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	Examinateur
Monsieur	Examinateur

⋆ Année Universitaire 2018 - 2019 ⋆
Promotion :

MENTION STIC
Science et Technologie de l'Information et de la Communication

ÉCOLE SUPÉRIEURE POLYTECHNIQUE D'ANTSIRANANA
MENTION MASTER STIC
B.P. O 201 - ANTSIRANANA - MADAGASCAR
Tél. : +261 (0)32 76 395 40 – **Courriel :** mentionsticespa@gmail.com

« *Maîtriser aujourd'hui la technologie de demain* »

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 publique 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, payement 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'aimerai exprimer, par ces quelques lignes de remerciements, mes gratitudes 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 ii

Liste des figures iii

Liste des tableaux iv

Liste des acronymes v

Glossaire vi

Introduction générale 1

Chapitre 1 PRESENTATION DE LA POLYCLINIQUE UNIVERSITAIRE NEXT 2
1.1 Historique . . . 3
1.2 Localisation et information sur le complex hospitalier . . . 3
1.3 Plan hospitalier . . . 4

Chapitre 2 Modélisation statique et dynamique du système 5
2.1 Introduction . . . 6
2.2 Analyse de fonctionnement de l'hôpital . . . 7
2.3 Diagrame de cas d'utilisation . . . 9
2.4 diagramme de séquence . . . 13
2.5 Diagramme de classe de l'application . . . 14

Chapitre 3 Programmation et réalisation du logiciel Chapitre 4 Modélisation de la base de donne 19
4.1 Introduction . . . 20
4.2 Le framwork Laravel . . . 20

Chapitre 5 Programmation et réalisation du logiciel 22
5.1 Introduction . . . 23
5.2 Les Systèmes de Gestion de Base de Données (SGBD) . . . 23
5.3 Les langages de programmation . . . 24
5.4 Conception . . . 26

Liste des figures

2.1 Diagramme de cas d'utilisation générale du systéme 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 gestion patient 15
2.6 Diagramme de classe de l'application 16

Liste des tableaux

Liste des acronymes

api

bdd

crmcsscv

ecespa

godgrcgrh

html

jeejsfjsonjspjstl

lmd

mcdmldmvc

rh

saassgbdsisslasoaspa

ueuml

Glossaire

multiTenant

cloudComputinggamingOnDemandmiddleware

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

PRESENTATION DE LA POLYCLINIQUE UNIVERSITAIRE NEXT

Sommaire

1.1 Historique 3
1.2 Localisation et information sur le complex hospitalier 3
1.2.1 localisation 3
1.2.2 Information concernant le complexe hospirtalier 4
1.3 Plan hospitalier 4

1.1 Historique

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.

1.2 Localisation et information sur le complex hospitalier

1.2.1 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 ».)

1.2.2 Information concernant le complexe hospirtalier

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 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

Chapitre 2

Modélisation statique et dynamique du système

Sommaire

2.1 Introduction 6
2.2 Analyse de fonctionnement de l'hôpital 7
2.2.1 L'accueil 7
2.2.2 Les départements d'admission 7
2.2.3 L'administration 8
2.2.4 La caisse 8
2.2.5 La direction 8
2.2.6 La médecine 8
2.3 Diagrame de cas d'utilisation 9
2.3.1 Diagramme de cas d'utilisation générale du systéme 9
2.3.2 Diagramme de cas d'utilisation réceptionniste 10
2.3.3 Diagramme de cas d'utilisation administrateur polyclinique 11
2.3.4 Diagramme de cas d'utilisation admin logiciel 12
2.4 diagramme de séquence 13
2.4.1 Diagrame de sequence gestion patient 13
2.4.2 Diagramme de séquence gestion facturation 14
2.5 Diagramme de classe de l'application 14

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 maitriser 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 Analyse de fonctionnement de l'hôpital

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.

2.2.1 L'accueil

Un nouvel arrivant dans le centre hospitalier doit impérativement se présenter à l'accueil. De là, le responsable enregistre 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)

2.2.2 Les 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, sa fiche sera modifiée, en rapport avec son suivi médical.

2.2.3 L'administration

Toutes les prestations effectuées par le patient sont notées et enregistrées par un responsable de administration.

2.2.4 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.

