



REPOBLIKAN'I MADAGASIKARA
FITIAVANA - TANINDRAZANA - FANDROSOANA
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



ECOLE SUPERIEURE POLYTECHNIQUE D'ANTSIRANANA

Mention STIC

PROJET DE FIN D'ETUDE

Parcours Électronique et Informatique Industrielles

Conception et réalisation du logiciel de gestion de la polyclinique universitaire NEXT

Par :

HAJALALAINA Fara Marie José

Encadreurs :

RAKOTOARISOA Jean Claude, Dr Ingénieur

RAMANAN'HAJA Hery Tina, Ingénieur

Dr Luigi BELLINI, Professeur

RAZAIARIMALALA Nirina Claudia, Chef de service

★ Année Universitaire 2018 - 2019 ★



REPOBLIKAN'I MADAGASIKARA
FITIAVANA - TANINDRAZANA - FANDROSOANA
MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR
ET DE LA RECHERCHE SCIENTIFIQUE



ECOLE SUPERIEURE POLYTECHNIQUE D'ANTSIRANANA

Mention STIC

MÉMOIRE DE FIN D'ÉTUDES

Parcours Électronique et Informatique Industrielles

Conception et realisation du logiciel de gestion de la polyclinique universitaire NEXT

Par :

HAJALALAINA Fara Marie José

Encadreurs :

Monsieur RAKOTOARISOA Jean Claude

Monsieur RAMANAN'HAJA Hery Tina

Dr Luigi BELLINI, Professeur

Madame RAZAIARIMALALA Nirina Claudia, Chef de service

Membres de jury :

Madame ANDRIAMIHARINJAKA Hasina	Président de jury
Monsieur RAKOTOARISOA Jean Claude	Encadreur
Monsieur RAMANAN'HAJA Hery Tina	Encadreur
Madame	Examineur
Monsieur	Examineur

★ Année Universitaire 2018 - 2019 ★

Promotion :

Projet et Mémoire de fin d'études – A.U : 2018/2019

Titre : Conception et réalisation du logiciel de gestion de la Polyclinique universitaire NEXT

Contexte

Dans le cadre de son programme de partenariat public privé, la polyclinique Universitaire Next s'est mise en accord de partenariat avec l'état Malagasy pour la prise en charge des fonctionnaires d'état, depuis juin 2019. Elle souhaite mettre en place un logiciel permettant de gérer les patients ainsi que les activités au sein de l'hôpital. Parmi ces activités sont compris, la réception des patients, gestion de la facturation, gestion de l'hospitalisation et la gestion de la pharmacie.

Après avoir effectué un état de lieux du logiciel existant, en accord avec l'administration et techniciens, la conception d'un nouveau logiciel s'impose, pour gérer ces fonctionnaires d'état.

Objectif

L'objectif de ce logiciel est de numériser les données, d'automatiser les activités au sein de la Polyclinique.

Travaux demandés

Cas d'utilisation :

Pour l'administration de la polyclinique :

- Elaboration de la statistique des patients (Nombre des patients par date, par service...)
- Facturation des patients (Payement, paiement du reste à payer, annulation, modification,...)
- Gestion des prestations diverses
- Bilan régulier

Pour la réception :

- Enregistrement des patients
- Triage des patients
- Enregistrement des sorties des patients
- Facturation
- Saisie des données médicales des patients

Pour l'administration du logiciel :

- Gestion des utilisateurs
- Gestion des fonctionnalités du logiciel

Encadreur(s)

- RAKOTOARISOA Jean Claude, Dr Ingénieur
- RAMANAN'HAJA Hery Tina, Ingénieur
- Dr Luigi BELLINI, Professeur
- RAZAIARIMALALA Nirina Claudia, Chef de service Administration

0.1 Lieu de travail

- Laboratoire d'Informatique Appliquée et de Mathématique, UNA
- Polyclinique Universitaire Next

0.2 Etudiant réalisateur

HAJALALAINA Fara Marie José

Remerciements

En préambule de ce travail, j'aimerais exprimer, par ces quelques lignes de remerciements, mes gratitude envers tous ceux en qui, par leur présence, leur soutien, leur disponibilité et leurs conseils j'ai trouvé courage pour réaliser ce travail.

En premier lieu, j'adresse mes respectueuses considérations à mon encadreur professionnel : Dr. RAKOTOARISOA Jean Claude, Monsieur RAMANAN'HAJA Hery Tina qui ont été l'esprit pensant donnant naissance à ce sujet et qui m'a dirigée par ses précieux conseils tout en me laissant la liberté d'initiative pour la conception de la plateforme. A mes encadreurs qui sont Dr Luigi BELLINI, Madame RAZAIARIMALALA Nirina Claudia pour avoir dirigés ce travail, pour leurs soucient et aussi pour leurs aides et leurs orientations qui m'ont permis de réaliser ce travail dans les meilleures conditions.

J'adresse mes remerciements anticipés et mes honorables considérations également aux membres de Jury qui vont évaluer et porter leur jugement à ce travail, aussi en enrichir le contenu par leurs précieuses propositions.

Je dois reconnaissance également à l'ensemble du corps enseignant de la mention STIC de le ESPA (Ecole Supérieure Polytechnique d'Antsiranana).

Une mention particulière à ma famille, pour leurs soutiens et leurs attentions sans faille, dont les encouragements et l'amour.

Bref, tout ceux qui ont contribué de près ou de loin à l'accomplissement de ce travail.

HAJALALAINA Fara Marie José

Table des matières

Remerciements	i
Table des matières	iii
Liste des figures	iv
Liste des acronymes	iv
Glossaire	iv
Introduction générale	1
Chapitre 1 GENERALITE	2
1.1 Introduction	3
1.2 Bref Historique du polyclinique Universitair NEXT	3
1.3 Localisation et information sur le complex hospitalier	3
1.4 Généralité sur le web	4
1.5 Choix de la technologie utilisé	4
Chapitre 2 Modélisation du système	5
2.1 Introduction	6
2.2 Quelque notion sur la modélisation UML	6
2.3 Spécification de besoin	7
2.4 Diagramme de cas d'utilisation	9
2.5 diagramme de séquence	13
2.6 Diagramme de classe de l'application	15
Chapitre 3 Réalisation et Déploiement l'application	17
3.1 Introduction	18
3.2 Les Systèmes de Gestion de Base de Données (SGBD)	18
3.3 PHP avec le framwork Laravel	18
3.4 Création de squelete de projet laravel	20
3.5 Déploiement de l'application sur Apache	21
3.6 Présentation de l'application	23
Conclusion générale	26
Bibliographie	I
Webographie	II
Bibliographie	I

Liste des figures

2.1	Diagramme de cas d'utilisation générale du système	9
2.2	Diagramme de cas d'utilisation réceptionniste	10
2.3	Diagramme de cas d'utilisation administrateur polyclinique	11
2.4	Diagramme de cas d'utilisation administrateur logiciel	12
2.5	Diagramme de séquence authentification	13
2.6	Diagramme de séquence gestion patient	14
2.7	Diagramme de séquence gestion facturation	15
2.8	Diagramme de classe de l'application	16
3.1	Fenêtre d'authentification	24
3.2	Tableau de bord	25

Introduction générale

Polyclinique Next dispose de différent branche qui offre de variété de service. que ce soit patient, personnel ou d'autre entité toute cette personnage engendre de grande quantité d'information. Qui nécessite d'être traité et organisé . la présence de grand quantité d'information qui s'échange constamment au sein de chaque service s'avère très difficile à manoeuvrer sans la présence d'un logiciel spécifique pour automatiser certaine fonctionnalité. Grâce au partenariat avec la polyclinique Next et l'école supérieur polytechnique, moi, les encadreur,et les gent du polyclinique ont mis d'accord pour oeuvrer à la conception et réalisation d'un logiciel de gestion qui a pur but d'automatiser la fonctionnalité précédemment évoque dans le cahier de charge suivante.

