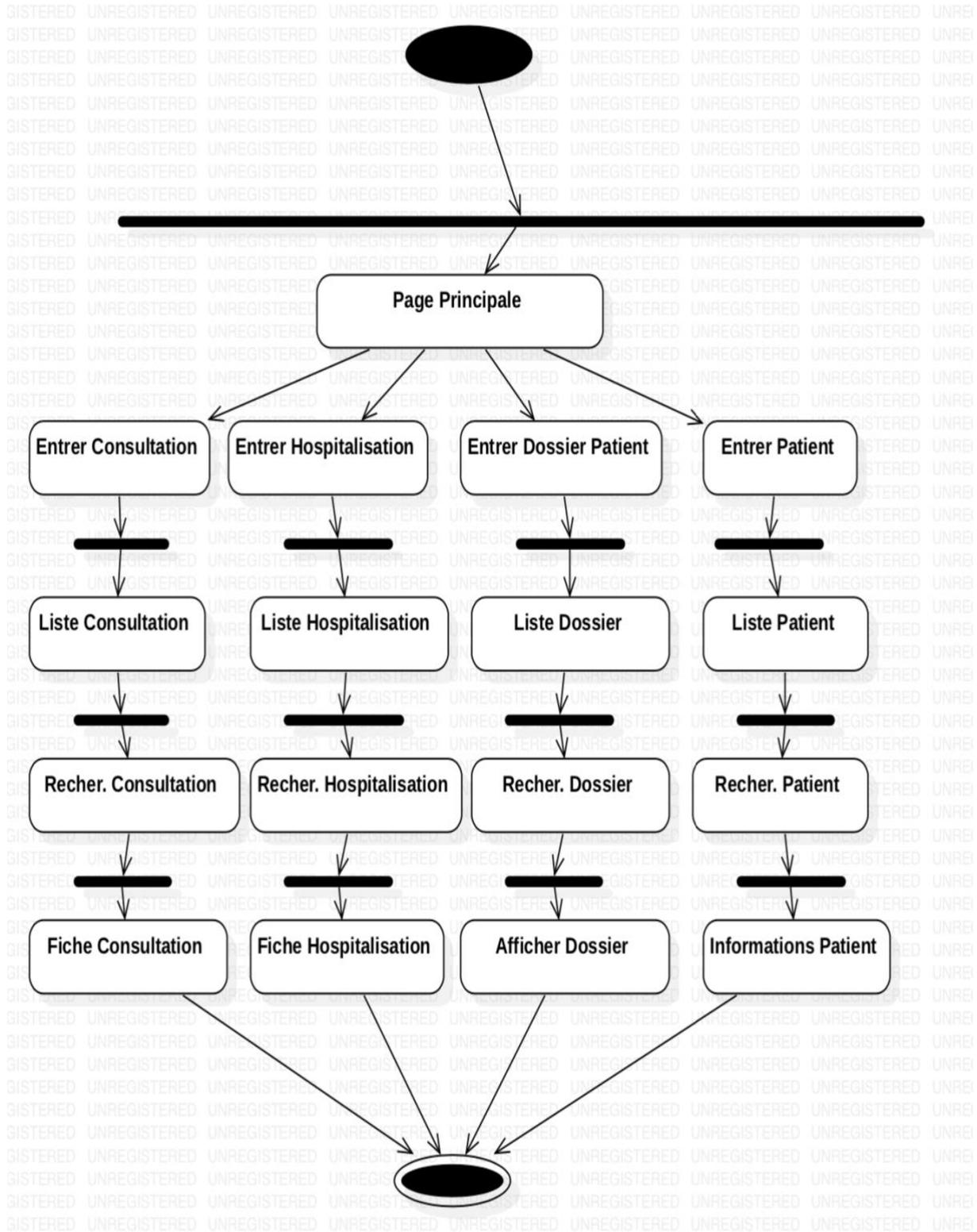


Figure 4. Diagramme d'activité

Comme l'illustre notre diagramme d'activité, le workflow de notre plateforme est le suivant :

- ✓ Après une authentification réussie, l'utilisateur accède à la page d'accueil avec tous les modules de la plateforme.
- ✓ Lorsqu'il clique sur un item exemple consultation, il verra un tableau avec une liste de toutes les consultations. Il pourra effectuer les actions telles que modifier, afficher et créer et même rechercher une consultation.
- ✓ Il en est de même pour tous les autres modules. Sauf pour la gestion des utilisateurs, seul un administrateur peut la voir et y accéder.
- ✓ Un lien de déconnexion aussi est intégrer pour permettre à l'utilisateur de quitter l'application.

2.1.2. Développement :

2.1.2.1. Téléchargement et installation de Symfony sous Windows :

2.2.10. Téléchargement

Pour créer notre application Symfony il faut :

- Installer PHP 7.2.5 ou supérieur et ces extensions PHP (qui sont installées et activées par défaut dans la plupart des installations PHP 7).
- Télécharger et Installer Composer, qui est utilisé pour installer les packages PHP.
Lien de téléchargement : <https://getcomposer.org/download/>
- Télécharger et Installer Symfony, qui crée sur notre ordinateur un binaire appelé Symfony qui fournit tous les outils dont nous avons besoin pour développer notre application localement.

Lien de téléchargement : [**https://symfony.com/download**](https://symfony.com/download)

- Cliquez sur télécharger

Télécharger Symfony

LINUX MACOS **LES FENÊTRES** Changelog

Télécharger setup.exe

Si vous préférez, vous pouvez télécharger le binaire directement: plates - formes 32 bits ou 64 bits .

► SHA-256 checksums du programme d'installation et des binaires

[Contrat de licence utilisateur final](#) du binaire Symfony.

Le symfony binaire créé par le programme d'installation fournit:

- La meilleure façon de [créer de nouvelles applications Symfony](#)
- Un puissant [serveur web local](#) pour développer vos projets
- Un outil pour [vérifier les failles de sécurité](#)

