

**MINISTERE DE LA PLANIFICATION DU
DEVELOPPEMENT ET DE LA
COOPERATION**



Institut Africain d'Informatique

Représentation du TOGO

Tel : 0022822204700

Email: iaitogo@iai-togo.tg

Site web: www.iai-togo.tg

07 BP 12456 Lomé 07, TOGO.



Tel : 0022899544141

Email : insight@insight-tg.com

Lomé-TOGO

RAPPORT DE STAGE PRATIQUE EN ENTREPRISE

Option : GENIE LOGICIEL

**GESTION DES ANALYSES ET DE LA
FACTURATION
DES ACTES MEDICAUX**

Période : Du 06 Juin 2022 au 05 août 2022

Rédigé et soutenu par :

ADJIMON Arnaud Fouayéyêmi

Etudiant en deuxième année Tronc Commun

Année Scolaire : 2021-2022

SUPERVISEUR

M.TETTEH Kodjo Amen

Chargé de cours à IAI-TOGO

MAITRE DE STAGE

M. LASSEY Théophile

Responsable Informatique

REMERCIEMENTS

Je remercie Dieu Tout Puissant pour la vie et la santé qu'il m'accorde, ainsi qu'à mes proches.

J'adresse mes remerciements à mes parents pour leurs conseils, soutiens et investissements et d'avoir choisi IAI-TOGO pour ma formation.

J'adresse aussi mes sincères remerciements aux personnes suivantes :

- ❖ M. AGBETI Kodjo, Représentant résidant de l'IAI-TOGO, pour les efforts qu'il ne cesse de déployer pour l'Institut,
- ❖ M. AMEYIKPO Kossi, Directeur des Affaires Académiques et de la Scolarité de l'IAI-TOGO, pour l'encadrement et les conseils fournis,
- ❖ M. TETTEH Kodjo Amen, Enseignant à l'IAI-TOGO et superviseur de notre travail pour ses conseils, apports, critiques et encadrement
- ❖ M. ADIKA Kodjo Mawuenam, Directeur exécutif d'INSIGHT,
- ❖ M. LASSEY Théophile, Responsable Informatique d'INSIGHT,
- ❖ Tous ceux qui de près ou de loin ont œuvré au bon déroulement de ce stage.

SOMMAIRE

REMERCIEMENTS	i
SOMMAIRE	ii
GLOSSAIRE	iii
LISTE DES FIGURES	iv
LISTE DES TABLEAUX.....	v
INTRODUCTION	1
1. Cahier des charges	3
1.1 Présentations	3
1.2 Formation à l'IAI-TOGO	3
1.3 Plan de localisation	3
1.4 Présentation du sujet.....	4
1.5 Problématique du sujet.....	4
1.6 Intérêt du sujet.....	5
2 Préprogrammation	8
2.1 Etude de l'existant	8
2.2 Critique de l'existant	9
2.3 Planning prévisionnel de réalisation	9
2.4 Etude détaillée de la solution	10
3 Réalisation et mise en œuvre	27
3.1 Matériels et logiciels utilisés	27
3.2 Sécurité de l'application	33
3.3 Evaluation financière de la solution	34
3.4 Présentation de l'application	38
CONCLUSION.....	51
BIBLIOGRAPHIE INDICATIVE	52
WEBOGRAPHIE INDICATIVE	53
TABLE DES MATIERES.....	54

GLOSSAIRE

Framework : ensemble d'outils et de composants logiciels organisés conformément à un plan d'architecture et des patterns.

ORM : acronyme de « Object Relational Mapping », technique de programmation informatique qui permet de simplifier l'accès à une base de données en proposant à l'informaticien des « objets » plutôt que d'accéder directement à des données relationnelles.

LISTE DES FIGURES

Figure 1 : Plan de localisation de l'IAI-TOGO	4
Figure 2 : Diagramme contexte statique	13
Figure 3 : Diagramme général de cas d'utilisation	16
Figure 4 : Diagramme de cas d'utilisation du directeur de la clinique	17
Figure 5 : Diagramme d'activité « S'authentifier »	20
Figure 6 : Diagramme d'activité « Enregistrer une facture »	21
Figure 7 : Diagramme de séquence « S'authentifier »	22
Figure 8 : Diagramme de séquence « Enregistrer une facture »	23
Figure 9 : Diagramme de classe finale du système	25
Figure 10 : Fichier settings.py de connexion à la base de données	40
Figure 11 : Exemple de modèle du fichier models.py	41
Figure 12 : Fichier admin.py pour la configuration de l'interface admin	42
Figure 13 : Interface administrateur de gestion des opérations CRUD des modèles Django	43
Figure 14 : Plan de navigation de l'application	44
Figure 15 : Page de connexion	45
Figure 16 : Formulaire de validation d'un patient	45
Figure 17 : Formulaire de validation d'un résultat d'analyse	46
Figure 18 : Formulaire d'ajout d'une ligne facture	46
Figure 19 : Page présentant la liste des patients	47
Figure 20 : Fiche numérique d'un résultat d'analyses effectuées par un patient	47
Figure 21 : Fiche imprimable du résultat d'analyse	48
Figure 22 : Facture numérique recensant les actes médicaux effectués par un patient	48
Figure 23 : Fiche imprimable de la facturation des actes médicaux d'un patient	49
Figure 24 : Page des statistiques du caissier	49
Figure 25 : Page des statistiques de l'administrateur	50

LISTE DES TABLEAUX

Tableau 1 : Planning prévisionnel de réalisation du futur système	9
Tableau 2 : Coût matériel de la solution	35
Tableau 3 : Coût du développement de la solution.....	36
Tableau 4 : Coût total de la solution	37

INTRODUCTION

De nos jours, l'usage de l'informatique s'est généralisé dans tous les secteurs. L'informatique devient alors le support permettant aux entreprises d'évoluer, de se développer librement et de se tourner vers la modernité.

La gestion d'une clinique est un domaine qui englobe un grand nombre d'activités mises en œuvre afin d'assurer la qualité et la sécurité des soins prodigués aux patients.

Dans le cadre de notre formation, l'Institut Africain d'Informatique Représentation du TOGO (IAI-TOGO) intègre dans le cursus de formation des Analystes-Programmeurs, à la fin de la deuxième année d'étude, un stage de formation pratique d'une durée de huit semaines. Etant étudiant en deuxième année, ce stage nous a permis de mettre en pratique les connaissances acquises durant les deux années de formation au sein de l'Institut.

Dans ce document, nous présenterons le thème de notre stage ayant pour objet la gestion des analyses et de la facturation médicale. Notre objectif consiste à développer une application spécifique destinée au service de facturation de la clinique.

Notre rapport de stage est subdivisé en trois grandes parties : le cahier des charges, le rapport de préprogrammation et le rapport de programmation.

PARTIE 1 : CAHIER DES CHARGES

1. Cahier des charges

Le cahier de charges est un document qui contient toutes les exigences et besoins du client et qui fournit les informations nécessaires pour réaliser le logiciel. Il permet également aux deux parties (client et prestataire) de mentionner les objectifs attendus, les moyens techniques et humains nécessaires à la réalisation du système, les délais de réalisation prévus et les évolutions futures envisagées.

Notre cahier de charges porte sur les présentations (de l'IAI-TOGO et du centre d'accueil) et notre thème de stage.

1.1 Présentations

L'institut Africain d'Informatique Représentation du TOGO est une école de formation en Informatique. L'IAI-TOGO fait partie du réseau d'écoles IAI créé le 29 Janvier 1971 à Fort Lamy (actuel N'Djamena) en république du TCHAD. Ce réseau est composé de 11 pays membres dont le BÉNIN, le BURKINA-FASO, le CAMEROUN, le CONGO, la COTE d'IVOIRE, le GABON le NIGER, la RÉPUBLIQUE CENTRAFRICAINE le SÉNÉGAL, le TCHAD et le TOGO.

1.2 Formation à l'IAI-TOGO

L'IAI-TOGO dispose de trois filières à savoir : le Génie Logiciel et Système d'Information (GLSI), l'Administration des Systèmes et Réseaux (ASR) et le Multimédia et Technologie du Web et Infographie (MTWI).

Les étudiants de cet Institut ont la possibilité de suivre l'une des trois formations proposées pour une durée de trois ans afin d'obtenir le diplôme d'Ingénieur des Travaux Informatiques (ITI) ou la licence professionnelle.

1.3 Plan de localisation

L'IAI-TOGO est situé à Lomé, dans les locaux du Centre National d'Études et de Traitements Informatique (CENETI), lui-même situé dans le quartier administratif, à proximité de la Direction Générale de la Communauté Électrique du Bénin CEB et de l'immeuble de SUNU Assurances.

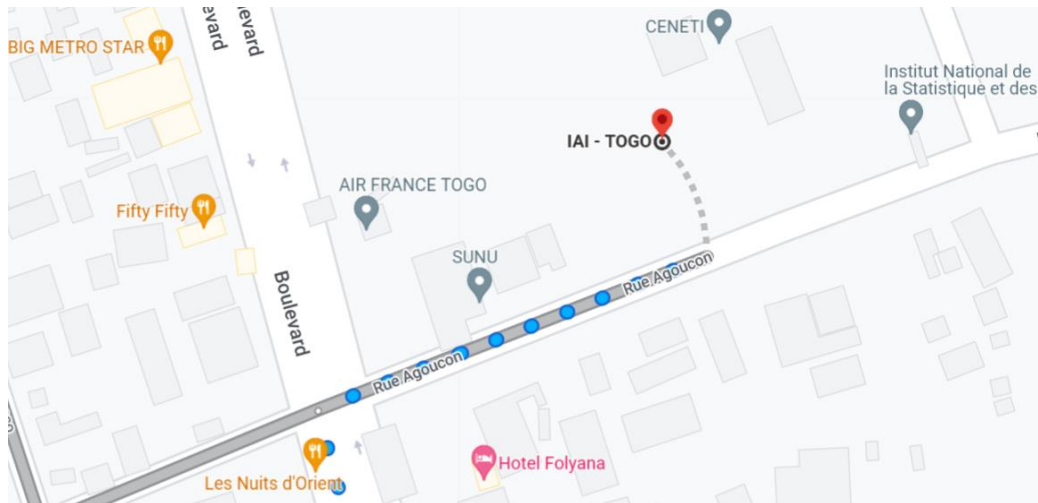


Figure 1 : Plan de localisation de l'IAI-TOGO

1.4 Présentation du sujet

La santé est primordiale pour tout organisme vivant et l'être humain, soucieux de son état, se confie aux services médicaux pour mieux se préserver ou remédier aux diverses menaces présentes dans la vie courante. Une bonne gestion des données par ces derniers s'avèrent capitale afin de sauvegarder le patrimoine médical et améliorer la qualité des services. C'est dans le but de faciliter cette gestion qu'il nous a été confié dans le cadre de notre stage, le thème suivant : « **Gestion des analyses et de la facturation des actes médicaux** ».

Ce thème consiste à développer une application web facilitant le processus de réalisation du bilan d'analyse et de facturation de la clinique.

1.5 Problématique du sujet

A l'état actuel la gestion des analyses et des factures au sein de la clinique « nom clinique » est un véritable problème. Au niveau de la facturation, le service comptable établit de façon manuelle les factures pour les services en fonction de chaque patient, du nombre d'actes, et de la quantité. De plus les calculs des frais supportés par chaque patient varient d'un modèle à un autre et aussi l'enregistrement des patients (au niveau de la consultation) se faisant dans un registre à la main, y est refaite au service de comptabilité pour tenir les comptes. Au

niveau des analyses, le laboratoire de la clinique se réfère au registre patients pour récupérer les informations de ce dernier et procède à la rédaction du bilan d'analyse sur un formulaire rempli à la main. Tout ceci implique :

- Une redondance des informations à l'étape d'enregistrement des patients ;
- Une perte de temps liée à la recherche manuelle fastidieuse des informations d'un patient et au traitement de plusieurs cas à la fois ;
- Des risques d'erreurs élevés lors du calcul des frais supportés par le patient et du remplissage des formulaires ;
- La perte des informations en cas de destruction du registre ou égarement des fiches de factures et de résultat des analyses;
- Une conservation indélicate du secret médical des patients.