Chapitre 1

GENERALITE

Sommaire

1.1 Introduction	3
1.2 Bref Historique du polyclinique Universitair NEXT	3
1.3 Localisation et information sur le complexe hospitalier	3
1.3.1 localisation	3
1.3.2 Information concernant le complexe hospitalier	4
1.3.3 Plan hospitalier	4
1.4 Généralité sur le web	4
1.5 Choix de la technologie utilisé	4

1.1 Introduction

1.2 Bref Historique du polyclinique Universitair NEXT

La NEXT onlus a été fondée en 1998 par un chercheur scientifique, Luigi BELLINI, qui, à la suite d'un voyage à Madagascar, profondément touché par les terribles conditions de vie de la majorité (80%) des personnes vivant avec moins d'un dollar par jour, a décidé d'entreprendre une action pour aider ces gens, en consacrant toutes ses énergies et ressources à ce but.

Le travail d'organisation et d'assistance s'est progressivement développé avec une particulière attention dans le domaine de la santé dans le Nord de Madagascar, où ont été accomplies des oeuvres importantes qui interagissent avec le système actuel de la Santé Publique

L'activité de la NEXT, développée au début avec ses propres moyens financiers et sans le soutien d'organismes extérieurs, a été reconnue par l'État Italien, qui, en Octobre 2006, lui a accordé le statut juridique d'ONG (Organisation Non Gouvernementale).

Le professeur Luigi BELLINI a progressivement abandonné ses activités en Italie, ses travaux de recherche à l'Université et professionnel, même sa collaboration scientifique avec l'Agence spatiale européenne. Il a décidé de réaliser un centre de diagnostic médical à Madagascar, dénommé « Le Samaritain ». Le centre, d'une superficie de 3 000 m², comprenant des laboratoires de radiologie et d'échographie, a débuté ses activités, en 2006. Il est doté d'équipements modernes de niveau européen.

En constatant que le centre de diagnostic à lui seul ne suffit pas, l'ONG a mis sur les rails la première clinique de maternité et de chirurgie dans le Nord. Une grande réalisation, unique à Madagascar. Les travaux de construction ont débuté en 2009 : ce fut les premières fondations de la Polyclinique universitaire NEXT.

1.3 Localisation et information sur le complexe hospitalier

1.3.1 localisation

Le Centre "Le Samaritain" ainsi que la Clinique de Maternité et Chirurgie se trouvent dans la région DIANA, située au Nord de Madagascar. (Pour arriver au complexe sanitaire il faut

s'engager sur la route nationale 6 (RN6) et procéder 200 m avant de tourner à gauche vers « Rue de la Fraternité ».)

1.3.2 Information concernant le complexe hospitalier

L'objectif de la NEXT était d'assurer l'assistance médicale aux gens de la Région d'Antsiranana et de toutes les Provinces du Nord de Madagascar par la construction et l'équipement d'une Clinique, située à côté du Centre de Diagnostic Médical "Le Samaritain". Cette nouvelle structure à trois étages abritera les départements de maternité, chirurgie générale et médecine et sera équipée de :

- une salle opératoire principale
- une salle opératoire d'urgence ou secondaire
- une salle d'accouchement
- une salle de stérilisation
- une salle postopératoire
- un centre d'hémodialyse
- un service d'urgences.

1.3.3 Plan hospitalier

L'établissement de santé est reparti comme suit :

REZ-DE-CHAUSSÉE	1ER ETAGE :	2EME ETAGE
Triage-urgence Hemodialyse Direction-administration	Bloc opératoire Chirurgie Gynéco-obstétrique Maternité Salle d'accouchement	Médecine Pharmacie Salle de classe Présidence

1.4 Généralité sur le web

1.5 Choix de la technologie utilisé

Chapitre 2

Modélisation du système

Sommaire

2.1	Introduction	6
2.2	Quelque notion sur la modélisation UML	6
2.3	Spécification de besoin	7
2.3.1	Besoin de l'accueil	7
2.3.2	départements d'admission	7
2.3.3	L'administration	8
2.3.4	La caisse	8
2.3.5	La direction	8
2.3.6	La médecine	8
2.4	Diagramme de cas d'utilisation	9
2.4.1	Diagramme de cas d'utilisation générale du système	9
2.4.2	Diagramme de cas d'utilisation réceptionniste	10
2.4.3	Diagramme de cas d'utilisation administrateur polyclinique	11
2.4.4	Diagramme de cas d'utilisation admin logiciel	12
2.5	diagramme de séquence	13
2.5.1	Diagramme de séquence authentification	13
2.5.2	Diagramme de séquence gestion patient	14
2.5.3	Diagramme de séquence gestion facturation	15
2.6	Diagramme de classe de l'application	15

2.1 Introduction

Tous les programmes informatiques n'ont pas été construits à l'aveuglette par les programmeurs, même les plus petits ont fait l'objet de quelques réflexions. Pour notre cas, quelques réflexions ne suffisent pas, il en faut beaucoup. Mais on peut facilement se perdre dans ces réflexions si on n'a pas de méthodes et un langage de modélisation bien adaptée. Nous construisons des modèles de systèmes complexes parce nous sommes incapables d'appréhender ces systèmes dans leur entièreté. Mais c'est quoi vraiment un modèle? Un modèle est une simplification de la réalité. la modélisation permet alors de maîtriser la complexité du système étudié, car chaque modèle donne accès à une représentation abstraite de différents aspects du système. modéliser un système consiste à créer une représentation schématique simplifié d'un problème. Le but est de :

- Nous aider à visualiser le système tel qu'il est ou tel qu'il devrait être.
- Spécifier la structure et le comportement d'un système
- Avoir un "patron" pour guider la construction du système.
- Documenter les décisions qui ont été prises

2.2 Quelques notions sur la modélisation UML

2.3 Spécification de besoin

Pour se lancer dans la modélisation, on a besoin des information concernant cette établissement sanitaire pour comprendre le mécanisme qui y se trouve. De façon générale, le fonctionnement de l'hôpital repose sur l'ordonnancement des branches ci dessous.

2.3.1 Besoin de l'accueil

Un nouvel arrivant dans le centre hospitalier doit impérativement se présenter à l'accueil. De là, le responsable a besoin d'enregistrer les renseignements de base du nouveau malade, à savoir : La date d'entrée, le nom Prénom, le sexe, l'age, l'adresse, le téléphone, le médecin traitant, la famille ou personne externe, la profession et l'unité d'admission. Sur l'unité d'admission, ceci est à propos du département dans lequel le patient va être orienté Il est à noter que les renseignements sus enregistrés, feront l'objet d'une fiche qui sera remplie au fur et à mesure du traitement du patient concerné, une fois admis dans l'unité d'admission

Quand un malade doit être hospitalisé, c'est également à l'accueil que cet évènement doit être pris en compte. En effet, le responsable enregistre :

Le numéro du patient (entre autre, son dossier médical), la chambre le lit et la catégorie de la chambre

Comme le rôle de l'accueil se repose surtout sur l'enregistrement du mouvement de la population hospitalière, à la sortie d'un patient, un personnel de l'accueil enregistre la sortie, avec les attributs suivants : le numéro de patient le nom et prénom, l'état de sortie (mort, guérison, amélioration, statu-quo), la décision de sortie (de l'hôpital ou volontaire)

2.3.2 départements d'admission

A La polyclinique Universitaire NEXT, on retrouve les départements suivants :

la maternité, chirurgie,bloc, opératoire, néphrologie, pédiatrie, gynécologie, mise en observation.

Une fois que le patient est orienté vers son unité d'admission, la département a besoin de modifier la fiche médical du malade, en rapport avec son suivi médical.

2.3.3 L'administration

Toutes les prestations effectuées par le patient sont notées et enregistrées par un responsable de administration.L'administration a besoin d'enregistrer toutes les prestations.

2.3.4 La caisse

La caisse est chargée du service facturation. Quand un patient va passer au paiement, il se présente à la caisse. Le caissier consulte les prestations saisies par l'administration. Son rôle vise également à l'évaluation et l'exploitation d'un certain nombre d'informations et de statistiques, liées à la comptabilité des prestations.