Figure 5. Téléchargement Symfony

2.2.10. Installation

- Choisir le dossier de destination

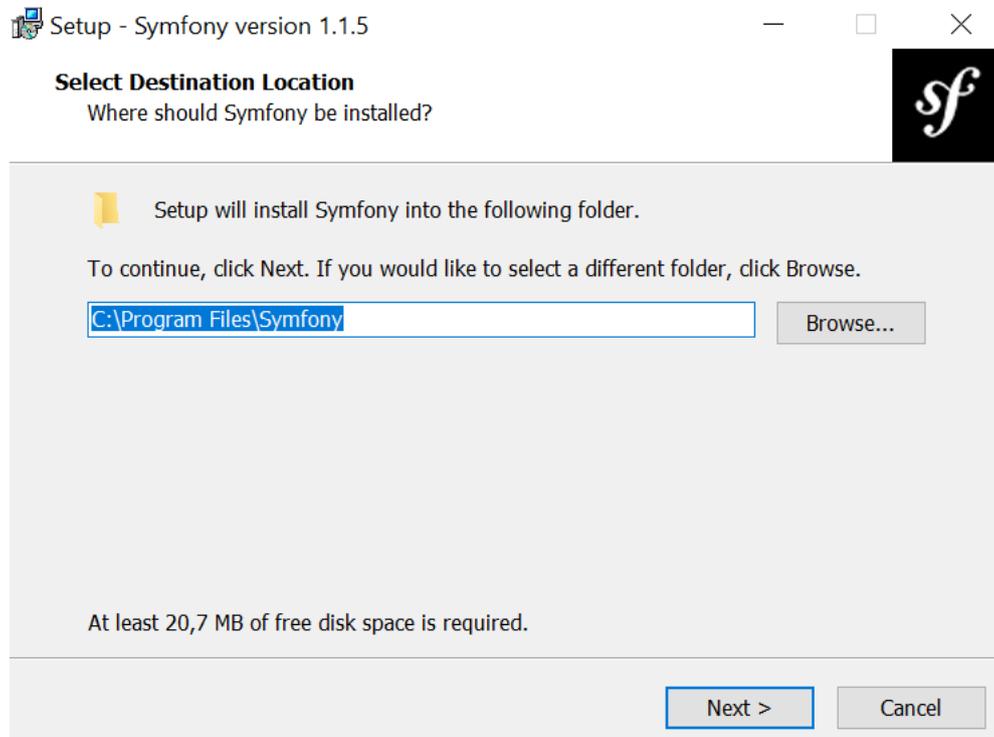


Figure 6. Choix dossier de destination

- Cliquez sur suivant

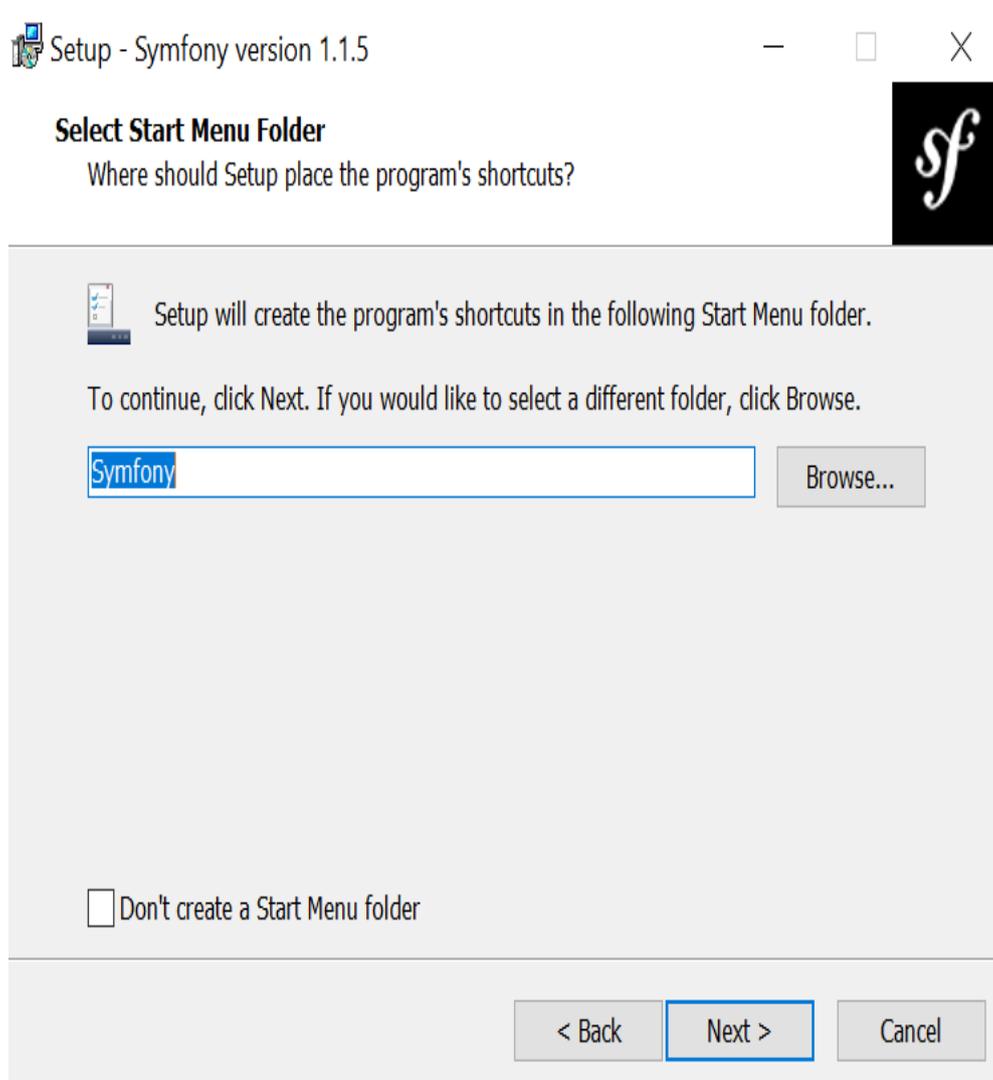


Figure 7. Choix du dossier d'installation

- Symfony est ajouté aux variables d'environnement pour être accessible globalement, cliquez sur suivant

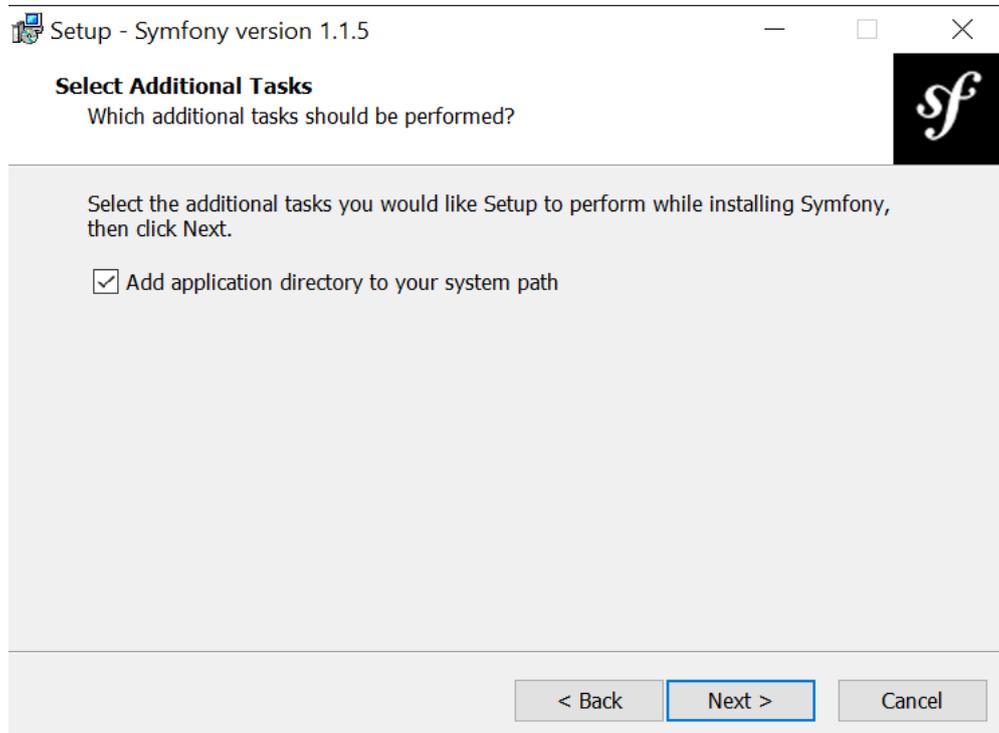


Figure 8. Ajout de Symfony aux variables d'environnement

- Installation terminée, nous pouvons commencer la vérification des prérequis.

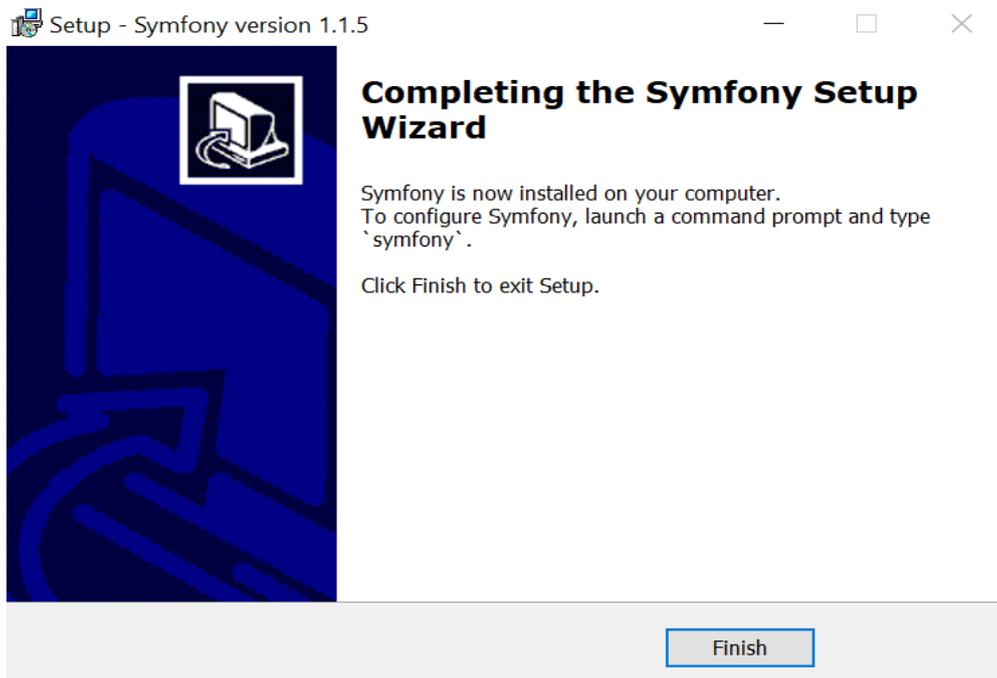


Figure 9. Fin d'installation

Le Symfony binaire fournit un outil pour vérifier si notre ordinateur répond à ces exigences. Pour ce faire il faut exécuter la commande suivante : **Symfony check:requirements**