Ayant remarqué des insuffisances dans leur gestion, l'administration de la clinique nous demande de lui proposer une solution efficace répondant à la gestion des analyses et de la facturation médicale. Il se dégage la problématique contenue dans les questions suivantes :

- Comment faciliter le processus d'élaboration du d'élaboration du résultat d'analyse ?
- Comment rendre simple et de manière sécurisée les factures pour de futures utilisations ?
- Comment faciliter le traitement des différents documents traités au sein de la clinique ?

1.6 Intérêt du sujet

1.6.1 Objectifs

1.6.1.1 Objectif général

Notre projet a pour objectif de produire une application permettant d'automatiser la gestion des analyses et de la facturation au sein de la clinique « nom clinique ».

1.6.1.2 Objectifs spécifiques

De façon spécifique, le projet doit permettre :

- D'enregistrer les informations des patients ;
- D'enregistrer les traitements effectués sur les patients ;
- D'enregistrer les résultats d'analyse d'un patient;

- D'éditer la facture du patient ;
- De consulter la liste des patients,
- De consulter les recettes de la clinique ;
- D'imprimer les résultats d'analyses destinés aux patients ;
- D'imprimer et sauvegarder les factures destinées aux patients ;
- D'effectuer des statistiques.

1.6.2 Résultats attendus

A la fin de notre stage, la clinique « nom clinique » disposera d'une application web sécurisée qui offrira les services suivantes :

- Les patients gérés efficacement et facilement ;
- Les factures de chaque patient imprimées;
- Les résultats d'analyses de chaque patient imprimées ;
- des statistiques fiables donnant des aperçus sur les recettes de la clinique.

PARTIE 2 : PRE-PROGRAMMATION

2 Préprogrammation

Au cours du développement d'un logiciel, la phase de préprogrammation reste primordiale. Pour mener à bien notre projet, une bonne observation des faits nous permettra d'atteindre un résultat satisfaisant. Nous présenterons dans cette partie, les solutions possibles qui répondent aux critiques. Ensuite nous proposerons et étudierons la solution sur laquelle notre travail s'est basé.

2.1 Etude de l'existant

La clinique « nom clinique » ne possède pas de systèmes informatiques qui répondent de façon efficace à leurs besoins. En effet, la plupart de leurs opérations se font manuellement sauf la facturation qui est répertoriée dans un fichier Excel pour l'utilisation future au niveau de la comptabilité. Ces fichiers restent sur le matériel ou le disque de stockage du comptable. De même, les fiches d'analyses sont créées en fichier Word et imprimées en cas de besoin pour établir un bulletin d'analyse.

Les opérations souvent effectuées manuellement sont :

➤ **L'enregistrement des patients qui se fait comme suit :**

Le réceptionniste remplit un cahier destiné à contenir les informations du patient. Il reporte ensuite sur une petite fiche, le numéro du patient prise en charge.

➤ **L'enregistrement des résultats d'analyses :**

Le technicien reporte sur un formulaire, les résultats trouvés lors de l'analyse d'un échantillon. Il transcrit ces données en fichier Word et fait une ensuite une copie faisant office d'un résultat d'analyse.

➤ **La gestion des factures :**

L'envoi des factures se fait par des inventaires, les calculs se font manuellement et le fichier inventorié se conçoit en Word puis est converti en fichier PDF.

2.2 Critique de l'existant

Comme constaté au niveau de la problématique, la sauvegarde et la cohérence des données sont vraiment primordiales pour une bonne connaissance des revenus.

D'après notre étude de l'existant, nous avons relevé les insuffisances suivantes :

- Les fichiers Excel pour la gestion de la comptabilité sont stockés uniquement sur la machine du comptable ;
- Les factures des actes médicaux sont sauvegardées au format papier et sont parfois difficile à retrouver ;
- Il n'existe pas de bases de données comportant les informations des patients ;
- Les résultats d'analyses sont elles aussi sauvegardées au format papier.

2.3 Planning prévisionnel de réalisation

Tableau 1 : Planning prévisionnel de réalisation du futur système

TACHES	DESCRIPTION	PERIODE (2022)	DUREE
Intégration et choix du thème	-Intégration dans le centre d'accueil -Prise de connaissance du thème	22-23 juin	2 jours
Recueil des besoins	-Rédaction du cahier des charges	24-26 juin	3 jours
Analyse et conception	-Elaboration des diagrammes des cas d'utilisation et du diagramme de classes	27 juin-01 juillet	5 jours
PHASE DE CODAGE			
Itération N° 1	Mise en place de la base de données, des fonctions du CRUD et de l'authentification	02-06 juillet	5 jours
Itération N° 2	Développement du module de gestion des analyses	07-10 juillet	4 jours

TACHES	DESCRIPTION	PERIODE (2022)	DUREE
Itération N° 3	Développement du module de gestion de la facturation	11-24 juillet	14 jours
Tests et corrections des erreurs		25-27 juillet	3 jours
Rédaction du document		24 juin – 31 juillet	38 jours
Correction du document		01-03 août	3 jours

2.4 Etude détaillée de la solution

2.4.1 Le langage de modélisation : UML

Langage de Modélisation Unifié, UML (Unified Modeling Language) est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

Il est une synthèse de langages de modélisation objet antérieurs comme Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG). La dernière version de la spécification validée par l'OMG est UML 2.5.1 (2017) 16. Il existe en ce jour, 14 diagrammes répartis en 3 grandes classes. A savoir :

❖ Les diagrammes structurels ou statiques (07) :

- Diagramme de classes (class diagram) ;
- Diagramme d'objets (object diagram) ;
- Diagramme de composants (component diagram) ;
- Diagramme de déploiement (deployment diagram) ;
- Diagramme des paquets (package diagram) ;
- Diagramme de structure composite (composite structure diagram) ;
- Diagramme de profils (profile diagram).

❖ Les diagrammes comportementaux (03) :

- Diagramme de cas d'utilisation (use-case diagram) ;
- Diagramme d'états-transitions (state machine diagram) ;
- Diagramme d'activité (activity diagram).

❖ **Les diagrammes d'interaction ou dynamiques (04) :**

- Diagramme de séquence (sequence diagram) ;
- Diagramme de communication (communication diagram) ;
- Diagramme global d'interaction (interaction overview diagram) ;
- Diagramme de temps (timing diagram).

2.4.2 Processus de développement

Un processus de développement logiciel est le processus de division du travail de développement logiciel en étapes ou sous-processus plus petits, parallèles ou séquentiels pour améliorer la conception, la gestion des produits et la gestion de projet. Pour réaliser un système d'information, nous avons le choix entre différents modèles de développement parmi lesquels nous distinguons :

- Le modèle en cascade ;
- Le modèle en V ;
- Le modèle par prototypage ;
- Les modèles de processus unifié
 - UP
 - RUP
 - 2TUP

Dans le cadre de notre projet nous avons fait le choix d'utiliser le processus de développement 2TUP car ce dernier est itératif et définit des étapes à suivre lors du développement d'un logiciel. Il fait également une large place à la technologie et à la gestion du risque tout en permettant d'identifier les différents intervenants dans le projet.

2.4.3 Outil de modélisation

Pour faire la modélisation de notre service nous avons choisi comme outil **PowerDesigner** dans sa version 15.1.

PowerDesigner est un logiciel de conception créé par la société SAP, qui permet de modéliser les systèmes d'informations. Il fonctionne en tant qu'application native sous Windows ou en tant que plugin dans un environnement Eclipse.

2.4.4 Etude préliminaire

Nous détaillerons dans cette partie les différents acteurs qui vont interagir avec notre système et nous ferons une représentation à l'aide d'un diagramme de contexte statique.

2.4.4.1 Les acteurs

Un acteur représente un rôle d'un interagissant avec le système modélisé. L'utilisateur peut être une machine, un système externe, une organisation ou un utilisateur humain.

Dans la conception de notre système, nous avons recensé les acteurs suivants :

- L'Administrateur ;
- Le directeur de la clinique ;
- Le personnel médical composé du médecin, de l'infirmier et de l'aide-soignant ;
- Le technicien du laboratoire ;
- Le caissier ;
- Le comptable ;
- Le réceptionniste.

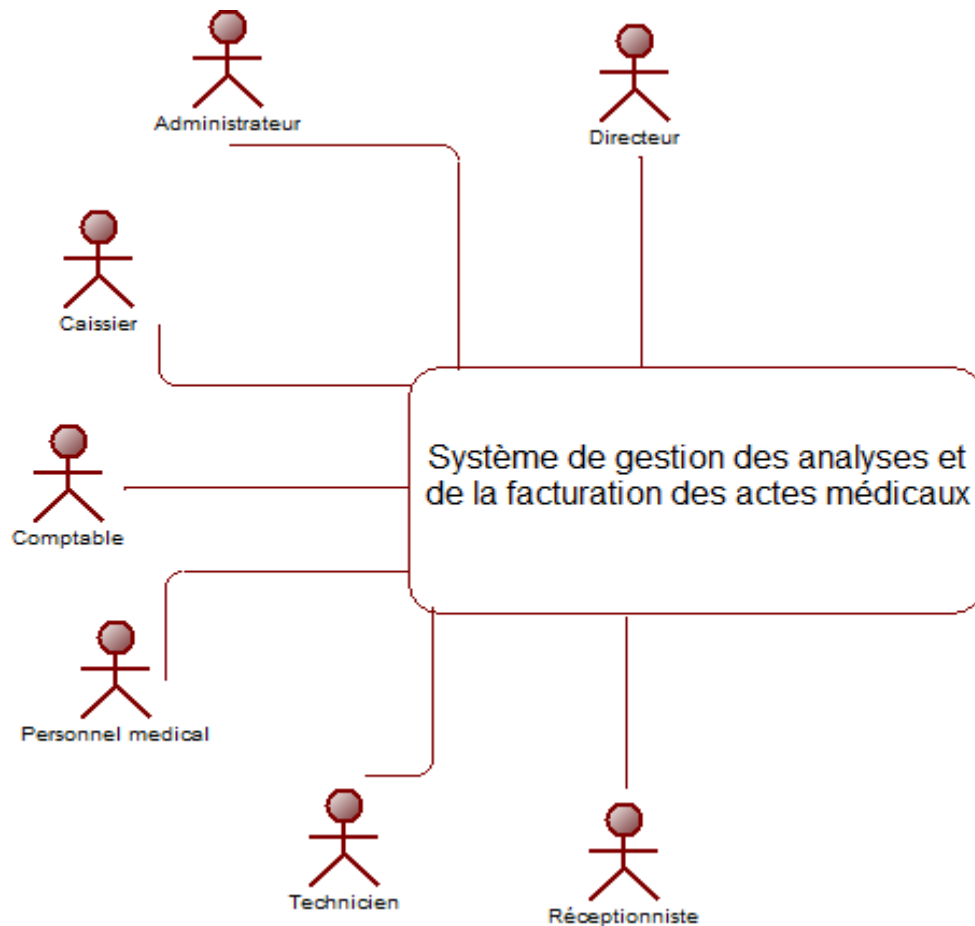


Figure 2 : Diagramme contexte statique

2.4.5 Capture des besoins fonctionnels

Dans cette partie, nous aurons à détailler les fonctions spécifiques au système à l'aide de quelques diagrammes notamment ceux de cas d'utilisation, d'activités et de séquences.

2.4.5.1 Diagrammes de cas d'utilisation

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour une représentation du comportement fonctionnel d'un système logiciel. Ils permettent d'illustrer ce que le système fait et comment les différents acteurs l'utilisent. Ils sont utiles pour des présentations auprès de la direction ou les acteurs d'un projet.

2.4.5.1.1 Les cas d'utilisation

Les cas d'utilisation correspondent à un ensemble d'actions réalisées par le système en interaction avec les utilisateurs qui renvoient des résultats observables et utiles pour ces utilisateurs. Nous verrons dans le tableau ci-dessous les cas d'utilisation recensés pour notre système.