2.3.5 La direction

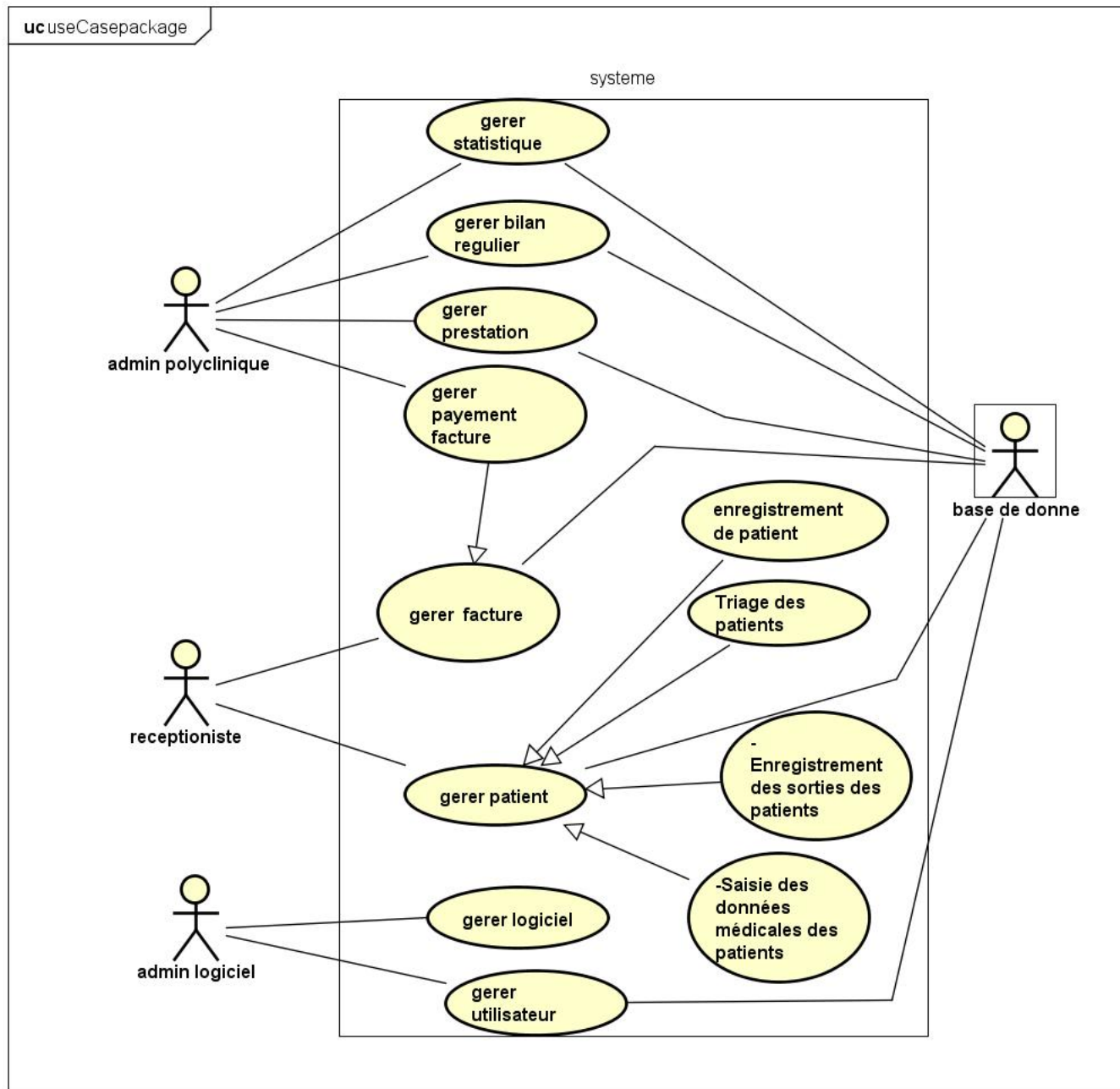
La direction a dans ses attributs, la gestion des utilisateurs, la diffusion d'informations/messages au personnel.

2.3.6 La médecine

La section médecine est dédiée aux médecins, plutôt dans le sens de prescription. Quand un patient se présente alors au cabinet, le médecin lui prescrit une ordonnance

2.4 Diagramme de cas d'utilisation

2.4.1 Diagramme de cas d'utilisation générale du système



powered by Astah

FIGURE 2.1 – Diagramme de cas d'utilisation générale du système

2.4.2 Diagramme de cas d'utilisation réceptionniste

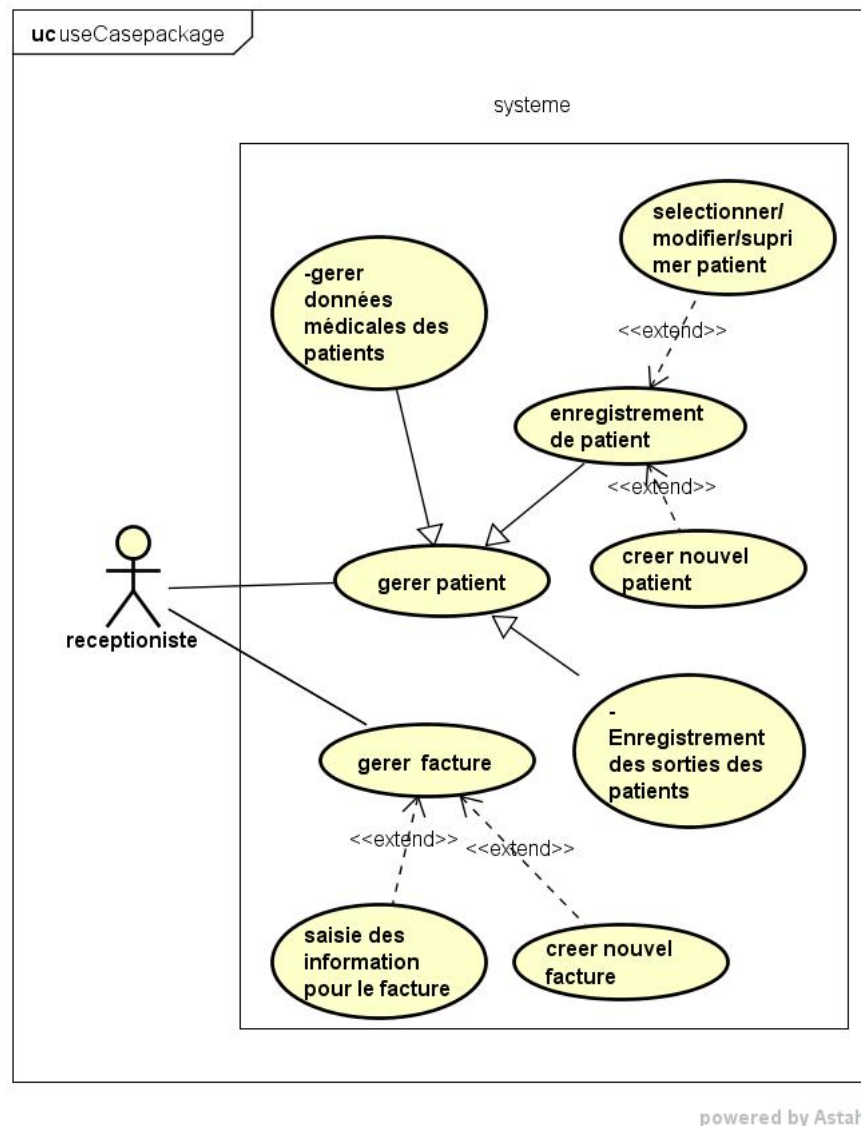
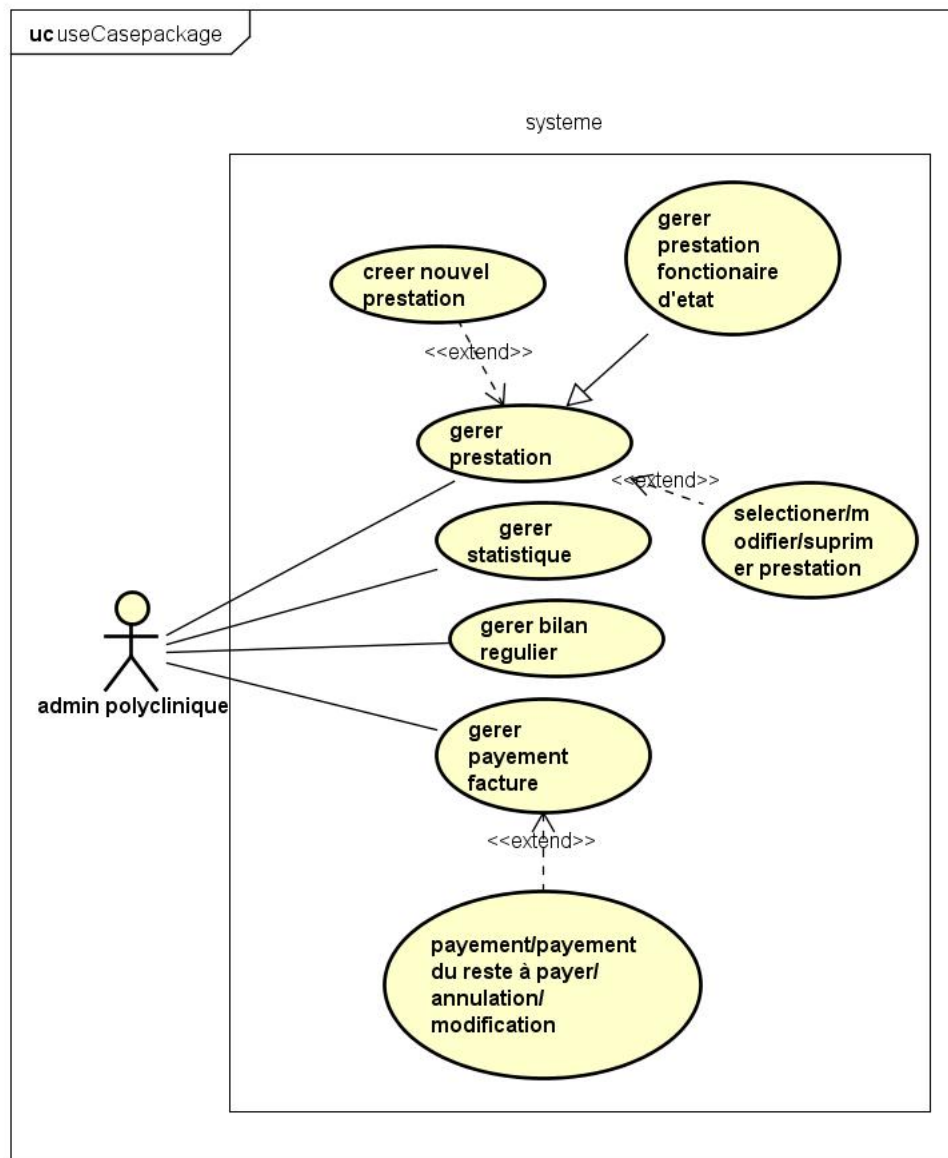