2.1.2.2. Création d'une application Symfony :

```
# run this if you are building a traditional web application
> composer create-project symfony/website-skeleton my_project_name

# run this if you are building a microservice, console application or API
> composer create-project symfony/skeleton my_project_name
```

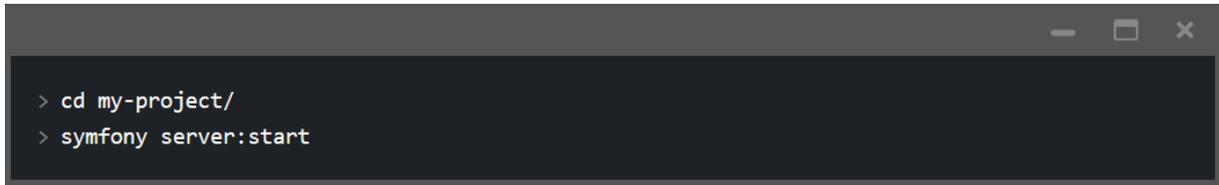
Figure 10. Création application

Ces deux commandes permettent de créer notre application Symfony.

La seule différence entre ces deux commandes est le nombre de packages installés par défaut. La première installe tous les packages dont vous avez généralement besoin pour créer des applications Web, de sorte que la taille de l'installation sera plus grande.

2.1.2.3. Démarrage application Symfony

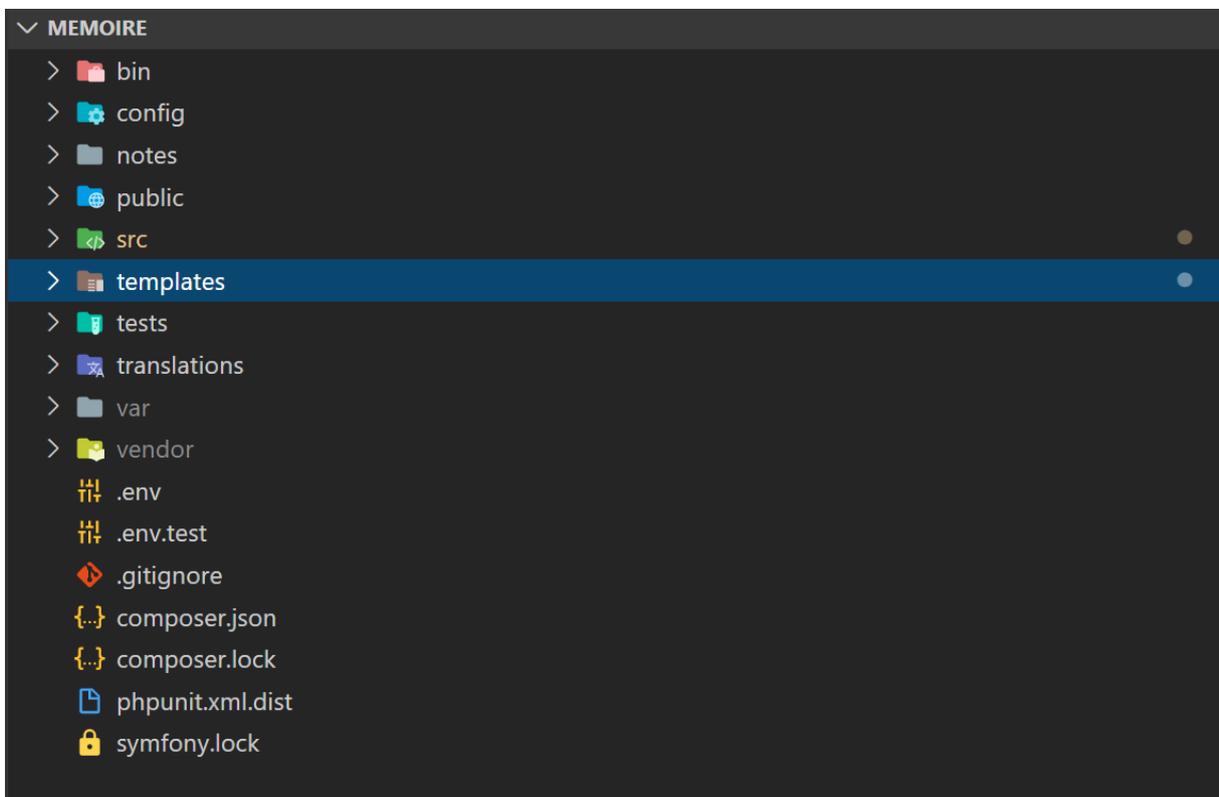
Tout d'abord il faut se positionner dans la racine du projet, puis démarrer le server de Symfony ou celui interne du PHP avec (**php bin/console server:run**).



```
> cd my-project/  
> symfony server:start
```

***Figure 11.** Démarrage application*

2.1.2.4. Squelette du projet



***Figure 12.** Squelette du projet*

/bin : Ce répertoire contient tous les exécutables (commandes PHP) dont nous allons nous servir pendant le développement.

`/Config` : Ce répertoire contient toute la configuration de notre site. C'est ici que nous configurerons Symfony lui-même, mais aussi les plug-ins (ou bundles) que nous installerons par la suite.

`/public` : Ce répertoire contient tous les fichiers images, fichiers CSS et JavaScript, etc. Il contient également le contrôleur frontal (`index.php`).

`/src` : Dans ce répertoire, nous avons le code source de notre application. Il contient les dossiers :

- `/Controller` (pour nos Controllers),
- `/Entity` (pour nos modèles), `/Form` (pour nos classes de formulaires),
- `/Migrations` (pour création des tables / schémas de base de données),
- `/Repository` (contient les requêtes pour interroger notre base de données).

`/templates` : Il contient tous les templates de notre application c'est-à-dire les vues de notre modèle MVC.

`/test` : Il contient les tests unitaires et tests fonctionnels.

`/translations` : contient les fichiers de traduction.

`/var` : Ce répertoire contient tout ce que Symfony va écrire durant son exécution : les logs, le cache.

`/vendor` : Ce répertoire contient toutes les bibliothèques externes à notre application. Dans ces bibliothèques externes telles que composer, doctrine, swiftmailer, Symfony, twig ...

`.env` : c'est ici que nous avons configuré les informations de connexion de notre base de données ainsi que nos identifiants de connexion à notre compte google pour les envois de mail.

`.env.test` : fichier pour l'environnement de test.

`.gitignore` : contient les fichiers et dossiers dont nous ne voulons pas envoyer vers notre dépôt distant lors d'un git push.

2.1.2.5. Coût du projet

Pour le coût d'un projet de plateforme avec Symfony, il faut prévoir :

- ✓ L'analyse des besoins (étude des besoins, modélisation) => 250.000F
- ✓ Le Développement => 800.000F
- ✓ L'Hébergement mutualisé avec accès SSH => minimum 48000F/an.
- ✓ La Maintenance trimestrielle => 25000F
- ✓ La formation des utilisateur (à négocier).

CHAPITRE 2 : Présentation de la plateforme

Dans ce chapitre, nous allons vous présenter plusieurs écrans de notre plateforme.