Tableau 2 : Récapitulatif des cas d'utilisation

Cas d'utilisation		Acteurs
S'authentifier		Tous les utilisateurs
Gérer les patients	Ajouter des patients	Réceptionniste
	Désactiver des patients	
	Modifier des patients	
	Consulter la liste des patients	
Gérer les factures	Ajouter une facture	Caissier
	Ajouter des lignes factures	
	Rectifier une facture	
	Imprimer la facture	
Gérer les actes médicaux	Ajouter	Caissier
	Modifier	
Gérer les comptes utilisateurs	Ajouter un utilisateur	Administrateur
	Désactiver un utilisateur	
	Modifier un utilisateur	
	Modifier un rôle	
Gérer les droits	Attribuer un droit	Administrateur
	Modifier un droit	
Gérer les prélèvements Et analyses	Ajouter	Technicien
	Attribuer	

Cas d'utilisation		Acteurs
Etablir un résultat d'analyse	Ajouter	Technicien
	Consulter	
	Imprimer	
Consulter les dossiers patients		Personnel médical
Consulter les actes médicaux		Directeur de la clinique, Caissier
Consulter les états		Caissier, Comptable, Directeur de la clinique
Consulter les actes médicaux		Caissier, Directeur de la clinique

2.4.5.1.2 Représentation des diagrammes de cas d'utilisation

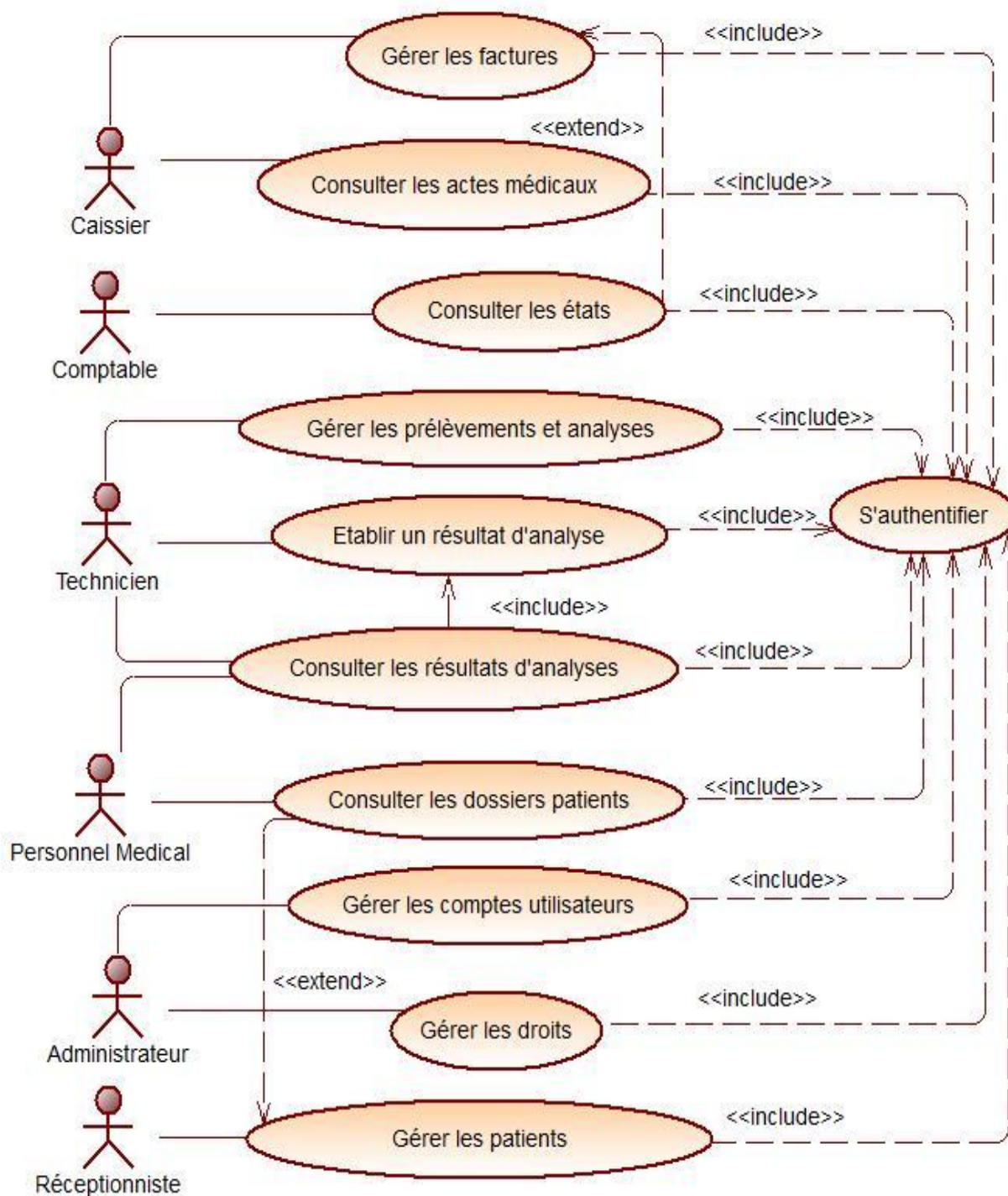


Figure 3 : Diagramme général de cas d'utilisation

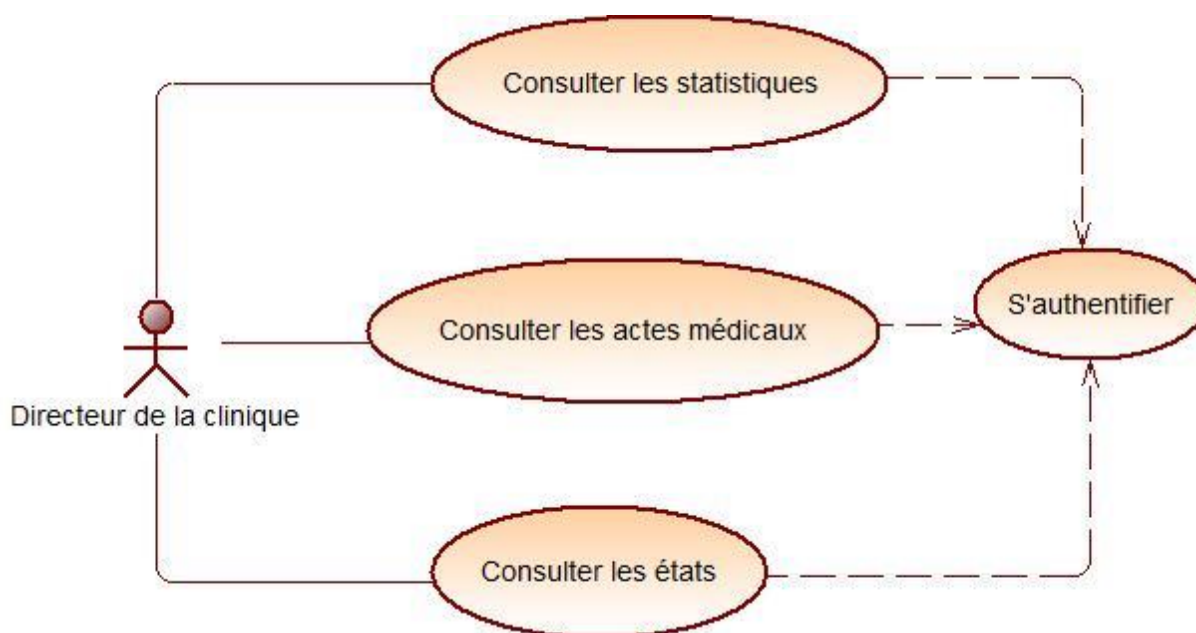


Figure 4 : Diagramme de cas d'utilisation du directeur de la clinique

2.4.5.1.3 Description textuelle de quelques cas d'utilisation

2.4.5.1.3.1 Cas d'utilisation « S'authentifier »

Sommaire d'identification	
Résumé : Permet à un utilisateur de se connecter à l'application web	
Acteurs : Tous les utilisateurs	Date de création :05/07/2022
Responsable : ADJIMON Arnaud F.	Version : 2.0
DESCRIPTION DES ENCHAINEMENTS	
Préconditions	
<ul style="list-style-type: none"> ✓ L'application web est fonctionnelle ✓ Le Client n'a pas accès aux ressources 	
Scénario nominal	
<ol style="list-style-type: none"> 1) L'Utilisateur saisit ses identifiants de connexion 2) Le Système vérifie les identifiants 3) Les informations sont correctes 4) L'Utilisateur est redirigé vers la page d'accueil 	

Sommaire d'identification
Enchainements alternatifs
✓ A1 : L'Utilisateur ne saisit pas ses identifiants de connexion ainsi il ne peut pas être connecté à l'application. Ce scénario se déroule au point 1.
Enchainements d'erreurs
✓ E1 : Les identifiants saisis par l'Utilisateur sont incorrects. Le système renvoie un message d'erreur. Ce scénario se déroule au point 3
Post conditions
✓ L'utilisateur a accès à l'application

2.4.5.1.3.2 Cas d'utilisation « Enregistrer une facture »

Sommaire d'identification	
Résumé : Permet à un caissier d'enregistrer une facture	
Acteurs : Caissier	Date de création :28/07/2022
Responsable : ADJIMON Arnaud F.	Version : 2.0
DESCRIPTIONS DES ENCHAINEMENTS	
Préconditions	
<ul style="list-style-type: none"> ✓ L'application web est fonctionnel ✓ L'Utilisateur est connecté 	
Scénario nominal	
<ol style="list-style-type: none"> 1) Le Caissier renseigne et valide le formulaire de création de la facture 2) Le Système vérifie les informations fournies 3) La facture est enregistrée 4) Un message de succès est renvoyé par l'application 5) Le Caissier sélectionne la facture concernée 6) Le Système envoie une requête pour récupérer les informations du patient 7) Le Caissier remplit le formulaire d'ajout d'une ligne facture 8) La ligne facture est enregistrée 9) Un message de succès est renvoyé par le Système 	
Enchaînements alternatifs	
<ul style="list-style-type: none"> ✓ A1 : L'Utilisateur ne fournit pas toutes les informations requises. L'application notifie l'Utilisateur de remplir le champ. Ce scénario se déroule au point 2. 	
Enchaînements d'erreurs	
<ul style="list-style-type: none"> ✓ E1 : Les informations envoyées sont incorrectes. Le Système renvoie un message d'erreur. Ce scénario se déroule au point 2. 	
Post conditions	
<ul style="list-style-type: none"> ✓ La facture est enregistrée ✓ Le Caissier imprime la facture. 	

2.4.5.2 Les diagrammes d'activité

Le diagramme d'activité est un diagramme comportemental d'UML, permettant de représenter le déclenchement d'événements en fonction des états du système et décrire un flux de travail.