FIGURE 2.2 – Diagramme de cas d'utilisation réceptionniste

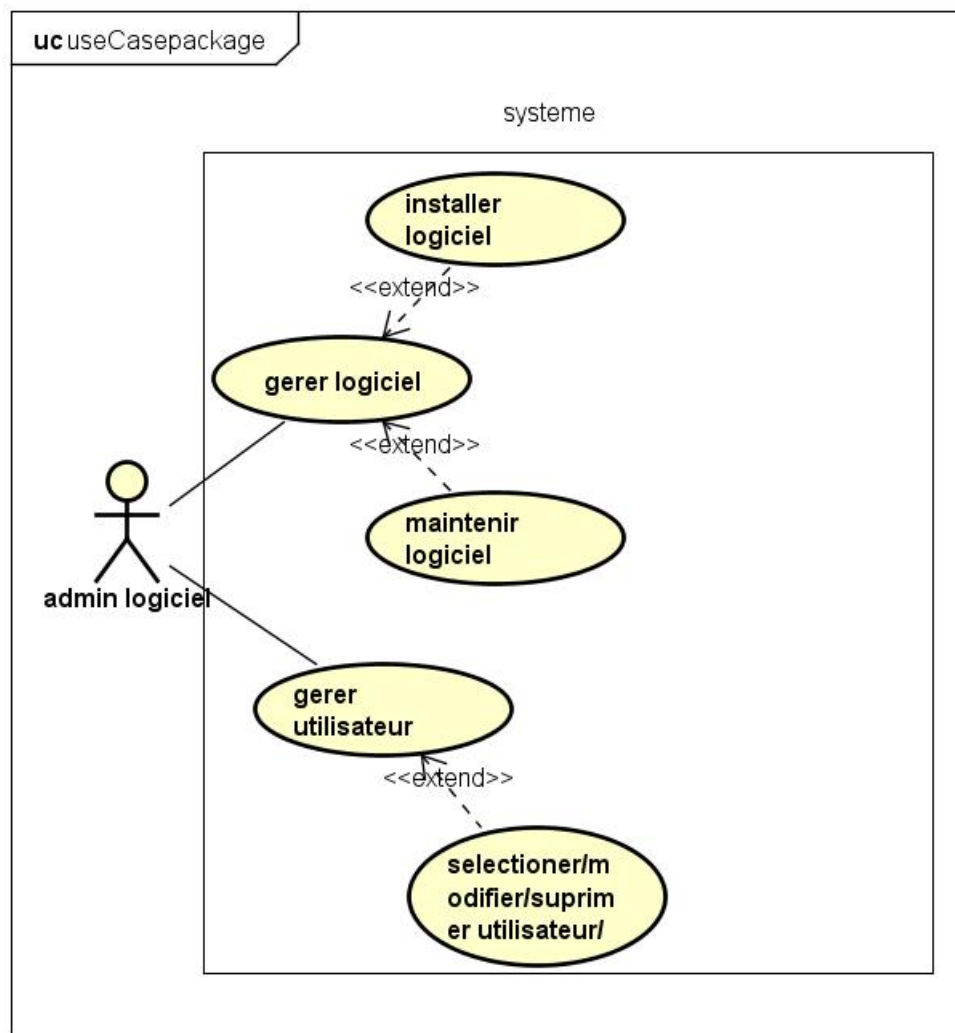
2.4.3 Diagramme de cas d'utilisation administrateur polyclinique