2.2.5 La direction

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

2.2.6 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

2.3 Diagrame de cas d'utilisation

2.3.1 Diagramme de cas d'utilisation générale du systéme

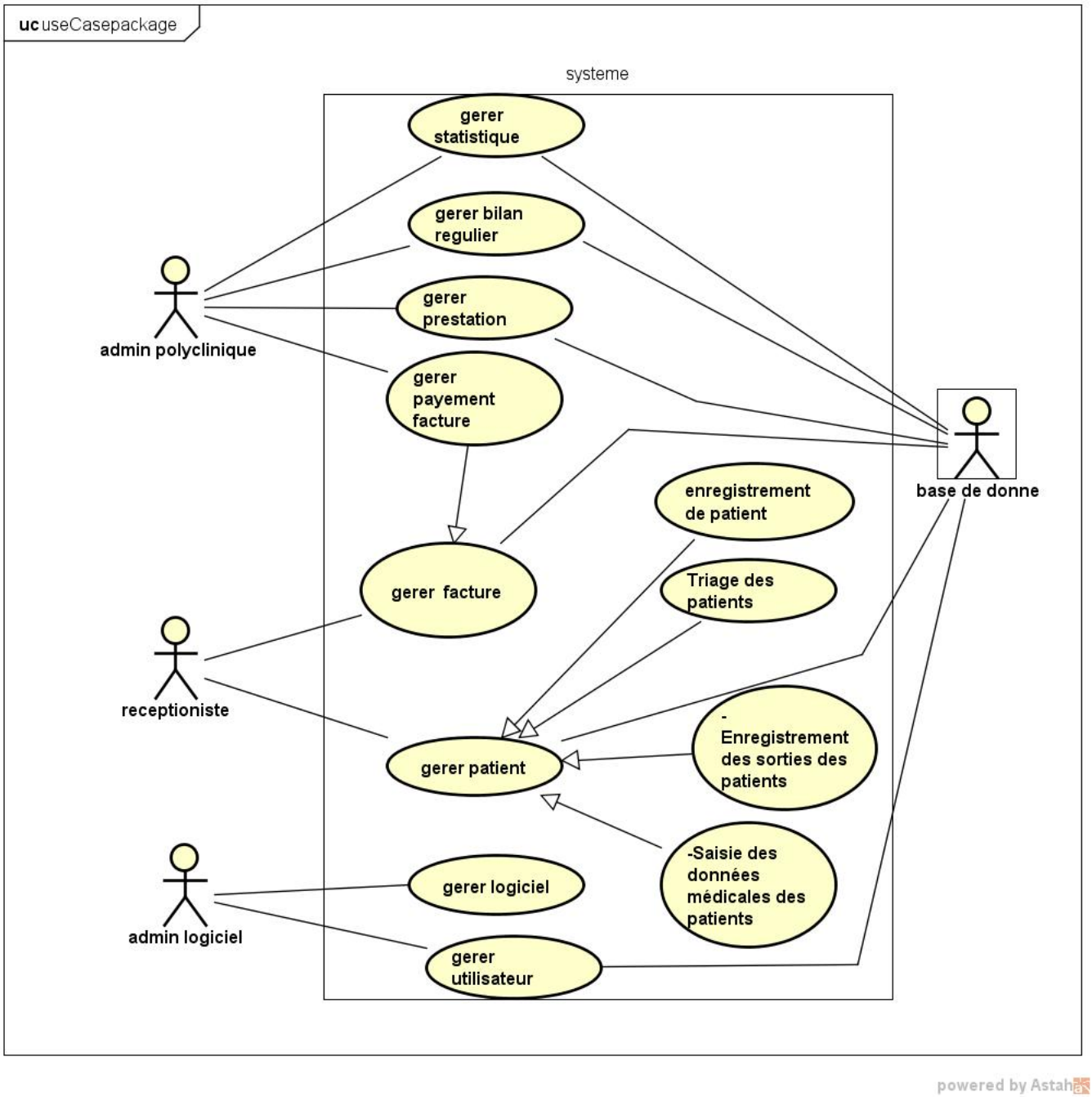

FIGURE 2.1 – Diagramme de cas d'utilisation générale du systéme

2.3.2 Diagramme de cas d'utilisation réceptionniste

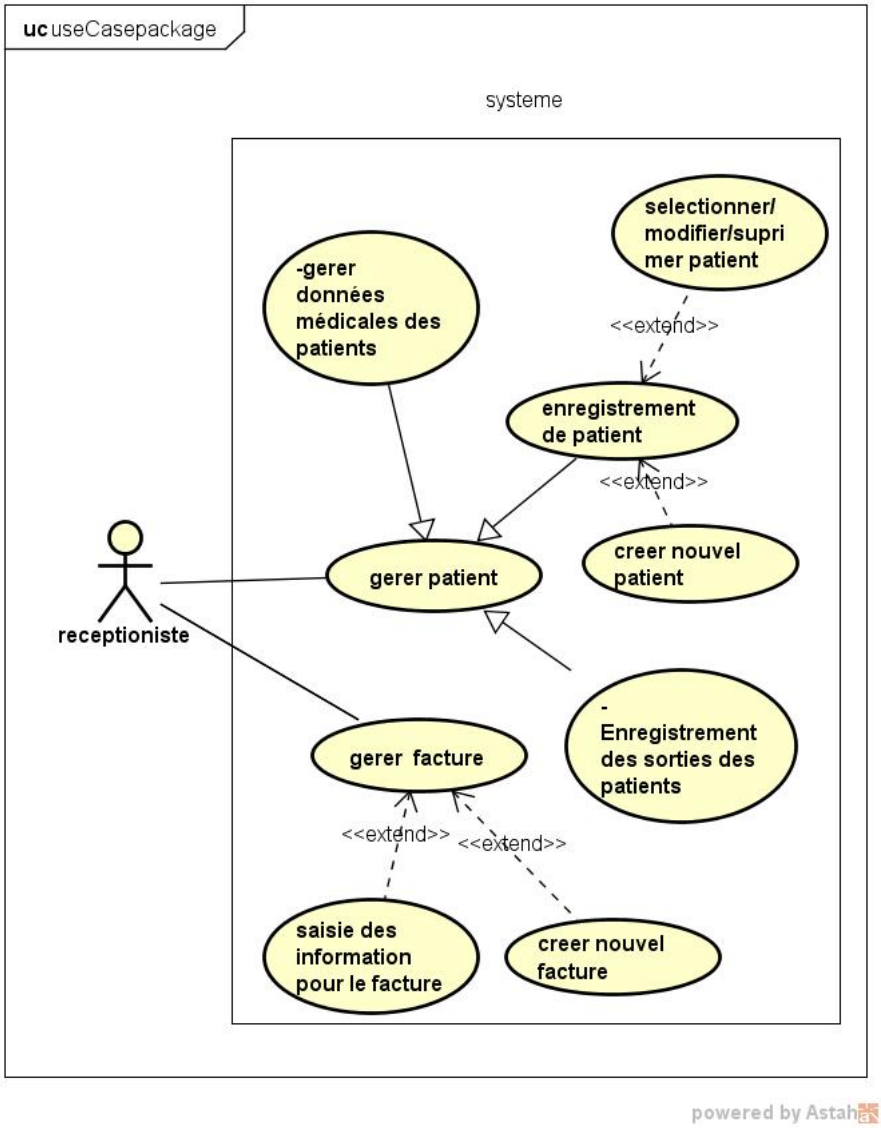

FIGURE 2.2 – Diagramme de cas d'utilisation réceptionniste

2.3.3 Diagramme de cas d'utilisation administrateur polyclinique

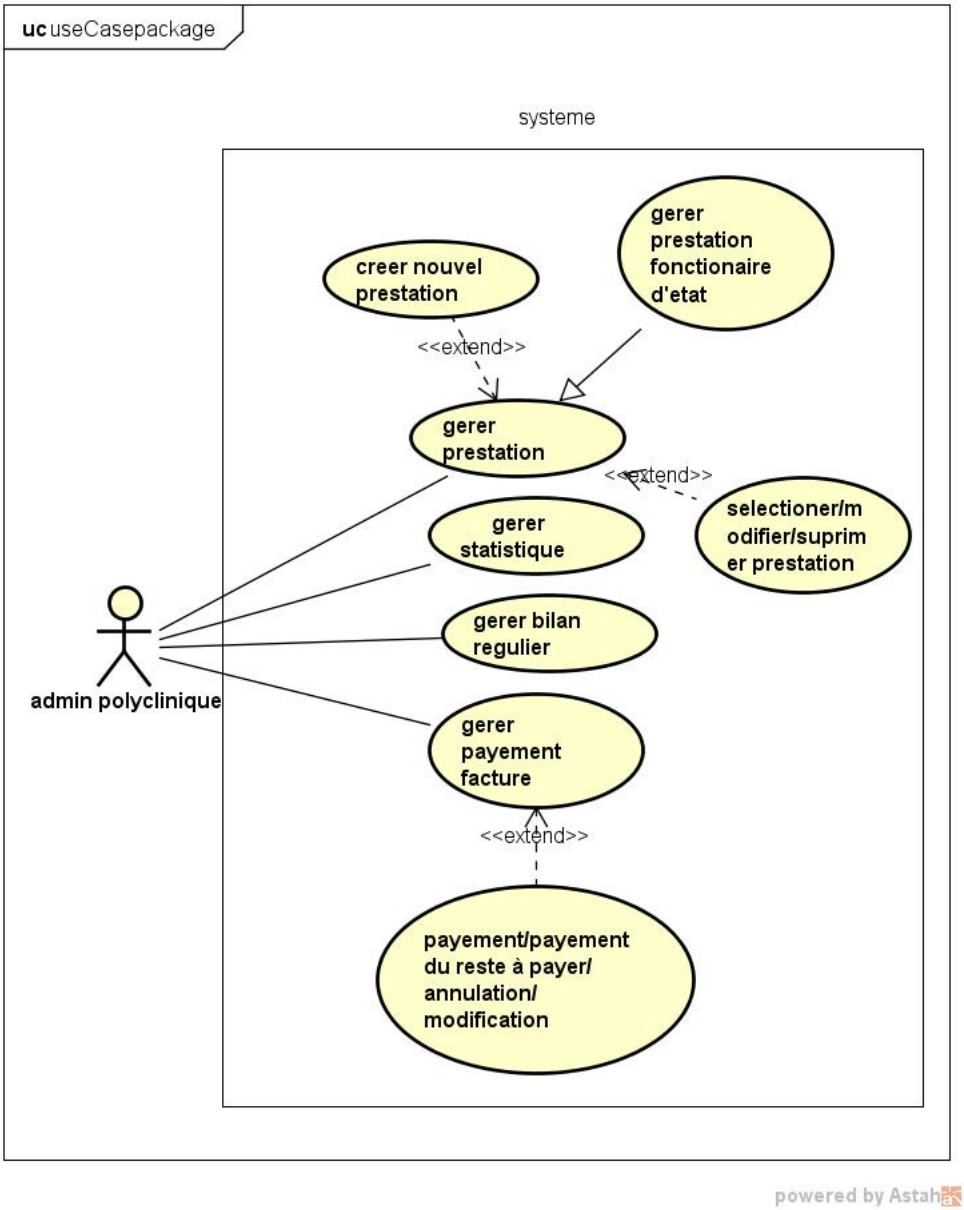

FIGURE 2.3 – Diagramme de cas d'utilisation administrateur polyclinique