2.2.1. Partie sécurité : Page login de notre plateforme

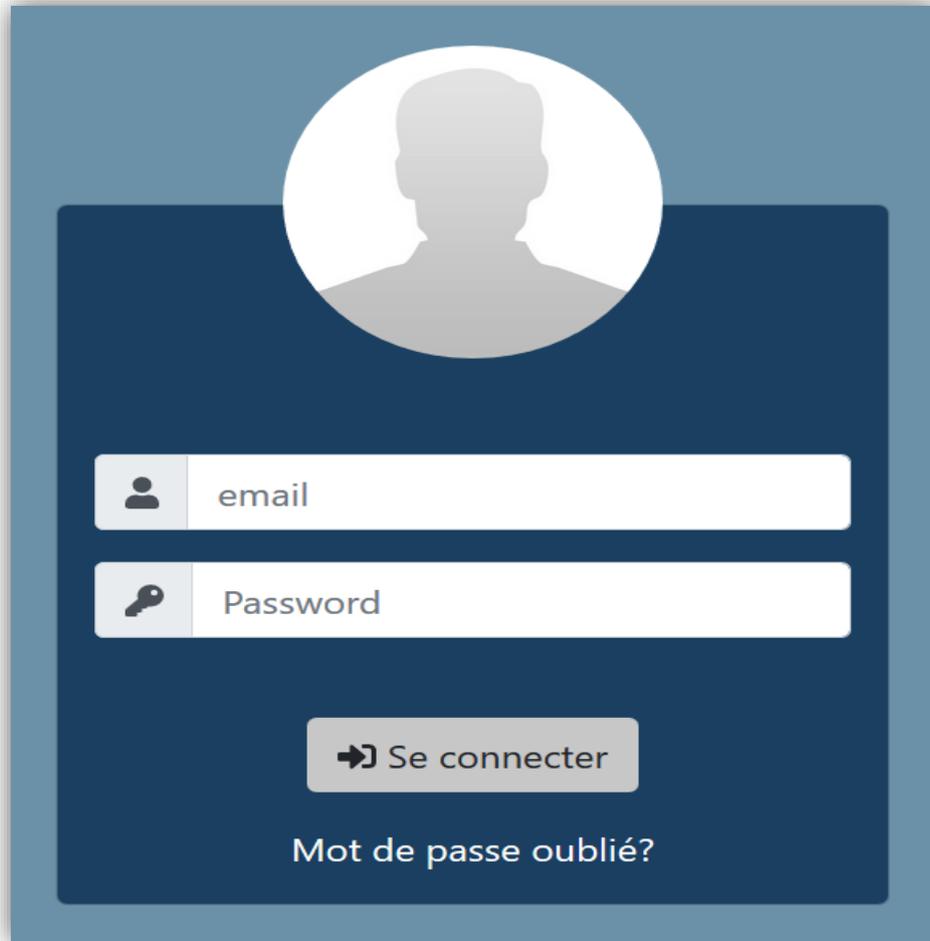


Figure 13. Page de connexion

Notre plateforme est accessible via le chemin /dashboard, mais une redirection vers le chemin /login est faite pour garantir la sécurité, de ce fait pour accéder à notre plateforme l'utilisateur devra s'authentifier avec son email et son mot de passe fournis par l'administrateur, ensuite notre système vérifie si le compte de l'utilisateur est actif au cas échéant il pourra accéder à notre plateforme. Donc en résumé l'utilisateur peut accéder au Dashboard si et seulement s'il fournit un email et un mot de passe valides et que son compte est actif.

2.2.2. Page Oublie de mot de passe

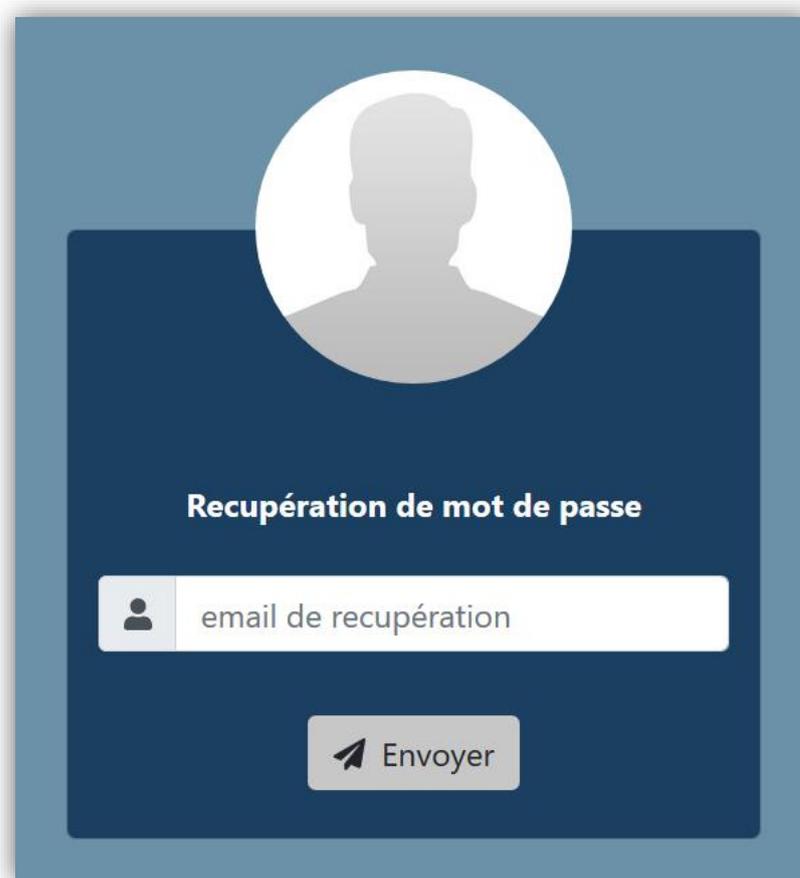


Figure 14. Page récupération mot de passe

Au cas où un utilisateur oublie son mot de passe il pourra le récupérer lorsqu'il clique sur le lien mot de passe oublié, son adresse mail lui sera demandé pour l'envoyer un mail avec un lien où il pourra le réinitialiser. Ce système de mailing est mis en place grâce à la librairie Swift Mailer de Symfony.

2.2.3. Récupération de mot de passe

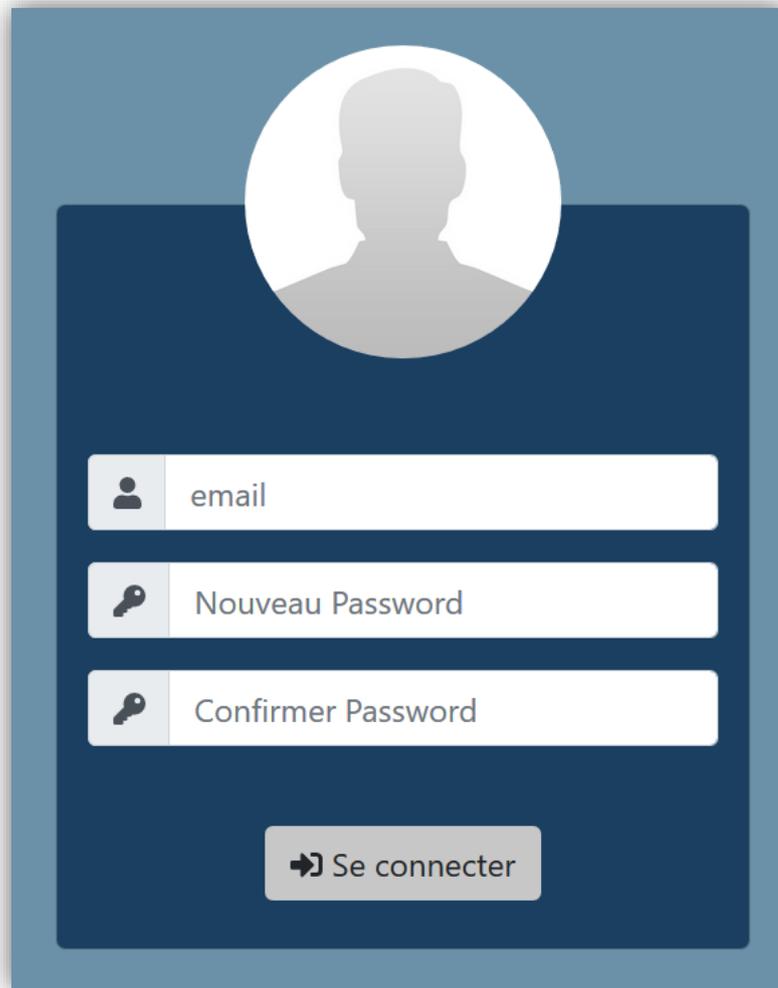


Figure 15. Page changement mot de passe

Comment se passe la récupération de mot de passe une fois qu'on accède au lien de récupération ?

Deux vérifications sont faites à ce niveau :

- ✓ Tout d'abord lorsque l'utilisateur saisie son email, on vérifie au niveau de la base de données si l'email entré existe bel et bien.
- ✓ Ensuite un nouveau mot de passe et une confirmation lui sont demandé.
- ✓ Si les deux mots de passe correspondent et que l'email est correct, il est redirigé vers la page de connexion sinon le système lui dit que l'email n'existe pas ou que les deux mots de passe ne correspondent pas.