powered by Astah

FIGURE 2.3 – Diagramme de cas d'utilisation administrateur polyclinique

2.4.4 Diagramme de cas d'utilisation admin logiciel

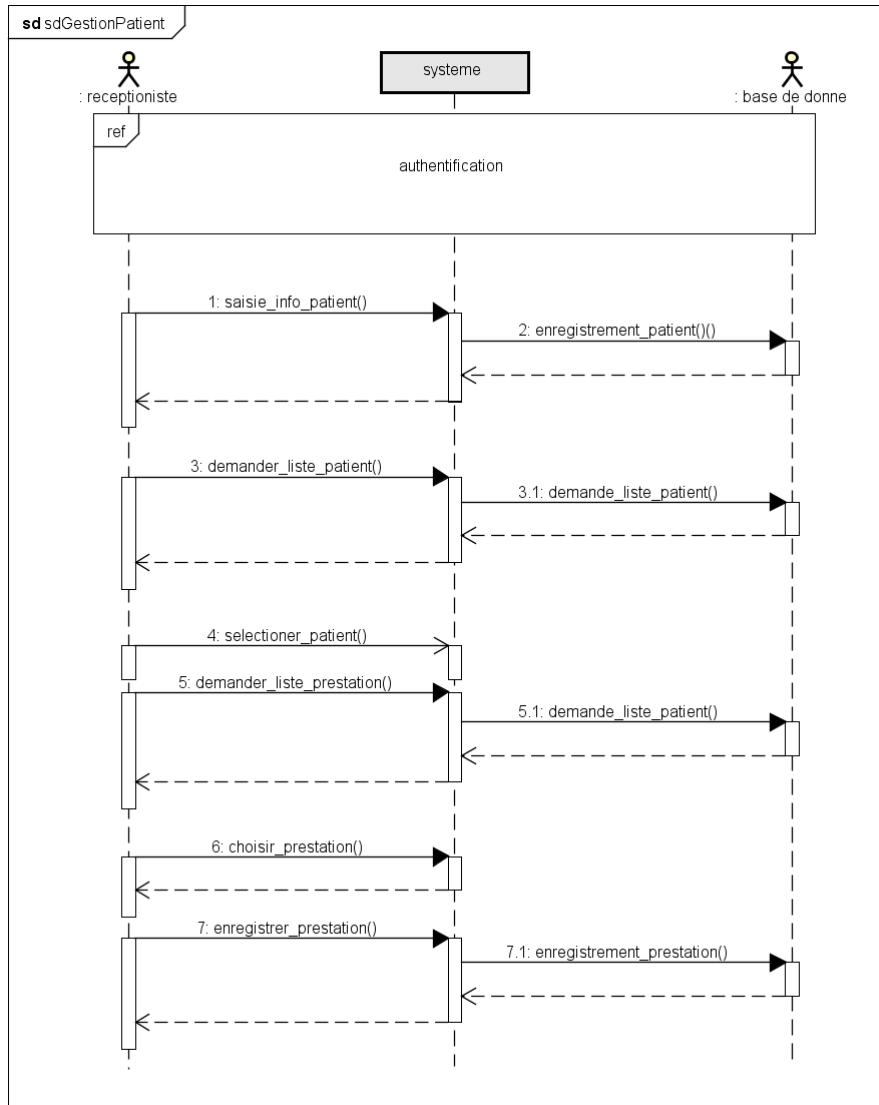


powered by Astah

FIGURE 2.4 – Diagramme de cas d'utilisation administrateur logiciel

2.5 diagramme de séquence

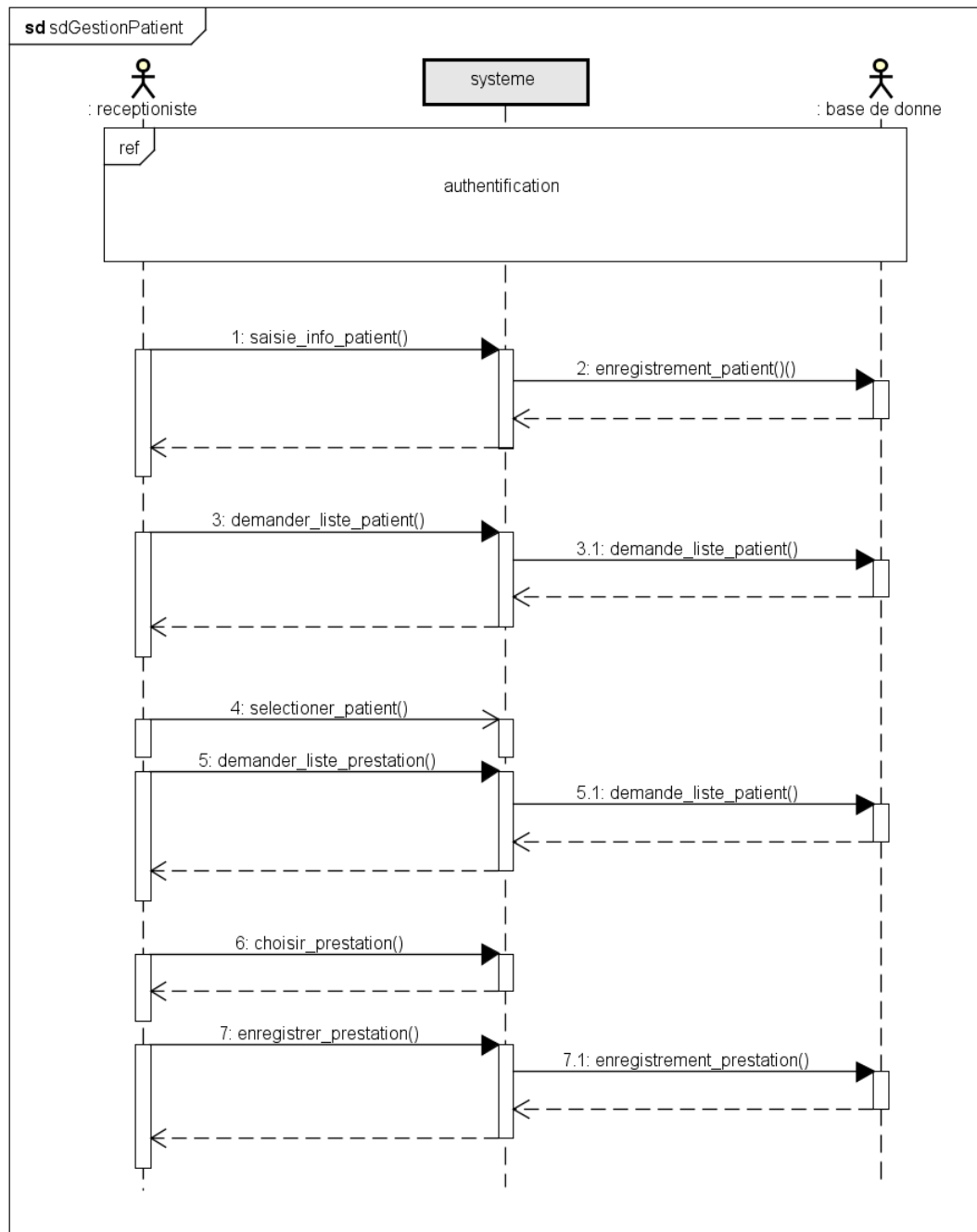
2.5.1 Diagramme de séquence authentification