2.4.5.2.1 Diagramme d'activité « S'authentifier »

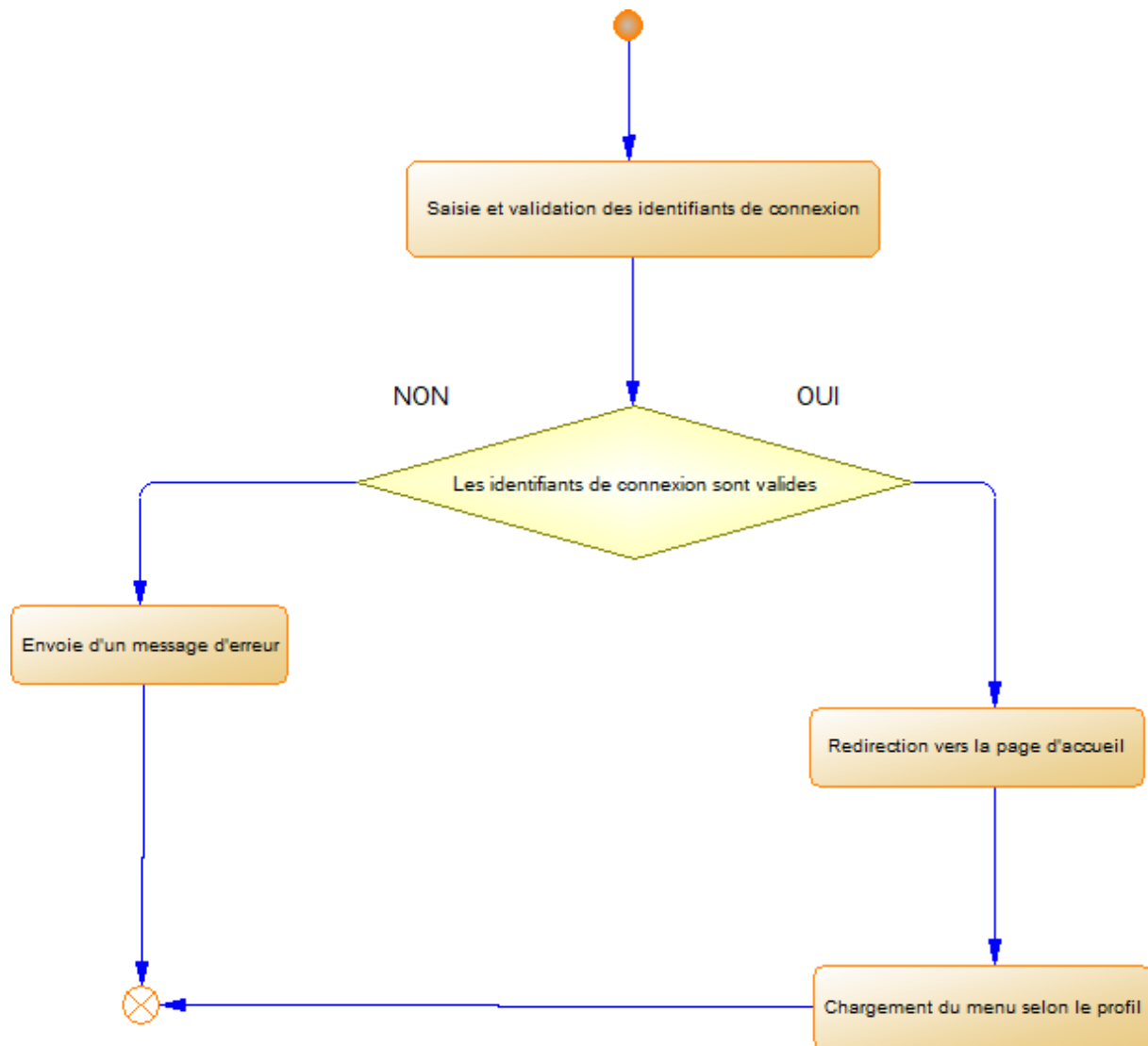


Figure 5 : Diagramme d'activité « S'authentifier »

2.4.5.2.2 Diagramme d'activité « Enregistrer une facture »

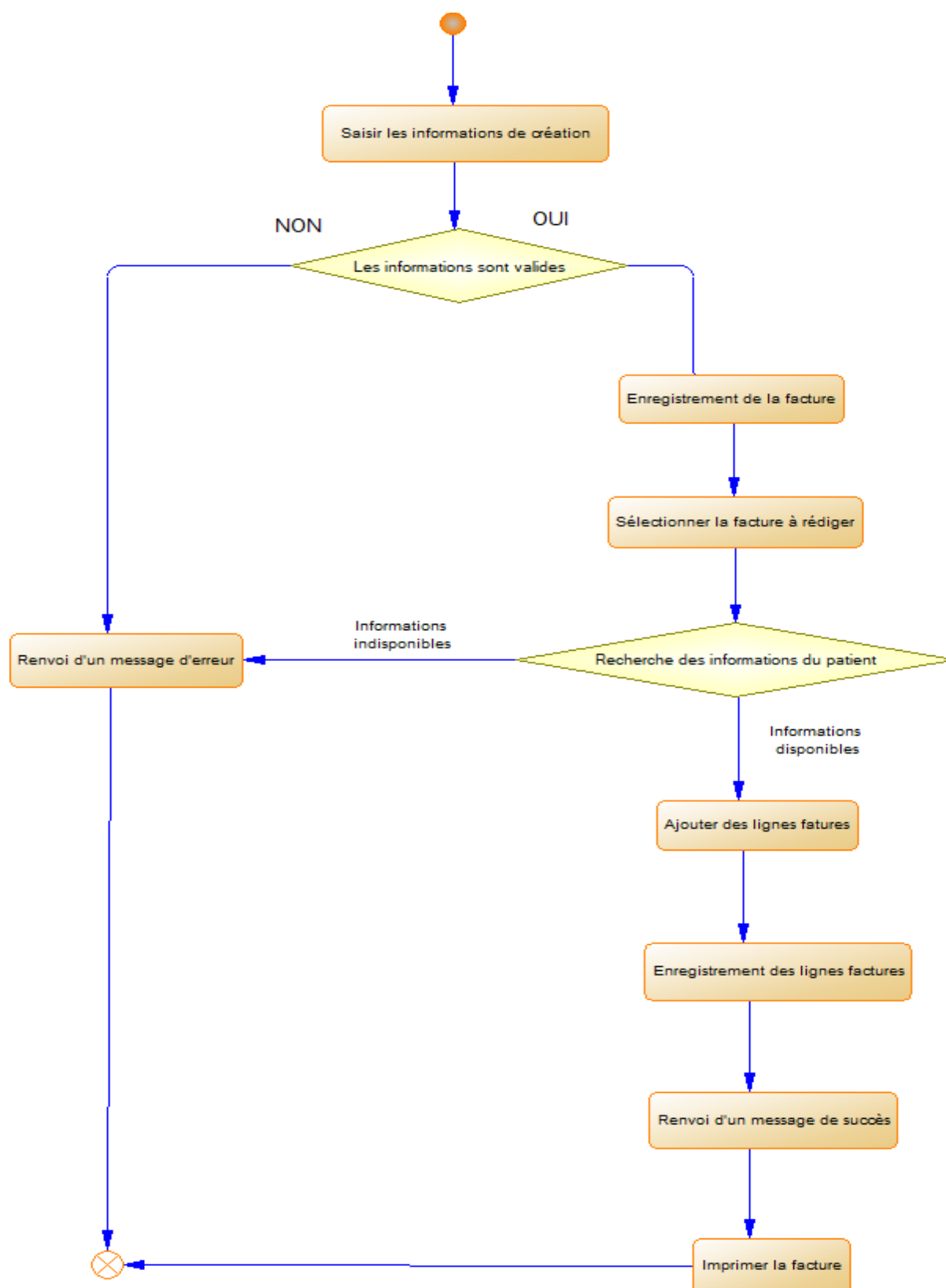


Figure 6 : Diagramme d'activité « Enregistrer une facture »

2.4.5.3 Les diagrammes de séquences

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique dans la formulation Unified Modeling Language.

2.4.5.3.1 Diagramme de séquences « S'authentifier »

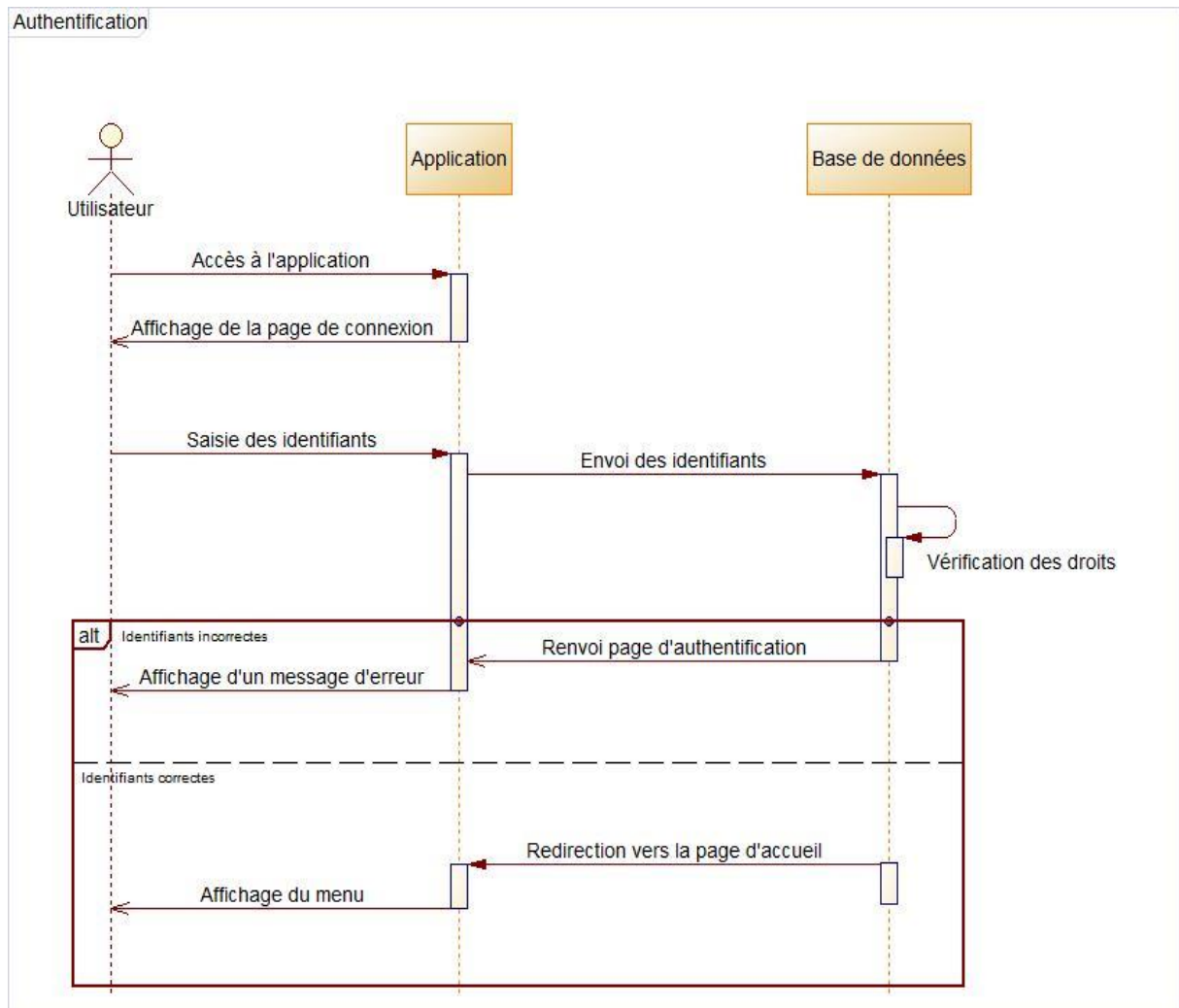


Figure 7 : Diagramme de séquence « S'authentifier »

2.4.5.3.2 Diagramme de séquence « Enregistrer une facture »

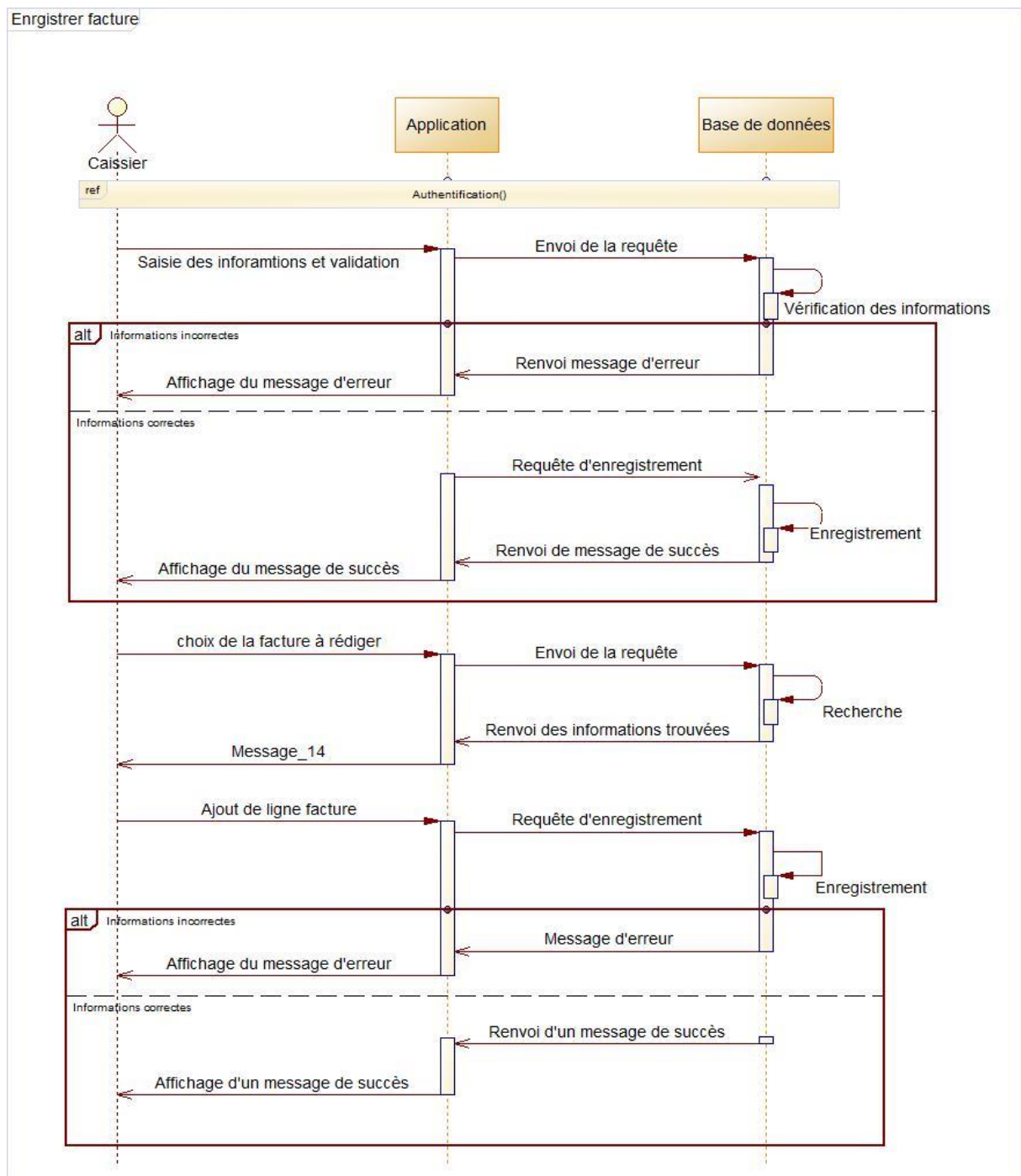


Figure 8 : Diagramme de séquence « Enregistrer une facture »

2.4.6 Capture des besoins techniques

La capture des besoins techniques représente la première étape de la branche droite du processus 2TUP.