2.2.4. Page d'accueil



Figure 16. Page d'accueil

Voici notre page d'accueil après une authentification et autorisation réussies.

Ici nous avons la liste des 3 trois dernières consultations, le nombre de consultations par service, le nombre d'hospitalisations mensuelles, le nombre total de chambres, de patients que compte la clinique, le nombre de comptes actifs, ici que l'ensemble des modules de notre plateforme.

2.2.6. Processus Patient

- Si prise de rendez-vous

Le patient appelle à la clinique en donnant son nom et son adresse email, la date et l'heure qu'il souhaite venir.

Arrivé à la clinique, le patient est orienté à l'accueil, il demande à faire une consultation.

Le secrétaire vérifie la liste des rendez-vous, après vérification enregistre les informations du patient.

Le médecin remplit le formulaire de consultation puis crée un dossier pour le patient.

Ensuite, le patient retourne chez le secrétaire pour payer sa facture.

- SINON

Le patient arrive à l'accueil, soit pour une consultation, soit pour une hospitalisation.

Pour une consultation, le patient s'inscrit sur la liste des consultations et est mis en attente.

Ensuite il s'en suit le même scénario pour une consultation.

Pour une hospitalisation, le secrétaire enregistre les informations du patient,

Ensuite le médecin remplit la fiche d'hospitalisation en ajoutant les informations du demandeur puis crée un dossier patient.

Après cela le médecin procède à une consultation du patient

La facturation se fait une fois que le patient reçoit une autorisation de sortie.

2.2.6. Page Liste Consultations / Détails d'une consultation

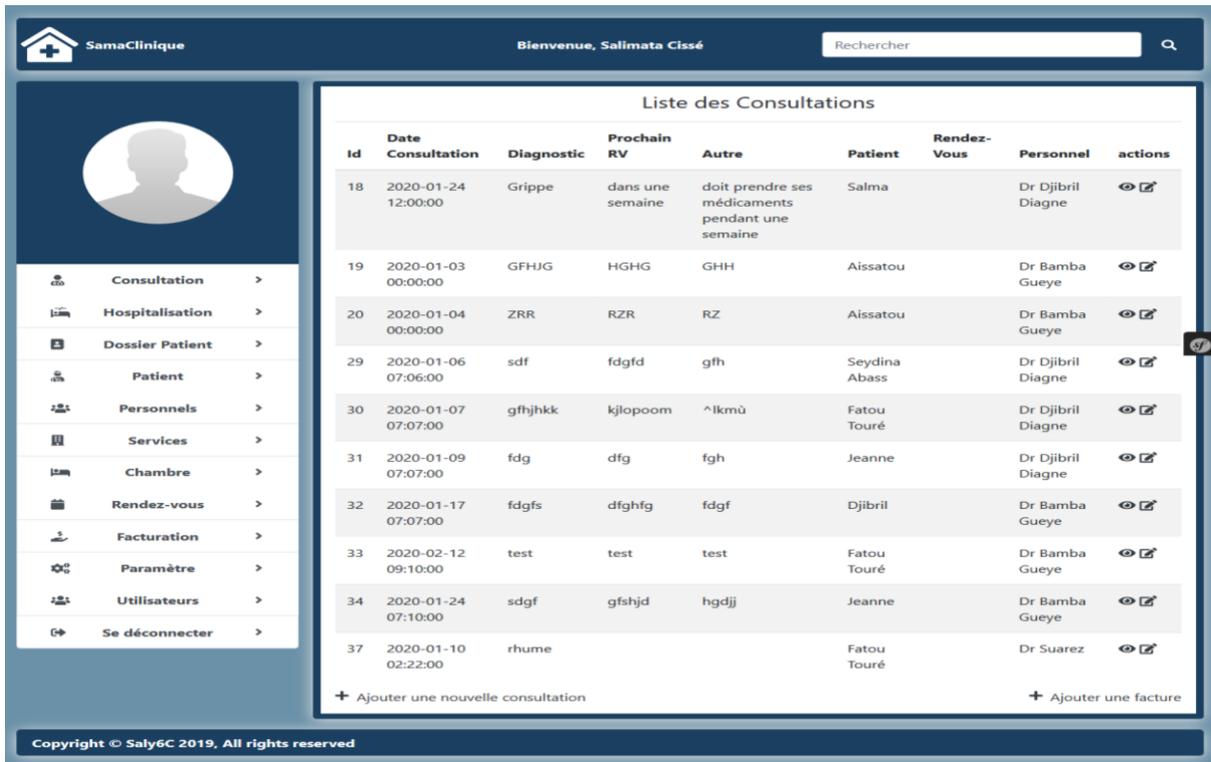


Figure 17. Page Liste des consultations

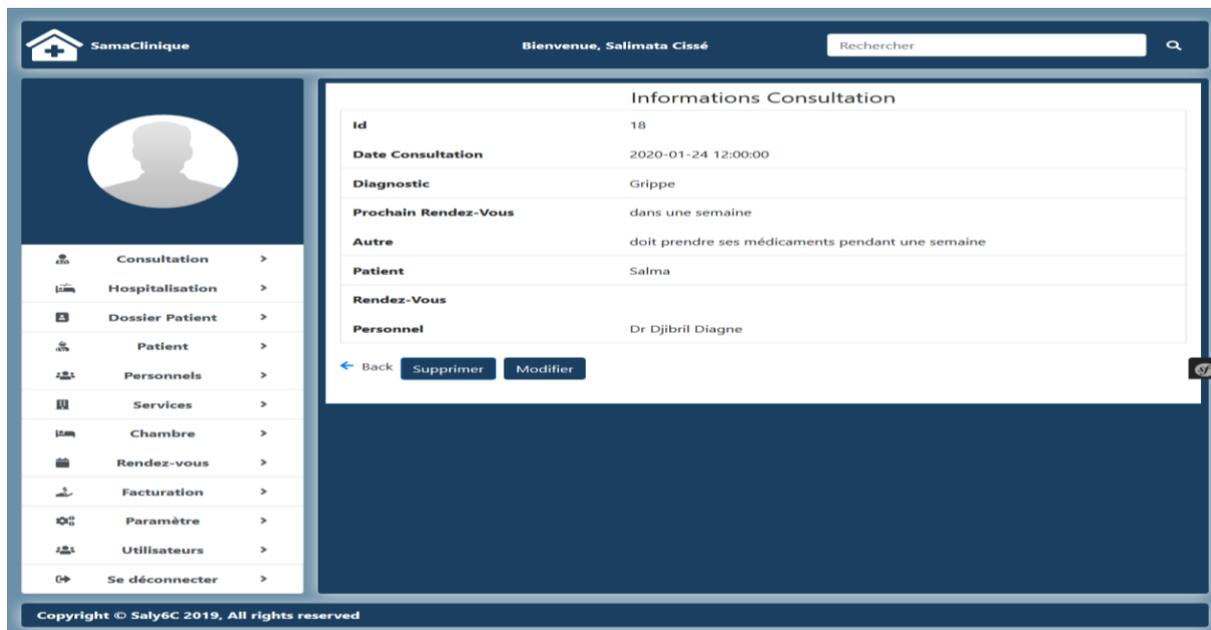


Figure 18. Page détail d'une consultation

Ici nous avons les informations sur une consultation x, à partir de là on peut modifier ou supprimer une consultation.

2.2.7. Page Liste/Modification hospitalisation

Id	Date Hospitalisation	Date Sortie	Chambre	Nom Patient	Piece Patient	Numéro piece Patient	Nom Demandeur	Piece Demandeur
3	2020-01-04 04:56:00	2020-01-07	Sophia	Jeanne	Passport	1234567890	safia	Passport
4	2020-02-08 05:57:00	2020-03-12	Mariama	Fatou Touré	CNI	2770199709898	Philippe Sagna	CNI
5	2020-01-10 06:32:00	2020-01-11	Dieynaba	Seydina Abass	Passport	123432344323	Joseph	Passport

Figure 19. Page Liste Hospitalisations

Modifier une Hospitalisation

Date hospitalisation: 2020-01-04 04:56
Date sortie: 2020-01-07 12:00
Chambre: Sophia
Patient: Jeanne
Nom demandeur: safia
Piece Malade: CNI, Passport
Numero piece malade: 1234567890
Numero piece demandeur: 234567890

Buttons: Enregistrer, Supprimer, Back

Figure 20. Page Modifier une Hospitalisation

La date de sortie est à modifier ultérieurement au moment de la facturation, nous pouvons aussi modifier le type de pièce du patient et/ou du demandeur, la chambre...