powered by Astah

FIGURE 2.5 – Diagramme de séquence authentification

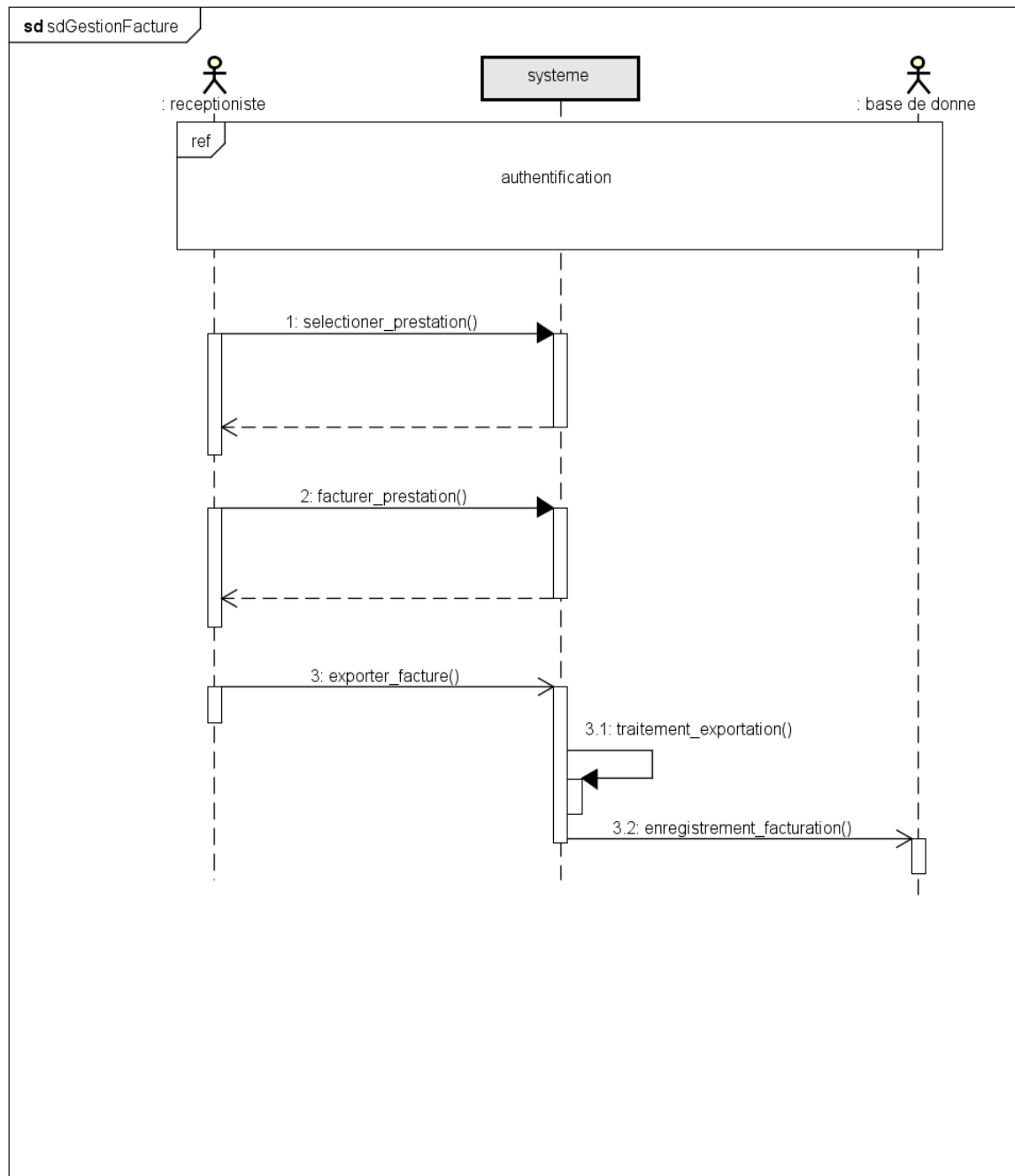
2.5.2 Diagramme de séquence gestion patient



powered by Astah

FIGURE 2.6 – Diagramme de séquence gestion patient

2.5.3 Diagramme de séquence gestion facturation



powered by Astah

FIGURE 2.7 – Diagramme de séquence gestion facturation

2.6 Diagramme de classe de l'application

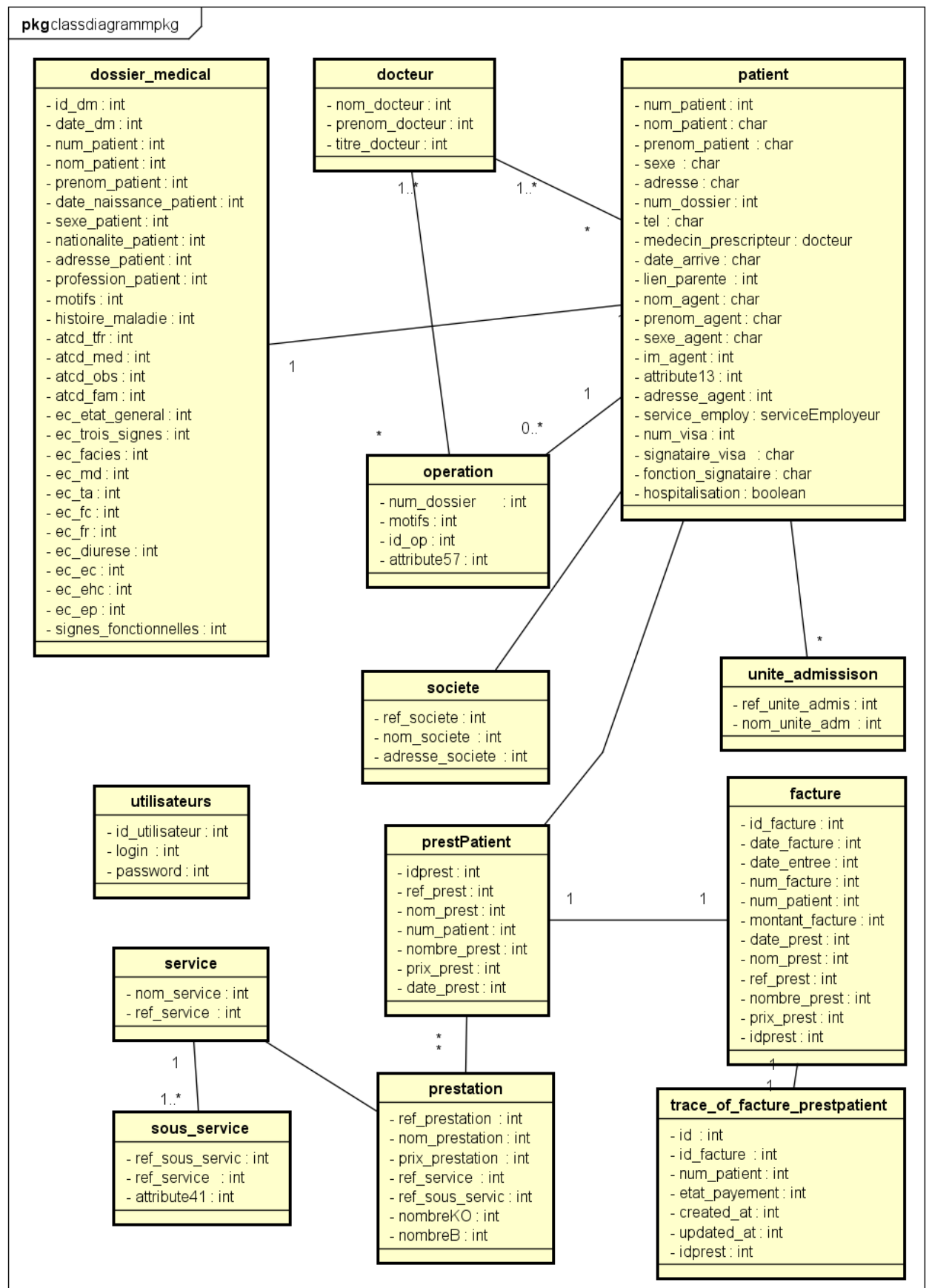


FIGURE 2.8 – Diagramme de classe de l'application

Chapitre 3

Réalisation et Déploiement l'application

Sommaire

3.1 Introduction	18
3.2 Les Systèmes de Gestion de Base de Données (SGBD)	18
3.2.1 Définition	18
3.2.2 Systèmes de Gestion de Base de Données MySQL	18
3.3 PHP avec le framework Laravel	18
3.3.1 Pourquoi le choix de laravel?	18
3.3.2 Moteur de modèle de lame	19
3.3.3 Système de routage très facile à manœuvrer	19
3.3.4 Eloquent ORM	19
3.3.5 Artisan	19
3.4 Création de squelette de projet laravel	20
3.4.1 Création du projet	20
3.4.2 La structure des dossiers	20
3.5 Déploiement de l'application sur Apache	21
3.5.1 Configurer apache	21
3.5.2 Configurer l'application Laravel	22
3.6 Présentation de l'application	23
3.6.1 Fenêtre d'authentification	23
3.6.2 Le tableau de bord	25

3.1 Introduction

Après avoir effectué la modélisation, on peut désormais entamer la programmation et la réalisation du système. Dans cette partie, on abordera en premier lieu la description de notre Système de Gestion de Base de Données (SGBD) ainsi que les langages de programmation utilisés et par la suite on présentera les démarches de conception pour l'implémentation du système.

3.2 Les Systèmes de Gestion de Base de Données (SGBD)

3.2.1 Définition

Les SGBD sont des logiciels système destinés à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations. En bref c'est un logiciel intermédiaires entre les utilisateurs et les bases de données. Une base de données est un magasin de données composé de plusieurs fichiers manipulés exclusivement par le SGBD. Ce dernier cache la complexité de manipulation des structures de la base de données en mettant à disposition une vue synthétique du contenu.