Elle permet de compléter la capture des besoins fonctionnels de la branche droite dudit processus en définissant des contraintes indépendantes des utilisateurs que l'application se doit de prendre en compte comme les contraintes de performance ou de développement.

Recensement des cas d'utilisation techniques

Un cas d'utilisation technique est une séquence d'actions produisant une valeur ajoutée opérationnelle ou purement technique.

Ces derniers sont indépendants des cas d'utilisation obtenus au niveau de la capture des besoins fonctionnels.

- Les données doivent être cohérentes ;
- Plusieurs utilisateurs doivent pouvoir utiliser le service en même temps ;
- Les ressources doivent être disponibles à tout moment ;
- Les ressources sensibles du système doivent être protégées par la mise en place de rôles et de permissions ;

2.4.7 Conception

2.4.7.1 Le diagramme de classes

Le diagramme de classes est un diagramme statique d'UML utilisé pour représenter les classes et les interfaces des systèmes de même que leurs relations.

Gestion des analyses et de la facturation des actes médicaux

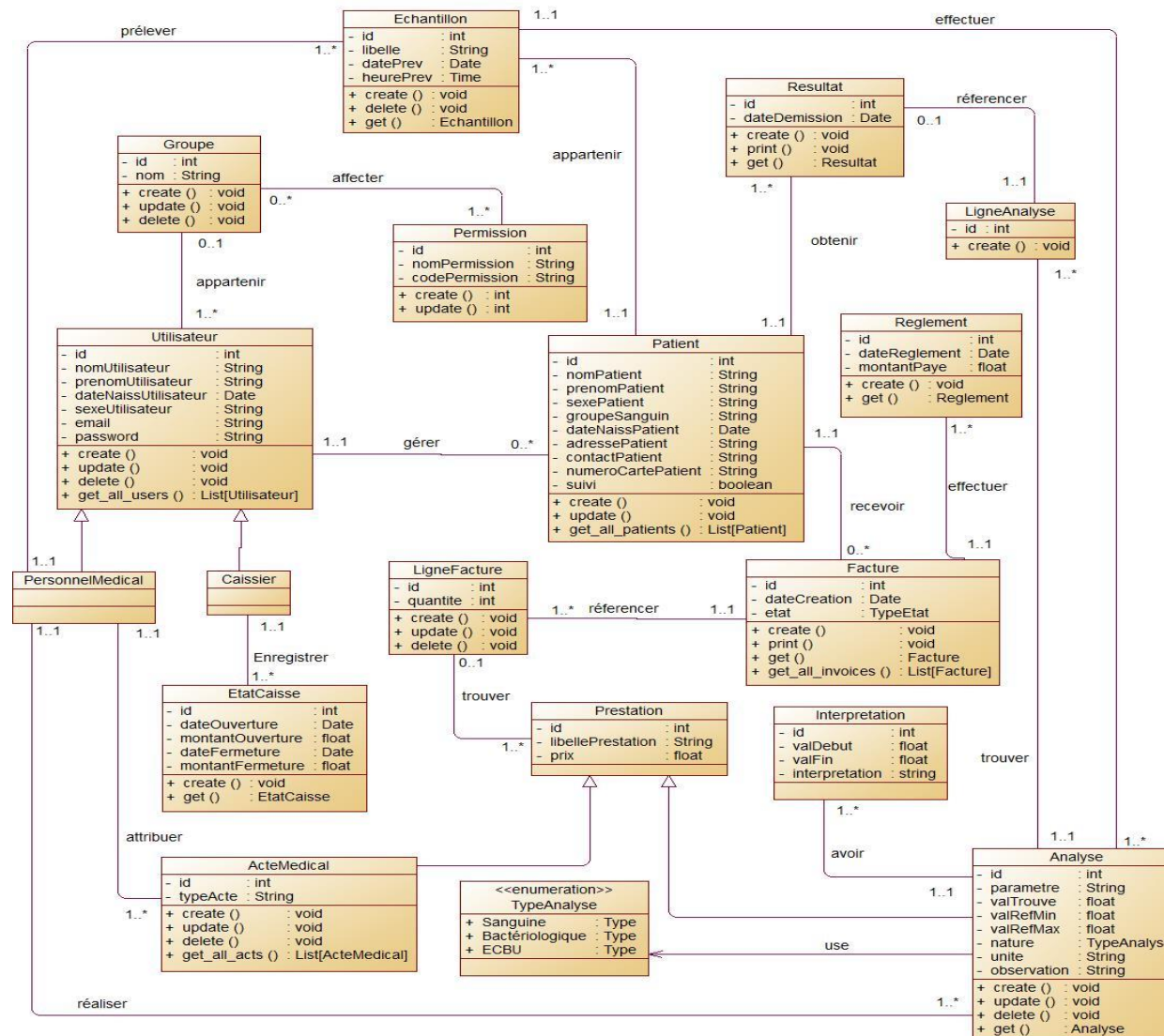


Figure 9 : Diagramme de classe finale du système

PARTIE 3 : REALISATION ET MISE EN OEUVRE

Dans cette partie nous ferons une brève présentation des différents outils tant matériels que logiciels que nous avons eu à utiliser durant la phase de programmation de notre stage.

3 Réalisation et mise en œuvre

3.1 Matériels et logiciels utilisés

3.1.1 Matériels utilisés

❖ Ordinateur portable

Notre système a été réalisé avec un ordinateur portable Pavilion 15-cs3xxx de marque HP (Hewlett Packard) disposant des caractéristiques suivantes :

- Système d'exploitation : Microsoft Windows 11 Home ;
- Type du système d'exploitation : Système d'exploitation 64 bits ;
- Processeur : Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz, 1190 Mhz ;
- Mémoire RAM installée : 8.00 Go ;
- Capacité du disque dur : 477 Go ;
- Type du disque dur : SSD ;
- Cartes graphiques : Intel(R) UHD Graphics ;
- Dalle : 1920 x 1080

3.1.2 Logiciels utilisés

Nous présentons dans cette section, les différents logiciels et technologies que nous avons utilisé tout au long de notre stage.

3.1.2.1 Backend

Le backend est un terme généralement utilisé pour décrire les ressources tant matérielles que logicielles mises en œuvre en arrière-plan qui alimentent une application ou un site web.

Il comporte un grand nombre d'éléments comme les serveurs d'application, de bases de données, et les langages de programmation.

3.1.2.1.1 Python

Python est un langage de programmation puissant et facile à apprendre. Il dispose de structures de données de haut niveau et permet une approche simple mais efficace de la programmation orientée objet.

❖ Justifications

- Facile à prendre en main ;
- Rapide ;
- Syntaxe légère ;
- Compatible avec la plupart des systèmes et plateformes.

❖ Alternatives

- Php ;
- Java ;
- C#

3.1.2.1.2 Django

Django est un framework python open-source consacré au développement web 2.0.

❖ Justifications

- Dispose d'une documentation détaillée ;
- Facile à prendre en main ;
- Facilité de maintenance ;
- Sécurité finale ;
- Supporter par les éditeurs de codes ;
- Compatible avec la plupart des technologies.

❖ Alternatives

- Flask ;
- Pyramid ;
- TurboGears.

3.1.2.1.3 Pipenv

Pipenv est un outil de packaging qui résout certains problèmes courants associés au flux de travail typique utilisant pip, virtualenv et le requirements.txt.

❖ Justifications

- Permet de gérer les packages et les dépendances des projets avec python ;
- Permet de fixer les versions des dépendances utilisées et de créer facilement un environnement lors du développement ;
- Permet que les dépendances de l'environnement de développement soient identiques à celles de l'environnement de production ;
- Dispose de la gestion d'environnement virtuel intégrée.

❖ Alternatives

- Pyenv ;
- Virtualenv ;
- Autoenv.

3.1.2.1.4 AJAX

AJAX est l'acronyme de Asynchronous JavaScript and XML, ce qui signifie "Javascript asynchrone et XML" en français. Il s'agit d'une méthode de communication entre serveur et client qui favorise la mise à jour des différentes parties d'un site sans rechargement de la page entière.

❖ Justifications

- Augmente la réactivité des pages web ;
- Evite le rechargement de la page ;
- Combine plusieurs technologies (CSS, JS, HTML, DOM).

❖ Alternatives

- Flash et Flex ;
- Silverlight ;
- JavaFX.

3.1.2.1.5 JQuery

jQuery est une bibliothèque JavaScript libre et multiplateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML des pages web.

❖ Justifications

- Implémente une interface de haut niveau pour faire des requêtes AJAX ;
- Facilite l'interaction des éléments HTML ;

❖ Alternatives

- Zepto ;
- Cash ;
- vQuery ;
- Umbrella.

3.1.2.1.6 Xhtml2pdf

Xhtml2pdf permet de générer facilement des documents PDF à partir de contenu HTML et avec un contrôle de flux automatisé tel que la pagination et la cohésion du texte. Le module Python peut être utilisé dans n'importe quel environnement Python, y compris Django.

❖ Justifications

- Intégration facile dans les frameworks python ;
- Facile à apprendre ;
- Rapide ;

❖ Alternatives

- ReportLab ;
- PyPdf.

3.1.2.1.7 PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle et objet.

❖ **Justifications**

- Il est libre ;
- Il est plus avancé dans la conformité aux standards SQL ;
- Il fonctionne sur diverses plates-formes matérielles et sous différents systèmes d'exploitation.

❖ **Alternatives**

- MySQL ;
- MariaDB ;
- SQLite ;
- Oracle.

3.1.2.1.8 **PgAdmin**

PgAdmin est une plateforme Open Source d'administration pour PostgreSQL.

❖ **Justifications**

- Il fonctionne sur plusieurs systèmes d'exploitation ;
- Il dispose d'un large panel de fonctionnalités et est la plateforme d'administration la plus populaire pour PostgreSQL ;

❖ **Alternatives**

- PostBird ;
- PopSQL ;
- Navicat.

3.1.2.1.9 **Visual Studio Code**

Visual Studio Code est un éditeur de code développé par la société Microsoft pour Windows, Linux et MacOS.

❖ **Justifications**

- Il offre une multitude d'extension permettant de faciliter le contrôle de version, la détection et l'affichage de message d'erreurs, la complétion intelligente de code ou la combinaison des projets.

❖ **Alternatives**

- Atom ;
- Pycharm ;
- Notepad ++.

3.1.2.2 Frontend

3.1.2.2.1 Bootstrap

Bootstrap est une collection d'outils utiles à la création du design de sites et d'applications web.