2.2.8. Page Dossier Patient

The screenshot shows the 'Page Dossier Patient' interface. At the top, there is a header with the logo 'SamaClinique', the user name 'Bienvenue, Salimata Cissé', and a search bar labeled 'Rechercher'. On the left, a sidebar menu contains various navigation options: Consultation, Hospitalisation, Dossier Patient (highlighted), Patient, Personnels, Services, Chambre, Rendez-vous, Facturation, Paramètre, and Se déconnecter. The main content area is titled 'Liste des dossiers Patients' and contains a table with the following data:

Id	Matricule	Nom Patient	Nom Maladie	Antécédant	Date Consultation	Date hospitalisation	Médecin	actions
1	1	Salma	hypoglycémie	néant	2020-01-24 12:00:00	2020-01-16 06:00:00	Dr Djibril Diagne	
2	6	Seydina Abass	palu	grippe	2020-01-06 07:06:00	2020-01-10 06:32:00	Dr Djibril Diagne	
3	7	niogou	Rhume	neant	2020-01-21 10:49:00		Dr Suarez	

Below the table, there is a button labeled '+ Ajouter un nouveau dossier'. At the bottom of the interface, a footer contains the text 'Copyright © Saly6C 2019, All rights reserved'.

Figure 21. Page Dossier Patients

Comme vous le voyez, dans cette interface nous avons la gestion des dossiers patients, nous avons les informations telles que le matricule, le nom, la date d'hospitalisation, la date de consultation du patient ainsi que le médecin traitant... Ce qui permet d'assurer une traçabilité de tous les traitements effectués.

2.2.9. Page Patient

The screenshot shows the 'Page Patient' interface. The top navigation bar includes the SamaClinique logo, the user name 'Bienvenue, Salimata Cissé', and a search bar labeled 'Rechercher'. The left sidebar menu contains the following items: Consultation, Hospitalisation, Dossier Patient, Patient, Personnels, Services, Chambre, Rendez-vous, Facturation, Paramètre, Utilisateurs, and Se déconnecter. The main content area displays a 'Liste des Patients' table with the following data:

Id	Nom	Adresse	Téléphone	Age	Actions
1	Salma	Rufisque	774521524	32	
2	Aissatou	Dakar	772495856	43	
3	Djibril	Dakar	774546396	25	
4	Jeanne	Medina, rue 11	775485256	12	
5	Fatou Touré	Pambal	774512586	35	
6	Seydina Abass	Liberté 4	767876565	12	

At the bottom of the table, there are buttons for '+ Ajouter un nouveau patient' and 'Créer un dossier patient'. The footer of the page contains the text 'Copyright © Saly6C 2019, All rights reserved'.

Figure 22. Page Patient

Ici nous avons la liste de tous les patients de la clinique avec toutes leurs informations personnelles. C'est le premier formulaire à remplir lorsqu'un patient arrive à la clinique. On peut y modifier des informations d'un patient, afficher des informations d'un patient, supprimer un patient (si l'utilisateur connecté est un administrateur), ajouter un nouveau patient. On peut aussi créer un dossier à partir de ce formulaire si on le souhaite.

2.2.10. Page Création facture

The screenshot shows the 'Créer une nouvelle facturation' (Create new invoice) page in the SamaClinique application. The interface is dark-themed with a blue header and sidebar. The header includes the SamaClinique logo, the user's name 'Bienvenue, Salimata Cissé', and a search bar. The sidebar on the left contains a navigation menu with icons and labels for various functions: Consultation, Hospitalisation, Dossier Patient, Patient, Personnels, Services, Chambre, Rendez-vous, Facturation, Paramètre, Utilisateurs, and Se déconnecter. The main content area is a form titled 'Créer une nouvelle facturation'. It contains several input fields: 'Montant facture' (a text input with a currency symbol), a checkbox for 'Prise en charge', 'Remise' (a text input with a currency symbol), 'Montant paye' (a text input with a currency symbol), 'Nom assurance' (a text input), 'Hospitalisation' (a dropdown menu), and 'Consultation' (a dropdown menu). At the bottom of the form are two buttons: 'Enregistrer' (dark blue) and 'Back' (light blue with a left arrow). The footer of the page contains the copyright notice: 'Copyright © Saly6C 2019, All rights reserved'.

Figure 23. Page Création Facture

Ici nous avons la gestion de la facturation liée aux consultations et aux hospitalisations. Pour créer une facture il faut saisir le montant de la facture, si le patient possède une prise en charge (IPM ou assurances) alors on coche prise en charge en indiquant le taux de remise, quant au montant payé c'est un champ automatique qui se remplit une fois que le montant de la facture et la remise sont saisis. Le nom de l'assurance ou de l'IPM est à saisir, pour ce qui est l'hospitalisation et la consultation nous avons des listes déroulantes de l'ensemble des hospitalisations et des consultations et des patients.

2.2.11. Page Modification/Suppression Facture

The screenshot displays the 'Modifier Facturation' interface. On the left, a sidebar contains a user profile and a menu with items: Consultation, Hospitalisation, Dossier Patient, Patient, Personnels, Services, Chambre, Rendez-vous, Facturation, Paramètre, Utilisateurs, and Se déconnecter. The main content area is titled 'Modifier Facturation' and contains the following fields and controls:

- Montant facture: Input field with value 10000.
- Prise en charge: Checked checkbox.
- Remise: Input field with value 10.
- Montant paye: Input field with value 9000.
- Nom assurance: Input field with value ascoma.
- Hospitalisation: Dropdown menu.
- Consultation: Dropdown menu with value Salma.
- Buttons: 'Modifier', 'Supprimer', and 'Back'.

Copyright © Saly6C 2019, All rights reserved

Figure 24. Page Modification/Suppression facture

La modification est très simple et une règle de validation est mise en place pour éviter de mettre du texte à la place d'un entier par exemple pour les champs montant facture et montant payé. Une fois la modification terminée, l'action du bouton modifier (si tout est valide) va permettre de mettre à jour la base de données. Quant à l'action de supprimer, elle va supprimer la facture en cours dans la base de données. Et le lien back permet de retourner à la liste des factures.

2.2.12. Page Personnel

The screenshot displays the 'Page Personnel' interface. At the top, the header includes the 'SamaClinique' logo, the user name 'Bienvenue, Salimata Cissé', and a search bar labeled 'Rechercher'. The left sidebar contains a navigation menu with items: Consultation, Hospitalisation, Dossier Patient, Patient, Personnels, Services, Chambre, Rendez-vous, Facturation, Paramètre, Utilisateurs, and Se déconnecter. The main content area features a 'Liste des Personnels' table with the following data:

Id	Fonction	Adresse	Telephone	Nom	Service	actions
1	Pédiatre	Mermoz	765825485	Dr Bamba Gueye	Pédiatrie	
2	Dermatologue	Thies	784526598	Dr Djibril Diagne	Dermatologie	

Below the table, there is a '+ Ajouter un nouveau personnel' button. The footer of the page contains the text 'Copyright © Saly6C 2019, All rights reserved'.

Figure 25. Page Personnel

Dans cette page, nous avons la gestion du personnel de la clinique. On peut y modifier afficher ou ajouter un personnel.