3.2.2 Systèmes de Gestion de Base de Donne MYSQL

Pourquoi le choix de MYSQL ?

MYSQL est le SGBD qui est familiarisé par les informaticien du Polyclinique, le serveur qui y se trouve dispose déjà les ressource nécessaire pour démarrer cette SGBD. Nous allons utilisé MYSQL car dans le cas contraire, les équipe qui va travailler au futur devra s'habituer à cette autre environnement, et cela va couté du temps.

3.3 PHP avec le framwork Laravel

3.3.1 Pourquoi le choix de laravel ?

Laravel est le framwork les plus suggerére par le communauté du PHP aujourd'hui. Laravel est basé sur symfony mais on y introduit des différent amélioration. Laravel aussi rassemble des divers avantage des autre framwork et y intégré. laravel offre un avantage de taille, celui de separer la vue, le model et le Controller. l'avantage majeur est d'éviter le mélange de HTML avec du PHP.

Il y a donc trois parties du cadre qui fonctionnent ensemble : les modèles, les vues et les contrôleurs. Les contrôleurs sont la partie principale de la majeure partie du travail. Ils se connectent aux modèles pour obtenir, créer ou mettre à jour des données et afficher les résultats sur les vues, qui contiennent la structure HTML de l'application.

3.3.2 Moteur de modèle de lame

Laravel est livré avec un moteur de template appelé Blade. L'une des caractéristiques que le moteur de template Blade ne partage pas avec les autres modèles populaires est sa permissivité ; permettant l'utilisation de code PHP simple dans les fichiers de moteur de template Blade. Il est important de noter que les fichiers du moteur de .blade Blade ont .blade ajoutés aux noms de fichiers juste avant l'habituel fichier .php qui n'est rien d'autre que l'extension du fichier. En tant que tel, .blade.php est l'extension de fichier résultante pour les fichiers de modèles Blade. Les fichiers du moteur de modèle de lame sont stockés dans le répertoire ressources /views.

3.3.3 Système de routage très facile à manœuvrer

Vous pouvez définir les URL de votre application à l'aide d'itinéraires. Ces itinéraires peuvent contenir des données variables, se connecter à des contrôleurs ou être intégrés dans des middlewares. Middleware est un mécanisme de filtrage des requêtes HTTP. Ils peuvent être utilisés pour interagir avec les requêtes avant qu'elles n'atteignent les contrôleurs et peuvent ainsi modifier ou rejeter les demandes.

3.3.4 Eloquent ORM

L'Eloquent est un ORM (Object Relational Model) inclus avec le Laravel. Il implémente le modèle d'enregistrement actif et est utilisé pour interagir avec des bases de données relationnelles. Pour connecter vos modèles à différents types de bases de données, Laravel propose son propre ORM Eloquent avec un large éventail de fonctions avec lesquelles travailler. Le framework fournit également la migration et l'amorçage et propose également des restaurations.

3.3.5 Artisan

Artisan est l'outil en ligne de commande que vous pouvez utiliser pour contrôler certaines parties de Laravel. Des nombreuses commandes sont disponibles pour créer des modèles,

des contrôleurs et d'autres ressources nécessaires au développement.

3.4 Création de squelette de projet laravel

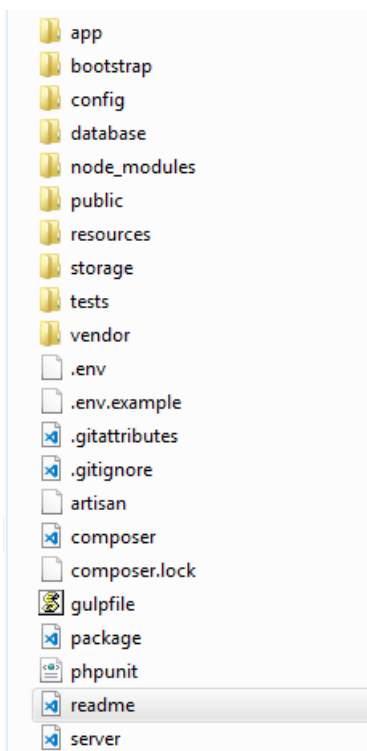
3.4.1 Création du projet

Nous devons avant tout installé Composer. Ensuite pour installer Laravel, il suffit de taper la commande suivante dans votre bash :

```
composer create-project --prefer-dist laravel/laravel projectname
```

3.4.2 La structure des dossiers

Après l'installation et la création de votre projet avec composer, nous allons voir la structure des dossiers et fichiers.



Nous allons quasiment passer tout notre temps dans le dossier *app,resources, route*. Ce dossier contient presque tous les fichiers dont nous avons besoin pour coder notre application. Nous pouvons aussi installer des dépendances avec le gestionnaire de package de php : "Composer". Ces dépendances seront installées dans le dossier "vandong".

3.5 Déploiement de l'application sur Apache

Après avoir développé une application il faut pouvoir la déployer sur un serveur pour la rendre disponible par les utilisateurs. On doit installer l'environnement suivant au préalable : Apache2, mysql, et PHP7.

3.5.1 Configurer apache

Rendez vous dans le dossier `/etc/apache2` on crée un fichier nommé `next.dev` avec d'extension `.conf` dans le dossier `site-availables` et on ajoute le code suivant :

```
<VirtualHost *:80>
DocumentRoot /var/www/next.dev/public_html
<Directory "/var/www/next.dev/public_html">
Options +FollowSymLinks
AllowOverride all
Require all granted
</Directory>
<Directory "/var/www/next.dev/laravel">
Options +FollowSymLinks
AllowOverride all
Require all granted
</Directory>
ServerName next.dev
ServerAlias www.next.dev
ErrorLog /var/log/apache2/error.public.com.log
CustomLog /var/log/apache2/access.public.com.log combined
</VirtualHost>
```

et puis exécutez le bash suivant :

```
a2ensite next.dev.conf
service apache2 reload
```

rendez vous dans le dossier `/etc` et insérez dans le fichier nommé `hosts` la ligne suivante :

```
127.0.0.1      next.dev
```

3.5.2 Configurer l'application Laravel

Par défaut, le dossier public du projet Laravel expose le contenu de l'application qui peut être demandé de n'importe où par n'importe qui, le reste du code de l'application est invisible ou inaccessible à quiconque sans les autorisations appropriées.

Après avoir développé l'application sur votre machine de développement, celle-ci doit être transmise à un serveur de production pour pouvoir y accéder

- Étape 1 Créez un dossier appelé laravel (ou tout ce que vous voulez) au même niveau que le dossier

```
public_html
```

Exemple :

```
/
|--var
|---www
|----laravel
|----public_html
```

- Étape 2 Copiez tout sauf le dossier public de votre projet laravel (sur la machine de développement) dans le dossier laravel (sur l'hôte du serveur)
- Étape 3 Ouvrez le dossier public de votre projet laravel (sur la machine de développement), copiez tout et collez-le dans le dossier
public_html (sur l'hôte du serveur)
- Étape 4 Ouvrez maintenant le fichier index.php dans le dossier

```
public_html
```

Changement :

```
require __DIR__.'/../bootstrap/autoload.php';
```

À :

```
require __DIR__.'/../laravel/bootstrap/autoload.php';
```

Et changer :

```
$app = require_once __DIR__.'/../bootstrap/app.php';
```

À :

```
$app = require_once __DIR__.'/../laravel/bootstrap/app.php';
```

Sauver et fermer.

- Étape 5 Allez maintenant dans le dossier laravel (sur le laravel hébergement partagé) et ouvrez le fichier server.php Changement

```
require_once __DIR__.'/public/index.php';
```

À :

```
require_once __DIR__.'/../public_html/index.php';
```

- Étape 6

Après avoir suivi toutes les étapes ci-dessus, créez un fichier .env dans le dossier laravel et ajoutez toute la variable d'environnement que vous avez utilisée du fichier .env du serveur local / de développement au fichier .env du serveur de production.