❖ **Justifications**

- S'adapte dynamiquement au format des supports depuis lesquels on accède (PC, tablette, smartphone) ;
- Compatible avec tous les navigateurs ;
- Il est personnalisable grâce au Sass ;
- Gain de temps de développement des interfaces.

❖ **Alternatives**

- TailwindCSS ;
- HTML KickStart ;
- Zimit.

3.1.2.3 Gestion de version

3.1.2.3.1 Git

Git est un logiciel libre et open source de gestion de versions. Il est utilisable via un terminal ou une application desktop disposant d'une interface utilisateur graphique.

❖ **Justifications**

- Permet de conserver un historique des modifications et des versions de tous les fichiers ;
- Rend facile le retour en arrière en cas de problème ;

- Offre la possibilité de travailler à plusieurs sur un même projet sans risquer de supprimer les modifications des autres collaborateurs

❖ Alternatives

- Azure Devops Server ;
- Helix Core ;
- AWS Code Commit

3.1.2.3.2 Github

Selon Wikipédia, Github est un service web d'hébergement et de gestion de développement de logiciels, utilisant le logiciel de gestion de versions Git.

❖ Justifications

- Permet d'avoir un historique des modifications et des versions de tous les fichiers et de les sauvegarder dans le cloud ;
- Permet de suivre les changements dans le code à chaque version

❖ Alternatives

- Bitbucket ;
- GitLab ;
- Google Cloud Source Repositories

3.2 Sécurité de l'application

Notre application manipule un grand nombre d'informations confidentielles et constitue donc un service critique dont la protection doit être une priorité. Nous avons eu à utiliser un certain nombre de technologies afin d'assurer la sécurité de notre application.

3.2.1 Authentification et autorisation

Pour accéder aux diverses ressources de l'application nous avons défini des règles que chaque utilisateur doit respecter. Nous avons eu à implémenter le système d'authentification par défaut du framework django contenu dans le package « django.contrib.auth ». Le système "auth" de django nous fournit donc les fonctionnalités suivantes :

- ✓ Restreindre l'accès à certains utilisateurs ;
- ✓ Enregistrement des profils d'utilisateurs ;
- ✓ Le hachage des mots de passes ;
- ✓ L'attribution des autorisations à des utilisateurs et des groupes d'utilisateurs spécifiques ;
- ✓ La vérification de la validité des informations d'identification.

3.2.2 Hypertext Transfer Protocol Secure (HTTPS)

Notre application étant web, nous avons mis en place le protocole HTTPS afin de sécurisée l'échange des données entre le client et le serveur.

Qu'est-ce que le HTTPS ?

HTTPS est une version chiffrée du protocole HTTP. Il permet entre autres de chiffrer l'intégralité des communications entre un client et un serveur. La connexion sécurisée permet aux clients d'échanger de manière sûre des données sensibles avec un serveur.

3.3 Evaluation financière de la solution

Afin de pallier les insuffisances précédemment remarquées, nous avons opté pour le développement d'un logiciel pour gérer la facturation des actes médicaux et l'émission des résultats d'analyses des patients, à partir de l'analyse des besoins et des langages de programmation.

3.3.1 Coût matériel de la solution

Tableau 2 : Coût matériel de la solution

Désignation	Description	Coût unitaire	Quantité	Montant (FCFA)
Achat d'un ordinateur pour la conception et la mise en œuvre et l'administration de la plateforme	Pc HP Pavilion 15-cs3xxx Intel Core i5 8 Go RAM 512 Go SSD	500 000	1	500 000
Connexion Internet	FAI (Fournisseur d'Accès à internet) Canalbox Premium 50 mb/s	30 000	2	60 000
TOTAL				560 000

3.3.2 Développement de la solution

Tableau 3 : Coût du développement de la solution

Désignation	Description	Coût unitaire	Quantité	Montant (FCFA)
Développement de la solution	Analyse, conception, développement et tests de l'application	5000	416 (8 heures * 52 jours)	2 080 000
Hébergement annuel de la base de données	Plan standard Heroku Postgres (64 Go de stockage et 4 Go de RAM)	32 207	12	386 484
Formation du personnel	Formation du personnel à l'utilisation du logiciel	20 000	12 (4 heures * 3 jours)	240 000
TOTAL				2 706 484

3.3.3 Coût total de la solution

Tableau 4 : Coût total de la solution

Désignation	Montant (FCFA)
Achat d'un ordinateur	500 000
Connexion Internet	60 000
Développement de la solution	2 080 000
Formation du personnel	240 000
TOTAL	2 880 000
Frais d'hébergement sur 1 an	
Hébergement annuel de la base de données	386 484
TOTAL	3 266 484

3.4 Présentation de l'application

❖ Description textuelle du logiciel

Notre application a pour but de permettre au personnel de la clinique de gérer la patientèle par la prise en charge, l'élaboration des résultats d'analyses et la facturation des actes médicaux effectués.

Ses objectifs sont :

- De concevoir des factures en fonction des actes médicaux effectués par un patient ;
- De suivre les états des factures délivrées au niveau de la comptabilité ;
- D'émettre une fiche de résultat d'analyse destinée à un patient à partir des prélèvements enregistrés ;
- De retrouver l'historique d'un patient grâce aux dossiers médicaux.

❖ Architecture logicielle de l'application

Notre application web étant développée avec la technologie Django, elle suit l'architecture logicielle **Model-View-Template**.

Qu'est-ce que le MVT ?

Le Model-View-Template est un design pattern qui s'inspire du MVC, qui vous permet de créer des modèles qui génèrent des bases de données et rendent l'interface utilisateur dynamiques à l'aide des vues HTML.

Ce pattern identifie trois objets distincts :

- **Le modèle** : il interagit avec une base de données via un ORM.
- **La vue** : elle reçoit une requête et renvoie une réponse. Si la requête est une interaction entre la base de données, la vue appelle un modèle pour récupérer les items demandés.
- **Le template** : il s'agit d'un fichier HTML récupéré par la vue et envoyé à l'utilisateur avec les données des modèles.

3.4.1 Mises en place de la base de données

Pour la mise en place de notre base de données, nous avons utilisé l'ORM de Django qui nous permet de créer des tables à partir des classes écrites en python et

de faciliter la connexion à la base de données ainsi que les migrations des clés entre les tables.

Dans cette partie nous ferons la présentation de quelques classes et fonctions qui nous ont permis de mettre en place la base de données obtenue à partir de notre analyse.

3.4.1.1 Le fichier **manage.py**

Ce fichier contient un script qui permettra d'exécuter les commandes utiles au sein de notre projet

3.4.1.2 Le fichier **settings.py**

Ce fichier nous permet de créer une connexion à notre base de données avec du code python et des identifiants de notre base de données, et de définir la configuration globale du projet.

3.4.1.3 Le fichier **models.py**

Les modèles Django sont des classes héritées de la classe Model du framework de base, qui leur confère ainsi les propriétés et méthodes relatives aux modèles.

3.4.1.4 Les opérations **CRUD** et les requêtes

❖ Les opérations **CRUD**

Django intègre une interface d'administration qui permet d'effectuer des opérations **CRUD** sur les différentes classes créées lors des migrations pour en faire des tables de bases de données. Celle-ci est configurable dans le fichier **admin.py**.

❖ Les requêtes

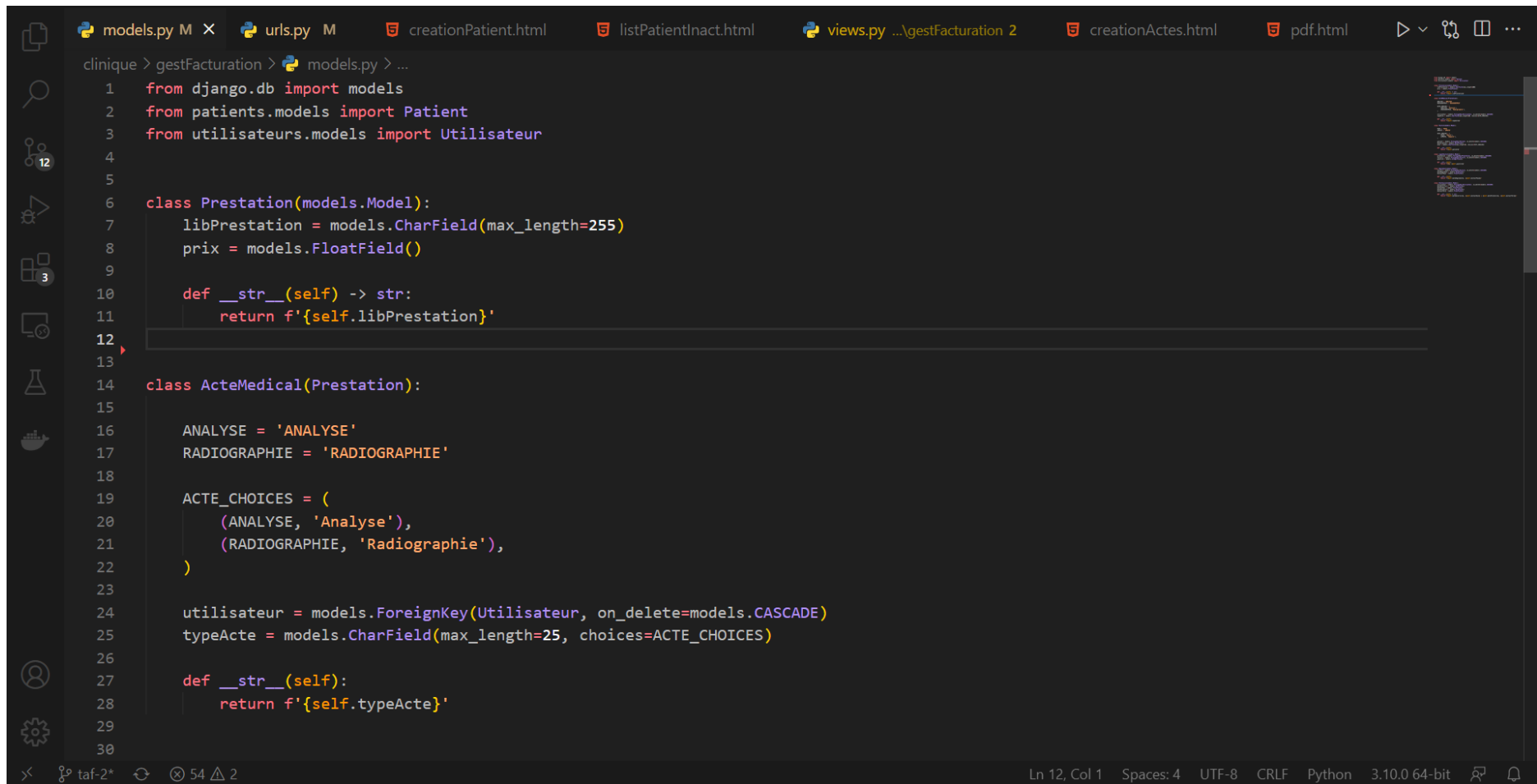
Le manager **Objects** de django nous permet de gérer également les opérations CRUD de la base de données.