2.3.4 Diagramme de cas d'utilisation admin logiciel

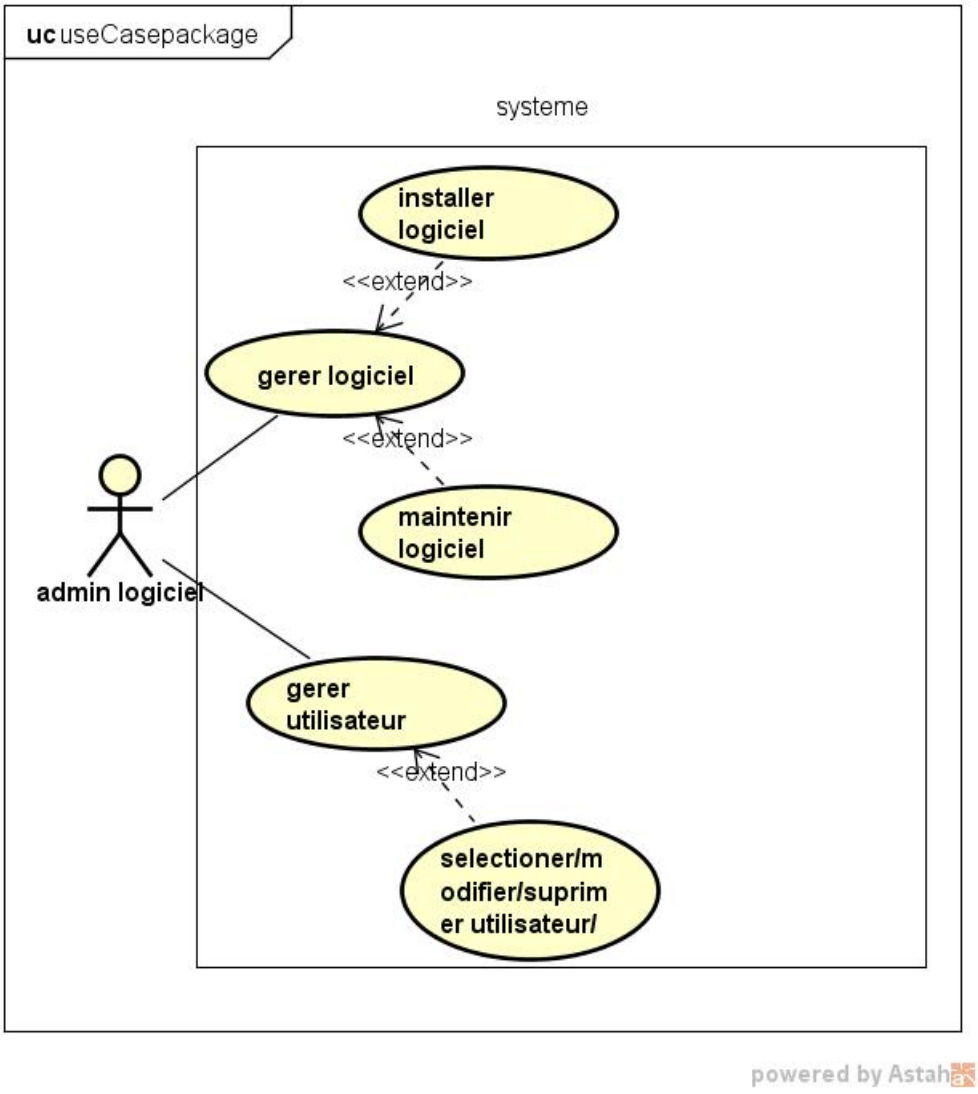

FIGURE 2.4 – Diagramme de cas d'utilisation administrateur logiciel

2.4 diagramme de séquence

2.4.1 Diagrame de sequence gestion patient

2.4.2 Diagramme de séquence gestion facturation

2.5 Diagramme de classe de l'application

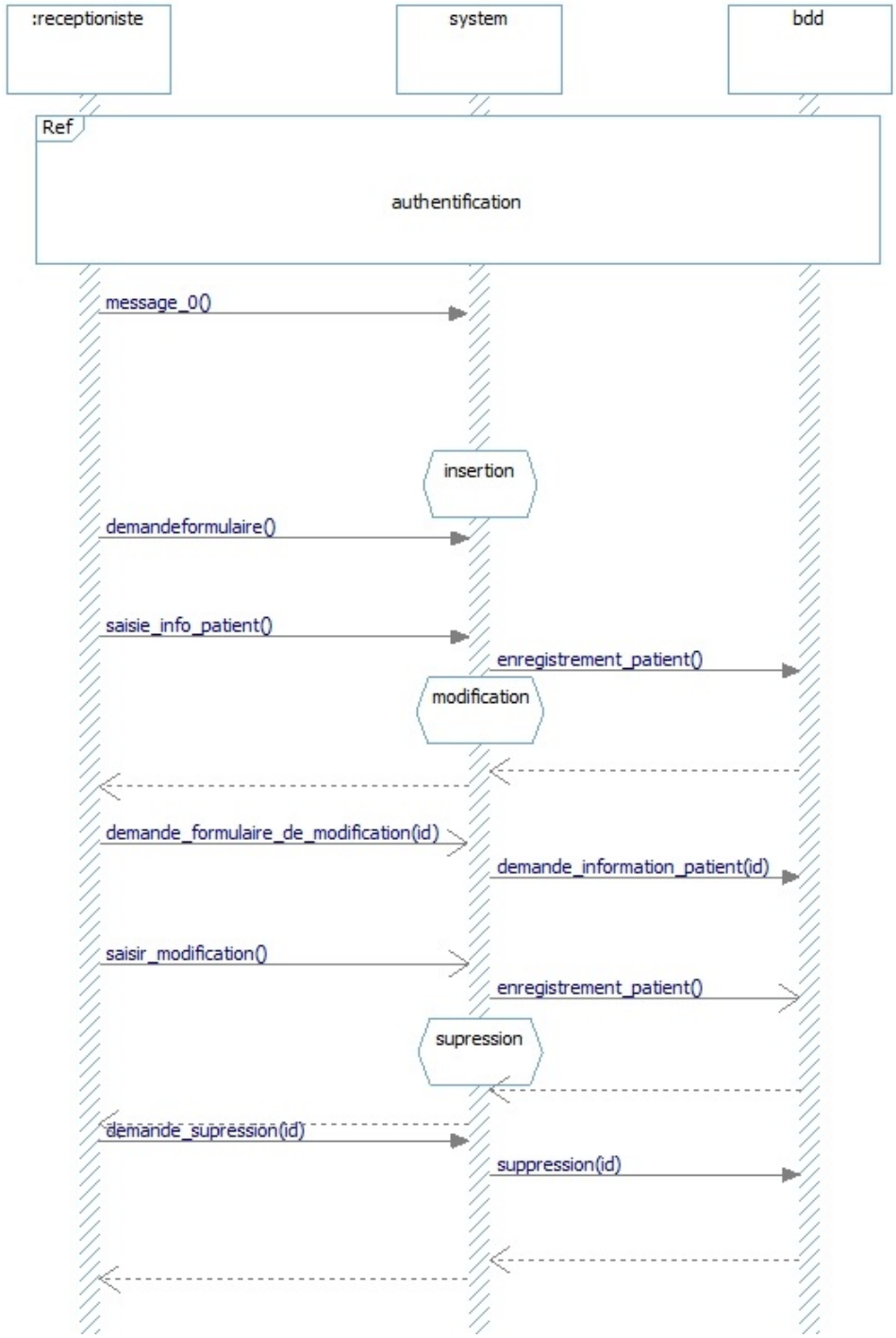

FIGURE 2.5 – Diagramme de séquence gestion patient

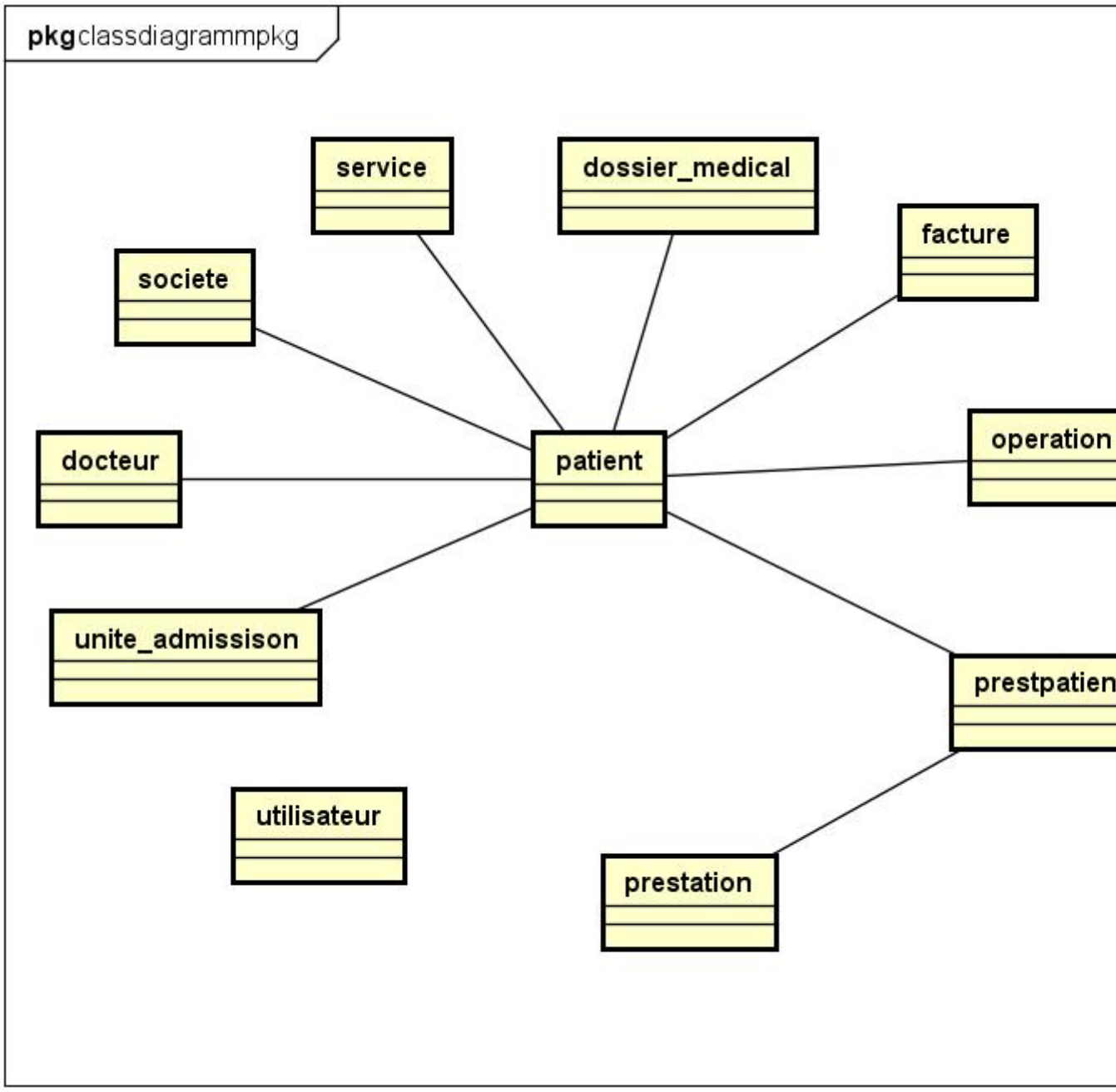

FIGURE 2.6 – Diagramme de classe de l'application

«««< HEAD

Chapitre 3

Programmation et réalisation du logiciel

=======

Chapitre 4

Modélisation de la base de donne