- Étape 7 rendez vous sur l'url : <http://next.dev/index.php>

A la racine de notre application Laravel, on exécute la commande suivante : il faut copier le fichier laravel

```
php artisan key:generate
```

```
php artisan migrate
```

Le déploiement effectué, notre application est maintenant accessible via l'URL :

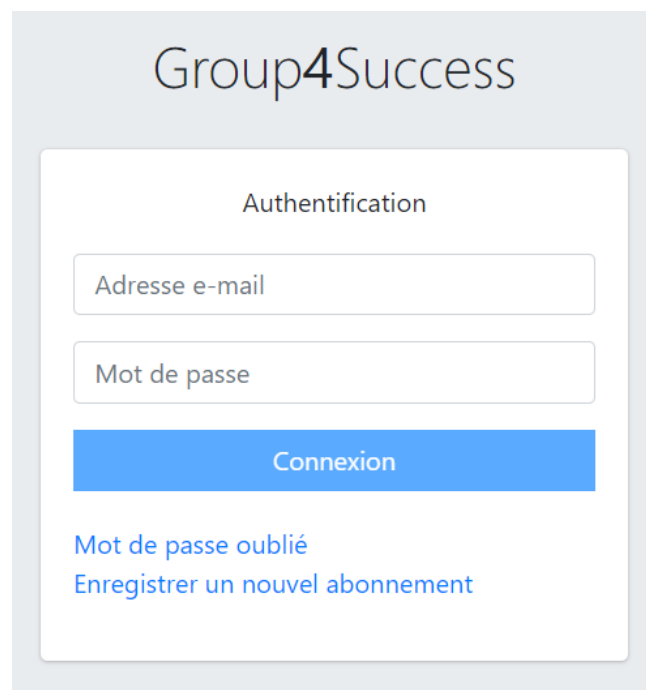
"next.dev/index.php"

3.6 Présentation de l'application

Nous allons faire une brève présentation de l'interface aussi bien que les fonctionnalités de l'application. Notons au tout début que c'est une application utilisé en offline qui a pour nomination : « NEXT app ». Cette application sera utilisé par l'administration et le secrétaire du polyclinique

3.6.1 Fenêtre d'authentification

Avant tout, l'utilisateur doit impérativement avoir un compte afin de logger sur l'application



The image shows a login window for 'Group4Success'. The window has a light gray background. At the top, the text 'Group4Success' is displayed in a large, dark gray font. Below this, the word 'Authentification' is centered in a smaller, dark gray font. There are two input fields: the first is labeled 'Adresse e-mail' and the second is labeled 'Mot de passe'. Both fields have a light gray border and a white background. Below the input fields is a blue button with the text 'Connexion' in white. At the bottom of the window, there are two links in blue text: 'Mot de passe oublié' and 'Enregistrer un nouvel abonnement'.

FIGURE 3.1 – Fenêtre d’authentification

3.6.2 Le tableau de bord



FIGURE 3.2 – Tableau de bord

Après authentification, l'utilisateur sera redirigé directement au tableau de bord

Conclusion générale

Notre projet intitulé Conception et réalisation du logiciel de gestion de la Polyclinique NEXT est destiné pour alléger les efforts de gestion de patient, la gestion de facturation, et d'autre mécanisme intéressant dans le Polyclinique.

Contrairement à la majorité des applications classique, notre application fondé à base de laravel permet aisement la continuité de développement en raison de sa modularité.

En ce qui concerne les étapes de notre travail, nous avons, en premier lieu, effectué une phase d'étude théorique pour déterminer les problématiques liés à la réalisation du projet pour ainsi fixer les solutions que nous devons adapter. En second lieu, nous avons défini les services fournis par le NEXT app en détaillant les solutions à faire pour satisfaire les attentes du polyclinique. En troisième lieu, nous avons modélisé l'application par rapport aux besoins fonctionnels et techniques de notre scénario de base. Après quoi, en quatrième partie, nous sommes passés à la réalisation de l'application en faisant soin de choisir l'outil idéal pour se faire. Dans la dernière étape, nous avons présenté l'application à travers des imprimés écrans et enfin le déploiement de celle-ci dans un serveur distant.

L'application ainsi développée fournit des services pour répondre aux besoins évolutifs de Polyclinique. Elle permet à l'administration, possédant des clés d'accès à l'application, de gérer, d'une part, les gestions des patients et des factures. Ces gestions peuvent être, l'enregistrement de nouveaux patients, demande de prestation, gestion de facturation, traitement de bilan régulière au sein de polyclinique. D'autre part, l'outil gère l'utilisateur, à savoir : la gestion des inscriptions. et la gestion de compte de l'utilisateur. l'utilisateur peut être secrétaire ou autre administrateur de polyclinique.

Malgré quelques difficultés à mettre en œuvre tous les modules, ce travail nous a présenté une réelle occasion d'apprendre et de faire un premier pas vers le monde du développement pour une société,

Bibliographie

- [1] Généralité sur la gestion des ressources humaines. *Robert* NIKIKO. IRIMAG - Licence en Administration et Management.
- [2] Deviens un ninja avec Angular. *Ninja Squad*. Learnangularjs, 2016.
- [3] Learn Angular 2 step by step. *Questpond*, 2016.
- [4] Cours de base de données. *R. Njakarison Menja*, 2013.
- [5] Les établissements d'enseignement supérieur : Structure et fonctionnement, 2017. Page 12-13. Brochures de PARFAIRE. Chapitre structure et fonctionnement. creativecommons.org/licenses/by-nc-nd/4.0/fr/.
- [6] Firebase, 2017. Gregory HOWARD - Alix DUCROS. *Build Extraordinary Apps*.
- [7] Angular 2 & 4 maîtriser le Framework Front-End de Google. ORSYS, La Grande Arche.
- [8] Firebase Realtime Database. *Landon COX*, 2017

Webographie

- [9] Études supérieures, 06 Mars 2016. Wikipédia. fr.wikipedia.org.
- [10] Réforme Licence-Master-Doctorat, Octobre 2017. Wikipédia. fr.wikipedia.org.
- [11] Gestion des ressources humaines, Septembre 2017. Wikipédia. fr.wikipedia.org.
- [12] Le cloud computing, 21 Mars 2016. Wikipédia. fr.wikipedia.org.
- [13] Le logiciel en tant que service, Juin 2016. Wikipédia. fr.wikipedia.org.
- [14] FCIC Média. Le mode SaaS, 2017. www.lecoindesentrepreneurs.fr/le-mode-saas/.
- [15] Secteur de l'éducation, 2017. Unit4. Erp. www.unit4.com/fr/secteurs/education.
- [16] Firebase, Novembre 2017. Wikipédia. fr.wikipedia.org/wiki/Firebase.
- [17] Les raisons clés de développer avec angular, 2016. Caroline PHILLIPS. www.pic.digital/blog/396-angularjs-pourquoi.
- [18] Initiation à la conception de bases de données avec merise, Février 2012. Idriss NEUMANN. ineumann.developpez.com/tutoriels/merise/initiationmerise/.
- [19] How to Use Local Storage and Session Storage In Angular 4, Janvier 2018. JsTutorials Team. www.js-tutorials.com/javascript-tutorial/use-localstorage-sessionstorage-using-webstorage-angular4/.
- [20] Angular Authentification Tutorial, 5 Décembre 2017. Ado KUKIC. auth0.com/blog/angular-2-authentication/
- [21] Firebase Guides, 8 Mai 2018. firebase.google.com/docs/guides/
- [22] Firebase Authentication, 8 Mai 2018. <https://firebase.google.com/docs/auth/>
- [23] One framework Mobile and Desktop, Angular. angular.io.
- [24] Bootstrap Documentation, v4.1. Core Team. <https://getbootstrap.com/docs/4.1/>
- [25] jQuery Tutorial, 2018. www.w3schools.com/Jquery/default.asp
- [26] SweetAlert, Hand-crafted, by Tristan EDWARDS. sweetalert.js.org/guides/