3.4.1.5 Quelques exemples montrant la mise en place de notre base de données

```
clinique > clinique > settings.py > ...
80     ]
81
82     WSGI_APPLICATION = 'clinique.wsgi.application'
83
84
85     # Database
86     # https://docs.djangoproject.com/en/4.0/ref/settings/#databases
87
88     DATABASES = {
89         'default': {
90             'ENGINE': 'django.db.backends.postgresql',
91             'NAME': 'gestionclinic',
92             'USER': 'naud',
93             'PASSWORD': 'naud.2002',
94             'HOST': 'localhost',
95             'PORT': 5432,
96         }
97     }
98
```

Figure 10 : Fichier settings.py de connexion à la base de données

Gestion des analyses et de la facturation des actes médicaux



```
1 from django.db import models
2 from patients.models import Patient
3 from utilisateurs.models import Utilisateur
4
5
6 class Prestation(models.Model):
7     libPrestation = models.CharField(max_length=255)
8     prix = models.FloatField()
9
10     def __str__(self) -> str:
11         return f'{self.libPrestation}'
12
13
14 class ActeMedical(Prestation):
15
16     ANALYSE = 'ANALYSE'
17     RADIOGRAPHIE = 'RADIOGRAPHIE'
18
19     ACTE_CHOICES = (
20         (ANALYSE, 'Analyse'),
21         (RADIOGRAPHIE, 'Radiographie'),
22     )
23
24     utilisateur = models.ForeignKey(Utilisateur, on_delete=models.CASCADE)
25     typeActe = models.CharField(max_length=25, choices=ACTE_CHOICES)
26
27     def __str__(self):
28         return f'{self.typeActe}'
29
30
```

Figure 11 : Exemple de modèle du fichier models.py

Gestion des analyses et de la facturation des actes médicaux

```
clinique > gestFacturation > admin.py > LigneFactureAdmin
1  from django.contrib import admin
2  from gestFacturation.models import Facture, LigneFacture, ActeMedical, Reglement
3
4
5  admin.site.site_header = 'Gestion Clinique'
6
7
8  class FactureAdmin(admin.ModelAdmin):
9      list_display = ('patient', 'dateCreation', 'etat')
10
11  admin.site.register(Facture, FactureAdmin)
12
13
14  class LigneFactureAdmin(admin.ModelAdmin):
15      list_display = ('prestation', 'quantite', 'facture')
16
17  admin.site.register(LigneFacture, LigneFactureAdmin)
18
19
20  class ActeMedicalAdmin(admin.ModelAdmin):
21      list_display = ('utilisateur', 'typeActe')
22
23  admin.site.register(ActeMedical, ActeMedicalAdmin)
24
25
26  class ReglementAdmin(admin.ModelAdmin):
27      list_display = ('facture', 'montantPaye', 'dateReglement')
28
29  admin.site.register(Reglement, ReglementAdmin)
30
```

Figure 12 : Fichier admin.py pour la configuration de l'interface admin

django Administration Django

- BIENVENUE, ADJIMON VOIR LE SITE / MODIFIER LE MOT DE PASSE / DÉCONNEXION

Accueil > Gestfacturation > Reglements > 2022-08-03, (2000.0)

Écrivez ici pour filtrer...

AUTHENTIFICATION ET AUTORISATION

Groupe

+ Ajouter

GESTANALYSE

Analyses

+ Ajouter

Echantillons

+ Ajouter

Interpretations

+ Ajouter

Resultats

+ Ajouter

GESTFACTORATION

Acte medicals

+ Ajouter

Factures

+ Ajouter

Ligne factures

+ Ajouter

Reglements

+ Ajouter

INTERFACE ADMIN

« Modification de reglement

HISTORIQUE

2022-08-03, (2000.0)

Facture : ABALO Imelda

DateReglement : 03/08/2022 Aujourd'hui

MontantPayé : 2000.0

Supprimer

Enregistrer et ajouter un nouveau

Enregistrer et continuer les modifications

ENREGISTRER

Figure 13 : Interface administrateur de gestion des opérations CRUD des modèles Django

3.4.2 Plan de navigation

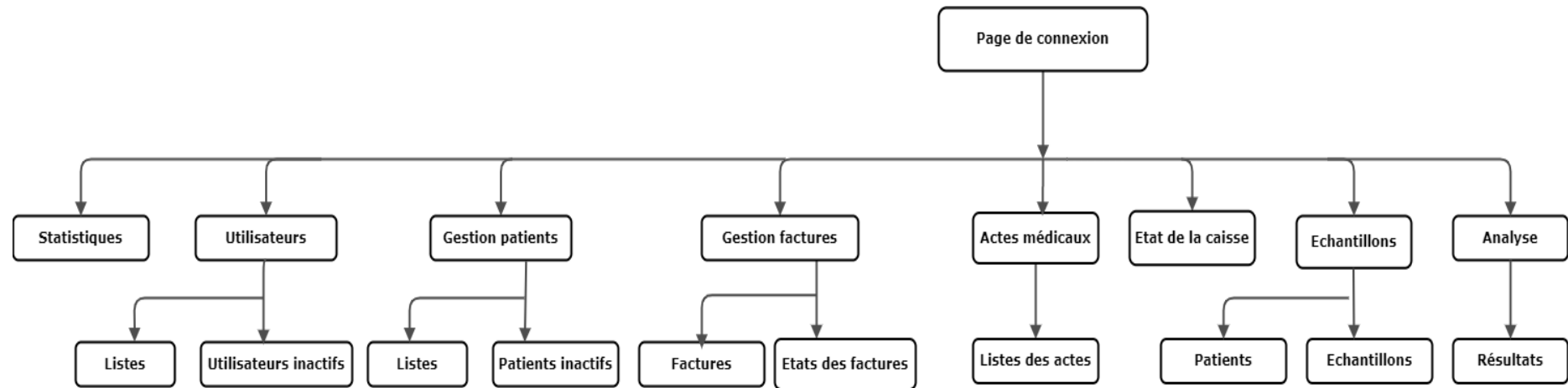


Figure 14 : Plan de navigation de l'application

3.4.3 Quelques masques de saisie

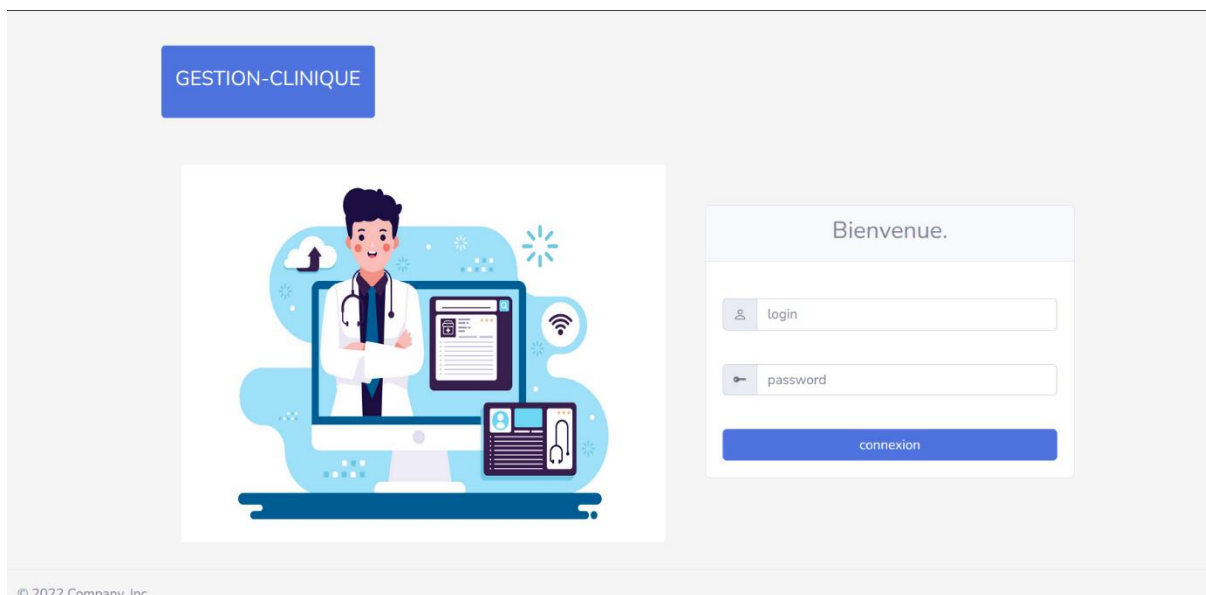


Figure 15 : Page de connexion

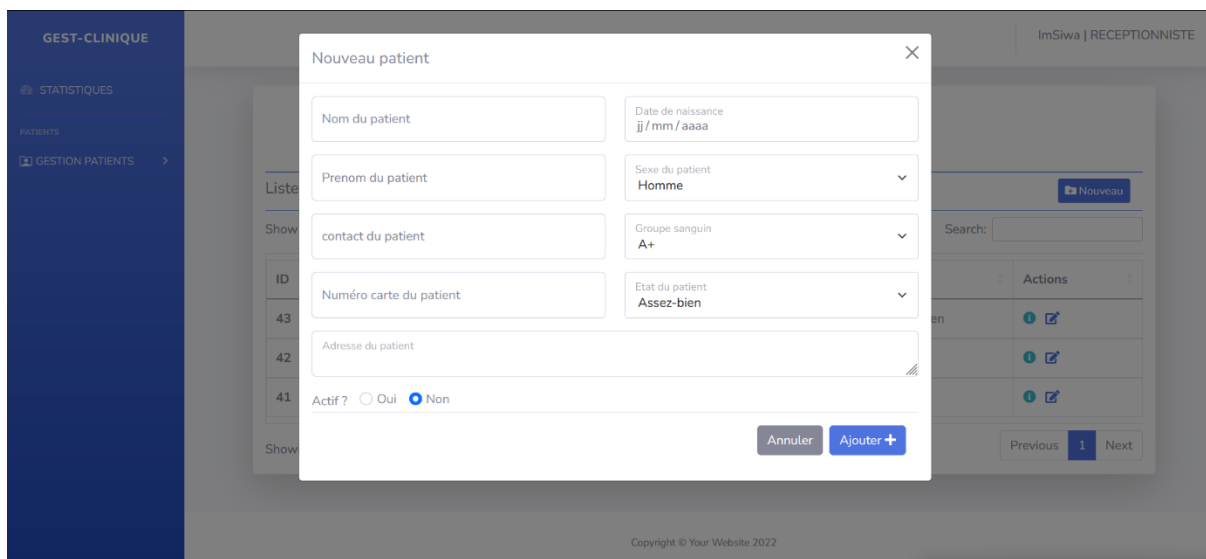


Figure 16 : Formulaire de validation d'un patient

Gestion des analyses et de la facturation des actes médicaux

The screenshot shows the 'Ajouter une analyse' (Add analysis) form in the GEST-CLINIQUE application. The form is a modal window with a close button (X) in the top right corner. It contains several input fields: 'Nom du paramètre' (Parameter name), 'Unité' (Unit), 'Valeur trouvée' (Found value), 'Référence minimale' (Minimum reference), 'Référence maximale' (Maximum reference), 'Nature' (Nature) with a dropdown menu showing 'Sanguine', 'Observation', 'Libellé de la prestation' (Service label), and 'Prix' (Price). At the bottom right, there are two buttons: 'Annuler' (Cancel) and 'Ajouter +' (Add +). The background shows a sidebar with navigation links like 'STATISTIQUES', 'PRELEVEMENT', 'ECHANTILLONS', and 'ANALYSE', and a main area with a search bar and a table of results.

Figure 17 : Formulaire de validation d'un résultat d'analyse

The screenshot shows the 'Nouvelle Ligne Facture' (New Invoice Line) form in the GEST-CLINIQUE application. The form is a modal window with a close button (X) in the top right corner. It contains a dropdown menu for 'prestation' (Service) with 'Glycosurie' selected, and a 'Quantité' (Quantity) input field. At the bottom right, there are two buttons: 'Annuler' (Cancel) and 'Ajouter +' (Add +). The background shows a sidebar with navigation links like 'STATISTIQUES', 'FACTURATION', 'GESTION FACTURES', 'ACTES MEDICAUX', and 'CAISSE', and a main area with a patient information section and a table of invoice lines.