2.2.13. : Page Rendez-vous

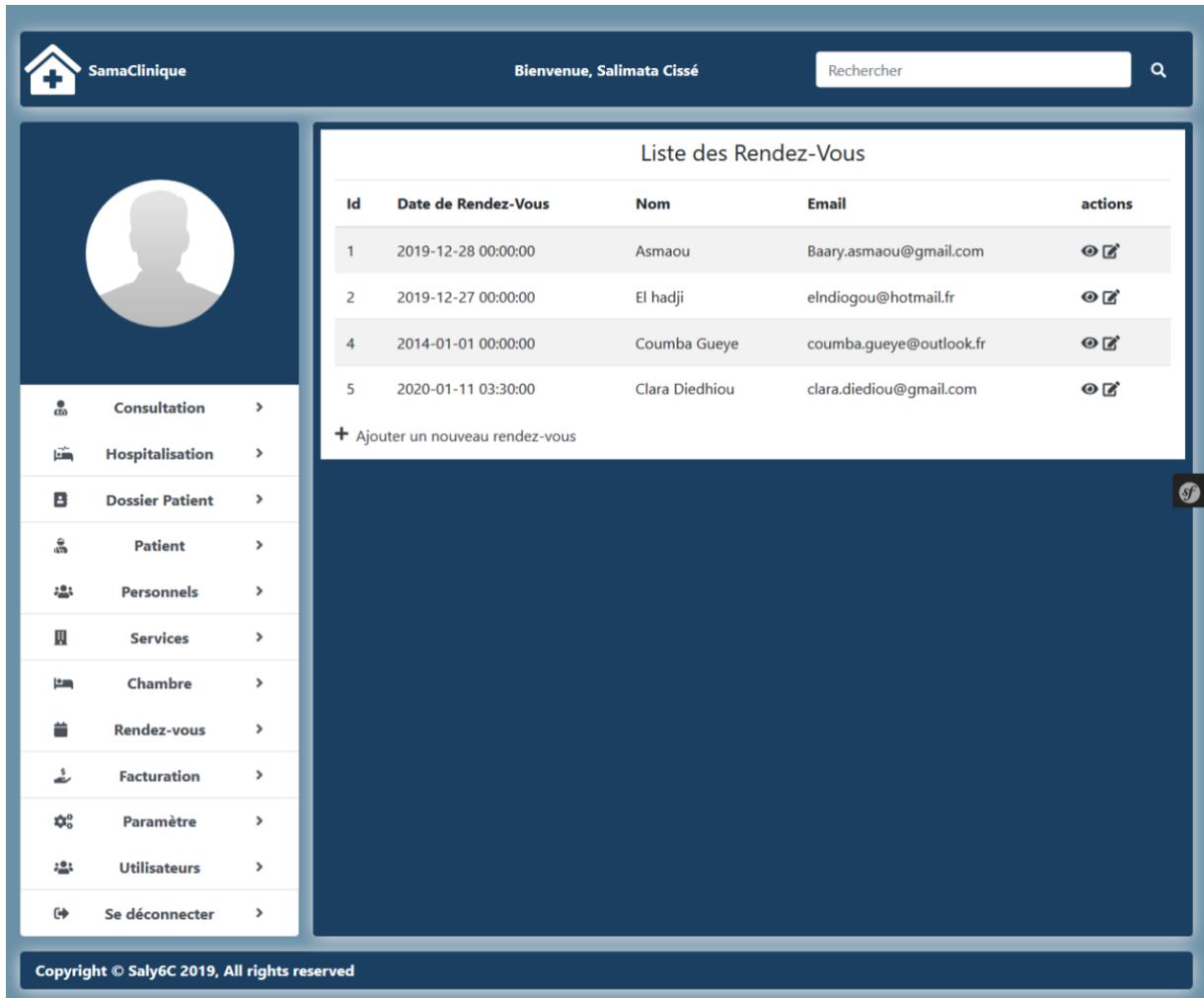


Figure 26. Page Rendez-vous

Cette interface, est la page de la gestion des rendez-vous. Pour prendre un rendez-vous le demandeur passe un appel téléphonique au niveau de la clinique , en indiquant son nom, la date à laquelle il souhaite venir ; ainsi que son adresse email qui est utilisée par la clinique pour l'envoi de mail de rappel. On peut y modifier, rechercher, afficher ou ajouter un rendez-vous.

2.2.14. : Page Création/Modification utilisateur

The screenshot shows the 'Nouvel utilisateur' page. On the left is a navigation menu with items like 'Consultation', 'Hospitalisation', 'Dossier Patient', 'Patient', 'Personnels', 'Services', 'Chambre', 'Rendez-vous', 'Facturation', 'Paramètre', 'Utilisateurs', and 'Se déconnecter'. The main content area has a header 'Nouvel utilisateur' and a search bar. Below the header are input fields for 'Email' (edouard.sarr@gmail.com), 'Status' (inactif), 'Username' (Edouard Sarr), and 'Password' (test@123@123). There are checkboxes for 'Admin', 'Assistant', and 'Medecin' (checked). A blue 'Enregistrer' button and a 'Back' link are at the bottom of the form. The footer contains 'Copyright © Saly6C 2019, All rights reserved'.

Figure 27. Page Création Utilisateur

Accessible que par l'administrateur, Il peut y activer/désactiver un compte, choisir le type de profile, renseigner l'adresse email qui est l'identifiant de l'utilisateur.

The screenshot shows the 'Modifier utilisateur' page. The navigation menu is identical to the previous page. The main content area has a header 'Modifier utilisateur' and a search bar. Below the header are input fields for 'Email' (aissatou@gmail.com), 'Status' (actif), 'Username' (Aissatou), and 'Password' (\$argon2id\$v=19\$m=65536,t=4,p=1\$U2VhVmhlZXpDZ). There are checkboxes for 'Admin', 'Assistant' (checked), and 'Medecin'. A blue 'Modifier' button, a blue 'Supprimer' button, and a 'Back' link are at the bottom of the form. The footer contains 'Copyright © Saly6C 2019, All rights reserved'.

Figure 28. Page Modification/Suppression Utilisateur

Après création d'un utilisateur, vous voyez que son mot de passe est encodé, c'est une des mesures de sécurité à considérer. Un mot de passe ne doit pas être affiché en clair.

3. Conclusion et perspectives :

En définitive, l'informatisation dans tous les secteurs d'activités occupe une place primordiale, toute entreprise qui se respecte doit être dotée d'un système informatique centralisé qui permet une parfaite traçabilité et une bonne gestion de leurs domaines d'activités.

Cependant notre plateforme à ce stade, constitue un véritable atout dans la gestion d'une structure sanitaire généraliste. Non seulement y'aura un gain de temps énorme mais aussi un mode de travail organisé et une parfaite sécurité sur les données des patients. Toute personne qui accède à la plateforme sera authentifiée de ce fait seules les personnes habilitées pourront y accéder.

Pour finir, plusieurs perspectives sont en vue pour une meilleure version de notre plateforme telles que : l'envoi de message de rappel pour les patients ayant pris rendez par email et/ou SMS, un système de messagerie instantanée pour les personnels, une impression des factures pour les clients via la plateforme, une gestion de profil pour l'utilisateur et la liste est non exhaustive.

REFERENCES

1. <https://mrproof.blogspot.com/2012/10/uml-cabinet-medical-gestion-des-rendez.html>, Consulté le 12/12/2019
2. <https://www.exelib.net/ado-net/gestion-des-rendez-vous-dans-un-hopital-en-mode-connecte.html>, Consulté le 12/12/2019
3. <http://www.gestionbienetre.com/lelogiciel.html>, Consulté le 15/02/2019
4. <http://newideait-dz.com/index.php/services/service-inner>, Consulté le 10/11/2019
5. <https://symfony.com/doc/current/index.html#gsc.tab=0>, Consulté le 12/12/2019
6. <https://openclassrooms.com/fr/courses/3619856-developpez-votre-site-web-avec-le-framework-symfony/3621961-les-services-theorie-et-creation>, Consulté le 23/12/2019
7. <https://stackoverflow.com/>, Consulté le 12/12/2019
8. <https://getbootstrap.com/docs/4.4/getting-started/introduction/>, Consulté le 11/08/2019
9. <https://fontawesome.com/icons>, Consulté le 12/12/2019
10. <https://getcomposer.org/download/>, Consulté le 11/08/2019
11. <https://github.com/>, Consulté le 12/12/2019
12. <https://www.apachefriends.org/fr/download.html>, Consulté le 12/12/2019
13. <https://fr.wikipedia.org/wiki/Git>, Consulté le 11/08/2019
14. https://fr.wikipedia.org/wiki/Visual_Studio_Code, Consulté le 30/11/2019
15. <https://www.php.net/manual/fr/>, Consulté le 12/12/2019
16. <https://twig.symfony.com/doc/2.x/>, Consulté le 12/12/2019
17. <https://github.com/smalot/bootstrap-datetimepicker/tree/master/js>, Consulté le 10/06/2019
18. <https://www.jqueryscript.net/time-clock/Date-Time-Picker-Bootstrap-4.html>, Consulté le 12/12/2019
19. <https://fr.wikipedia.org/wiki/Symfony>, Consulté le 04/12/2019
20. <https://hackr.io/blog/python-vs-php>, Consulté le 04/12/2019
21. <https://www.christophe-meneses.fr/article/creer-des-graphiques-avec-google-charts-dans-un-projet-symfony>, Consulté le 04/12/2019
22. <https://www.csschopper.com/blog/php-vs-java-opt-best-one-web-development/>, Consulté le 04/12/2019

BIBLIOGRAPHIES

- a.** Armand, S. (2014). *Extending Symfony2 Web Application Framework*. Packt Publishing Ltd.
- b.** Du, S., & Yuan, J. (2012, October). Bilan de l'utilisation du Framework Symfony2 pour la refonte complète de 2 applications Web.
- c.** Hu, Q. J. (2014, June). The research on intelligent system of digital publishing based on the framework of Symfony2. 0. In *2014 IEEE 5th International Conference on Software Engineering and Service Science* (pp. 796-799). IEEE.
- d.** Peltier, M. Développement d'applications Web avec le framework PHP Symfony 2.
- e.** Rey, P. (2013). *Analyse et création d'une plateforme de crowdfunding* (Doctoral dissertation, Haute école de gestion de Genève).