Figure 18 : Formulaire d'ajout d'une ligne facture

3.4.4 Quelques états et statistiques

Gestion des analyses et de la facturation des actes médicaux

GEST-CLINIQUE ImSiwa | RECEPTIONNISTE

GESTION PATIENTS

Listes des patients [Nouveau](#)

Show **10** entries Search:

ID	Nom	Sexe	Naissance	Etat	Actions
43	AHLOU Honorine	Homme	22 février 2002	Assez-bien	Info Edit
42	ABALO Imelda	Femme	22 octobre 2002	Critique	Info Edit
41	ALPAN Fitness	Homme	29 juin 2002	Critique	Info Edit

Showing 1 to 3 of 3 entries [Previous](#) **1** [Next](#)

Copyright © Your Website 2022

Figure 19 : Page présentant la liste des patients

GEST-CLINIQUE LaTheo | TECHNICIEN

RESULTAT N° 4

Patient : ALPAN Fitness [Nouveau](#)

Prélèvement	Paramètre	Val. Trouve	Val. Reference
Bandelette Urinaire	Acide Urique	380,0	400,0 - 800,0
Ponction lombaire	Ig totales	25,0	6,3 - 33,5

[Précédent](#) [PDF](#)

Copyright © Your Website 2022

Figure 20 : Fiche numérique d'un résultat d'analyses effectuées par un patient

Gestion des analyses et de la facturation des actes médicaux

analyse(0).pdf

CLINIQUE GEST-CLINIC

LABORATOIRE D'ANALYSES DE BIOLOGIE MEDICALE

Patient: ALPAN Fitness
Emis: le 8 août 2022
Examen n°: 4

PRELEVEMENT	PARAMETRE	VAL. TROUVEE	REFERENCE	UNITE	OBSERVATION
Bandelette Urinaire	Acide Urrique	360,0	400,0 - 800,0	mg/24h	Bas
Ponction lombaire	Ig totales	25,0	6,3 - 33,5	g/L	Normal

Bon rétablissement

Figure 21 : Fiche imprimable du résultat d'analyse

GEST-CLINIQUE

Ajavi | CAISSIER

FACTURE N° 6

Patient : ALPAN Fitness

Acte medical	P.U	Qte	Montant Total
Prise de sang	1000,0	1	1000,0
Scanner du genou	15000,0	2	30000,0
Numération blanche	1500,0	4	6000,0
Ponction lombaire	5000,0	1	5000,0
Total :			42000,0

Précédent PDF

Figure 22 : Facture numérique recensant les actes médicaux effectués par un patient

Gestion des analyses et de la facturation des actes médicaux

Clinique GEST-CLINIC

FACTURE DES ACTES MEDICAUX

Patient: ALPAN Fitness
Date: 30 juillet 2022
Facture n°: 6

ACTE MEDICAL	QTE	P.U	TOTAL
Prise de sang	1	1000,0	1000,0
Scanner du genou	2	15000,0	30000,0
Numération blanche	4	1500,0	6000,0
Ponction lombaire	1	5000,0	5000,0
TOTAL:			42000,0

Bon rétablissement

Figure 23 : Fiche imprimable de la facturation des actes médicaux d'un patient

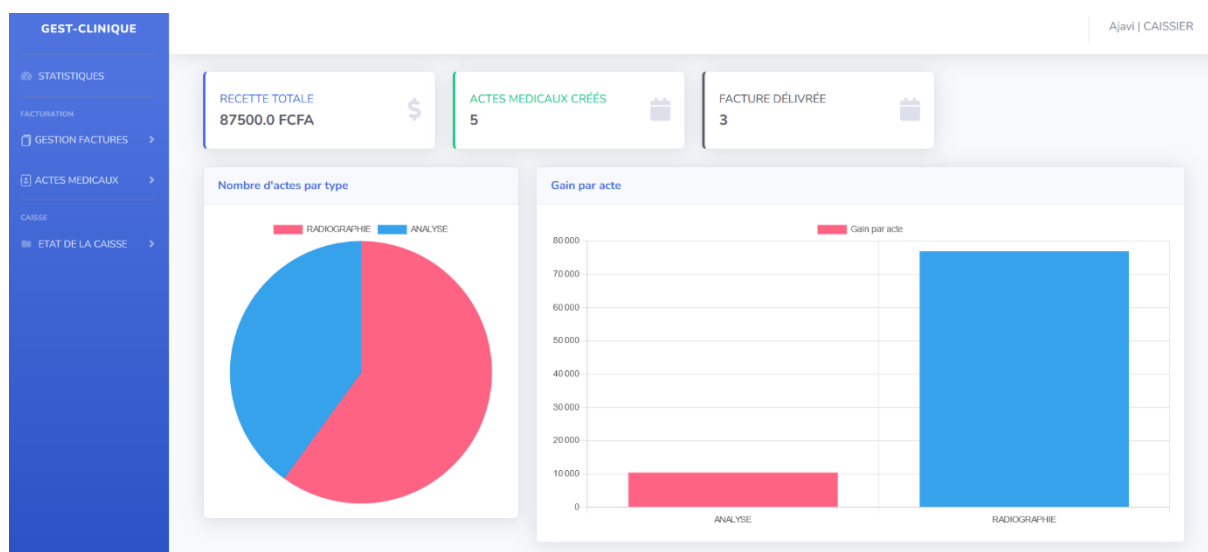


Figure 24 : Page des statistiques du caissier

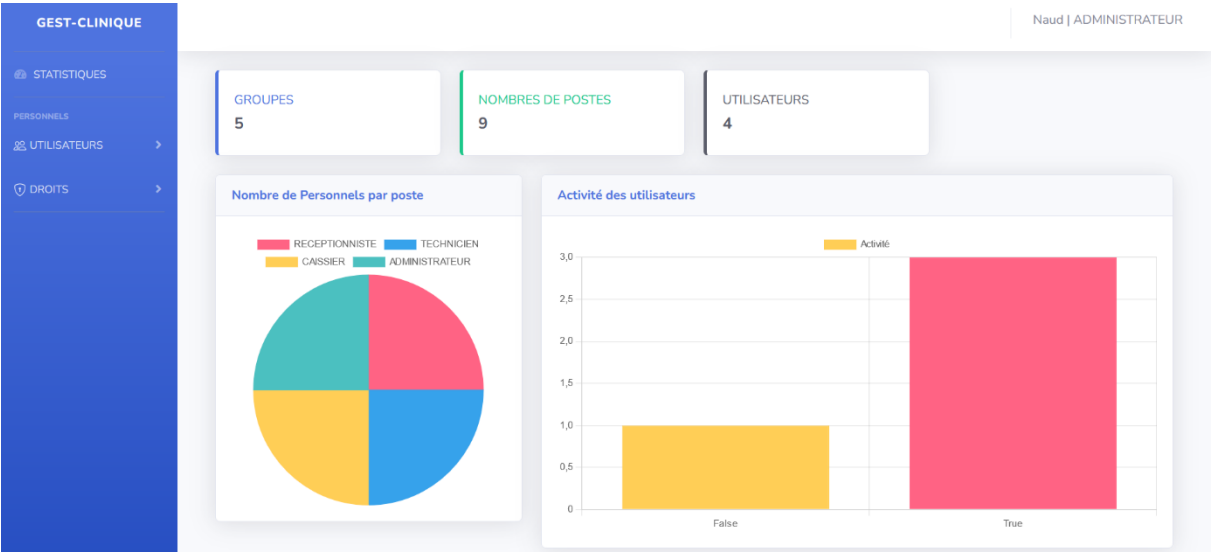


Figure 25 : Page des statistiques de l'administrateur

CONCLUSION

Nous avons eu l'occasion d'effectuer un stage d'une durée de deux mois de l'entreprise **INSIGHT** qui souhaite mettre à la disposition d'une clinique, une application qui facilite la gestion des patients et des différents actes médicaux effectués, qui automatise la facturation et l'élaboration des résultats d'analyses du laboratoire. Nous nous sommes attelés à la mise en œuvre de cette solution.

Nous avons conçu notre système à l'aide des Framework **Django** basé sur le langage de programmation **Python** et **Bootstrap** basé sur le **CSS** et le **HTML**. Lors de la réalisation, nous avons passé plus de temps sur l'analyse et la conception ce qui impacté notre temps de développement. Néanmoins, cela nous a permis de progresser sans difficultés et facilité la main pendant la phase de codage.

Ces deux mois nous ont permis de mettre en pratique les enseignements acquis au sein de **l'Institut Africain d'Informatique** et d'approfondir notre connaissance en programmation. Ce stage nous a permis également de nous confronter au monde professionnel et à ses réalités.

BIBLIOGRAPHIE INDICATIVE

❖ Notes de cours

- Droit des TIC, de **M. AMOUDOKPO Edem** Enseignant à l'IAI-TOGO
- Méthodes et processus de développement, de **M. SEWAVI Maurice** Enseignant à l'IAI-TOGO
- Rédaction Scientifique, de **M. GBODUI Roland Joseph**

❖ Anciens rapports et mémoires consultés

- « GESTION DE LA FACTURATION » de **AYITE Yawo Mawulom Guillaume Florian** (2019 – 2020) ;
- « MISE EN PLACE D'UN SERVICE DE CONGES ET D'EVALUATION DES PERFORMANCES DU PERSONNEL D'UNE ENTREPRISE » de **AYITE Yawo Mawulom Guillaume Florian** (2020 – 2021) ;

WEBOGRAPHIE INDICATIVE

LIENS	DATE DE CONSULTATION
https://www.djangoproject.com/	Tout au long du projet
https://stackoverflow.com/	Tout au long du projet
https://getbootstrap.com/	Tout au long du projet
https://www.youtube.com/	Tout au long du projet
https://www.heroku.com/	Tout au long du projet

TABLE DES MATIERES

REMERCIEMENTS	i
SOMMAIRE	ii
GLOSSAIRE	iii
LISTE DES FIGURES	iv
LISTE DES TABLEAUX	v
INTRODUCTION	1
1. Cahier des charges	3
1.1 Présentations	3
1.2 Formation à l'IAI-TOGO	3
1.3 Plan de localisation	3
1.4 Présentation du sujet.....	4
1.5 Problématique du sujet.....	4
1.6 Intérêt du sujet	5
1.6.1 Objectifs.....	5
1.6.1.1 Objectif général	5
1.6.1.2 Objectifs spécifiques	5
1.6.2 Résultats attendus	6
2 Préprogrammation	8
2.1 Etude de l'existant.....	8
2.2 Critique de l'existant.....	9
2.3 Planning prévisionnel de réalisation	9
2.4 Etude détaillée de la solution	10
2.4.1 Le langage de modélisation : UML.....	10
2.4.2 Processus de développement.....	11
2.4.3 Outil de modélisation	11
2.4.4 Etude préliminaire	12
2.4.4.1 Les acteurs.....	12
2.4.5 Capture des besoins fonctionnels.....	13
2.4.5.1 Diagrammes de cas d'utilisation.....	13
2.4.5.1.1 Les cas d'utilisation	13
2.4.5.1.2 Représentation des diagrammes de cas d'utilisation	15
2.4.5.1.3 Description textuelle de quelques cas d'utilisation	17

2.4.5.2	Les diagrammes d'activité	20
2.4.5.2.1	Diagramme d'activité « S'authentifier »	20
2.4.5.2.2	Diagramme d'activité « Enregistrer une facture »	21
2.4.5.3	Les diagrammes de séquences.....	22
2.4.5.3.1	Diagramme de séquences « S'authentifier »	22
2.4.5.3.2	Diagramme de séquence « Enregistrer une facture »	23
2.4.6	Capture des besoins techniques.....	24
2.4.7	Conception.....	24
2.4.7.1	Le diagramme de classes.....	24
3	Réalisation et mise en œuvre	27
3.1	Matériels et logiciels utilisés	27
3.1.1	Matériels utilisés	27
3.1.2	Logiciels utilisés	27
3.1.2.1	Backend	27
3.1.2.1.1	Python.....	28
3.1.2.1.2	Django.....	28
3.1.2.1.3	Pipenv	29
3.1.2.1.4	AJAX.....	29
3.1.2.1.5	Jquery	30
3.1.2.1.6	Xhtml2pdf.....	30
3.1.2.1.7	PostgreSQL	30
3.1.2.1.8	PgAdmin	31
3.1.2.1.9	Visual Studio Code.....	31
3.1.2.2	Frontend.....	32
3.1.2.2.1	Bootstrap.....	32
3.1.2.3	Gestion de version	32
3.1.2.3.1	Git	32
3.1.2.3.2	Github	33
3.2	Sécurité de l'application	33
3.2.1	Authentification et autorisation	33
3.2.2	Hypertext Transfer Protocol Secure (HTTPS).....	34
3.3	Evaluation financière de la solution	34
3.3.1	Coût matériel de la solution	35
3.3.2	Développement de la solution.....	36

3.3.3	Coût total de la solution	37
3.4	Présentation de l'application	38
3.4.1	Mises en place de la base de données.....	38
3.4.1.1	Le fichier manage.py	39
3.4.1.2	Le fichier settings.py.....	39
3.4.1.3	Le fichier models.py	39
3.4.1.4	Les opérations CRUD et les requêtes	39
3.4.1.5	Quelques exemples montrant la mise en place de notre base de données 40	
3.4.2	Plan de navigation	44
3.4.3	Quelques masques de saisie.....	44
3.4.4	Quelques états et statistiques.....	46
CONCLUSION.....		51
BIBLIOGRAPHIE INDICATIVE.....		52
WEBOGRAPHIE INDICATIVE		53
TABLE DES MATIERES		54