ANNEXE 1 : Extrait Modèle hospitalisation

```
<?php
namespace App\Entity;
use Doctrine\ORM\Mapping as ORM;
/**
 * @ORM\Entity(repositoryClass="App\Repository\HospitalisationRepository")
 */
class Hospitalisation
{
    /**
     * @ORM\Id()
     * @ORM\GeneratedValue()
     * @ORM\Column(type="integer")
     */
    private $id;

    /**
     * @ORM\Column(type="datetime")
     */
    private $dateHospitalisation;

    /**
     * @ORM\Column(type="date")
     */
    private $dateSortie;

    /**
     * @ORM\ManyToOne(targetEntity="App\Entity\Chambre", inversedBy="hospitalisation")
     * @ORM\JoinColumn(nullable=false)
     */
    private $chambre;

    /**
     * @ORM\ManyToOne(targetEntity="App\Entity\Patient", inversedBy="hospitalisations")
     * @ORM\JoinColumn(nullable=false)
     */
    private $patient;

    /**
     * @ORM\OneToOne(targetEntity="App\Entity\Facturation", inversedBy="hospitalisation", cascade={"p
ersist", "remove"})
     * @ORM\JoinColumn(nullable=true)
     */
    private $facturation;

    /**
     * @ORM\Column(type="string", length=255)
     */
    private $nomDemandeur;

    /**
     * @ORM\Column(type="array")
     */
    private $pieceDuMalade = [];

    /**
     * @ORM\Column(type="array")
     */
    private $pieceDuDemandeur = [];

    /**
     * @ORM\Column(type="bigint")
     */
    private $numeroPieceMalade;

    /**
     * @ORM\Column(type="bigint")
     */
    private $numeroPieceDemandeur;

    public function getId(): ?int
    {
        return $this->id;
    }
}
```

Figure 29. Modèle hospitalisation (extrait)

ANNEXE 2 : Vue hospitalisation

```

<div class="container">
  <h4 class="text-center py-2">Liste des Hospitalisations</h4>

  <table class="table table-striped table-hover table-responsive">
    <thead>
      <tr>
        <th scope="col">Id</th>
        <th scope="col">Date Hospitalisation</th>
        <th scope="col">Date Sortie</th>
        <th scope="col">Chambre</th>
        <th scope="col">Nom Patient</th>
        <th scope="col">Piece Patient</th>
        <th scope="col">Numéro piece Patient</th>
        <th scope="col">Nom Demandeur</th>
        <th scope="col">Piece Demandeur</th>
        <th scope="col">Numéro piece Demandeur</th>
        <th scope="col">actions</th>
      </tr>
    </thead>
    <tbody id="hospitalisationList">
      {% for hospitalisation in hospitalisations %}
        <tr>
          <td>{{ hospitalisation.id }}</td>
          <td>
            {{ hospitalisation.dateHospitalisation ? hospitalisation.dateHospitalisation|date
('Y-m-d H:i:s') : '' }}
          </td>
          <td>{{ hospitalisation.dateSortie ? hospitalisation.dateSortie|date('Y-m-d') : '' }}
        </td>
          <td>{{ hospitalisation.chambre }}</td>
          <td>{{ hospitalisation.patient }}</td>
          <td>{{ hospitalisation.pieceDuMalade[0] }}</td>
          <td>{{ hospitalisation.numeroPieceMalade }}</td>
          <td>{{ hospitalisation.nomDemandeur }}</td>
          <td>{{ hospitalisation.pieceDuDemandeur[0] }}</td>
          <td>{{ hospitalisation.numeroPieceDemandeur }}</td>
          <td>
            <a href="{{ path('hospitalisation_show', {'id': hospitalisation.id}) }}">
              <i class="fas fa-eye text-dark"></i>
            </a>
            <a href="{{ path('hospitalisation_edit', {'id': hospitalisation.id}) }}">
              <i class="fas fa-edit text-dark"></i>
            </a>
          </td>
        </tr>
      {% else %}
        <tr>
          <td colspan="12">no records found</td>
        </tr>
      {% endfor %}
    </tbody>
  </table>

  <a href="{{ path('hospitalisation_new') }}" class="d-flex pb-2">
    <i class="fa fa-plus text-dark" aria-hidden="true"></i>
    <h6 class="m1-2 text-dark">Ajouter une nouvelle hospitalisation</h6>
  </a>
</div>

```

Figure 30. Vue hospitalisation

ANNEXE 3 : Controller hospitalisation

```
<?php

namespace App\Controller;

use App\Entity\Hospitalisation;
use App\Form\HospitalisationType;
use App\Repository\HospitalisationRepository;
use Symfony\Bundle\FrameworkBundle\Controller\AbstractController;
use Symfony\Component\HttpFoundation\Request;
use Symfony\Component\HttpFoundation\Response;
use Symfony\Component\Routing\Annotation\Route;

/**
 * @Route("/dashboard/hospitalisation")
 */
class HospitalisationController extends AbstractController
{
    /**
     * @Route("/", name="hospitalisation_index", methods={"GET"})
     */
    public function index(HospitalisationRepository $hospitalisationRepository): Response
    {
        return $this->render('hospitalisation/index.html.twig', [
            'hospitalisations' => $hospitalisationRepository->findAll(),
        ]);
    }

    /**
     * @Route("/new", name="hospitalisation_new", methods={"GET","POST"})
     */
    public function new(Request $request): Response
    {
        $hospitalisation = new Hospitalisation();
        $form = $this->createForm(HospitalisationType::class, $hospitalisation);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $entityManager = $this->getDoctrine()->getManager();
            $entityManager->persist($hospitalisation);
            $entityManager->flush();
            $this->addFlash('success', 'Hospitalisation enregistrée avec succès!');

            return $this->redirectToRoute('hospitalisation_index');
        }

        return $this->render('hospitalisation/new.html.twig', [
            'hospitalisation' => $hospitalisation,
            'form' => $form->createView(),
        ]);
    }

    /**
     * @Route("/{id}", name="hospitalisation_show", methods={"GET"})
     */
    public function show(Hospitalisation $hospitalisation): Response
    {
        return $this->render('hospitalisation/show.html.twig', [
            'hospitalisation' => $hospitalisation,
        ]);
    }

    /**
     * @Route("/{id}/edit", name="hospitalisation_edit", methods={"GET","POST"})
     */
    public function edit(Request $request, Hospitalisation $hospitalisation): Response
    {
        $form = $this->createForm(HospitalisationType::class, $hospitalisation);
        $form->handleRequest($request);

        if ($form->isSubmitted() && $form->isValid()) {
            $this->getDoctrine()->getManager()->flush();
            $this->addFlash('success', 'Hospitalisation modifiée avec succès!');
            return $this->redirectToRoute('hospitalisation_index');
        }

        return $this->render('hospitalisation/edit.html.twig', [
            'hospitalisation' => $hospitalisation,
            'form' => $form->createView(),
        ]);
    }

    /**
     * @Route("/{id}", name="hospitalisation_delete", methods={"DELETE"})
     */
    public function delete(Request $request, Hospitalisation $hospitalisation): Response
    {
        if ($this->isCsrfTokenValid('delete', $hospitalisation->getId(), $request->request->get('_token'))) {
            $entityManager = $this->getDoctrine()->getManager();
            $entityManager->remove($hospitalisation);
            $entityManager->flush();
            $this->addFlash('success', 'Hospitalisation supprimée avec succès!');
        }

        return $this->redirectToRoute('hospitalisation_index');
    }
}
```

Figure 31. Controller hospitalisation