»»»»> add5a5e37f01a192b346b315d06b0689239d27ba

Sommaire

4.1 Introduction 20
4.2 Le framwork Laravel 20

4.1 Introduction

La réalisation et le codage de l'application est une étape cruciale et inévitable, dans le développement de l'application, l'application ne se construit tout seul après la modélisation, pour que l'application ne soit pas devenir une simple rêve il faut le produire. Nous avons utilisé laravel pour le codage.

4.2 Le framwork Laravel

Les applications Laravel sont installées et gérées avec Composer , un gestionnaire de dépendance PHP. Il existe deux manières de créer une nouvelle application Laravel.
Sur le terminal : composer create-project laravel/laravel [foldername]

«««< HEAD =======

Chapitre 5

Programmation et réalisation du logiciel

Sommaire

5.1 Introduction 23
5.2 Les Systèmes de Gestion de Base de Données (SGBD) 23
5.2.1 Définition 23
5.2.2 Firebase 23
5.2.3 Pourquoi le choix de Firebase 23
5.3 Les langages de programmation 24
5.3.1 De JSF à Angular 24
5.3.2 Angular 24
5.3.3 Une navigation plus fluide pour le visiteur 25
5.3.4 Une meilleure gestion de contenu dynamique 25
5.3.5 Une plateforme extensible et modulaire 25
5.3.6 Présentation en HTML5/CSS3/JS/jQuery 26
5.4 Conception 26
5.4.1 Création du projet 26
5.4.2 La structure des dossiers 26
5.4.3 Firebase Realtime Database, une base de données en temps réel . . 28
5.4.4 Sécurisation des données 29

5.1 Introduction

Après avoir franchi l'étape de 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.

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

5.2.1 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.

5.2.2 Firebase

Firebase est à la fois un SGBD et un ensemble de services d'hébergement pour n'importe quel type d'application (Web service, Android, iOS, Javascript, Node.js, Java, Unity, PHP, C++ ...). Il propose d'héberger en NoSQL et en temps réel des bases de données, du contenu, de l'authentification sociale (Google, Facebook, Twitter et Github), et des notifications, ou encore des services, tel que par exemple un serveur de communication temps réel [**?**].

5.2.3 Pourquoi le choix de Firebase

Lorsqu'on développe une application, qu'elle soit destinée au grand public ou réservée à un usage interne à l'entreprise, certaines fonctionnalités sont systématiquement requises, telles que la gestion des utilisateurs, de la connexion et des notifications. La gestion de ces fonctionnalités est fastidieuse, répétitive si le système se compose de plusieurs applications, et critiques en termes de sécurité, dans la mesure où l'on

va stocker des mots de passe. D'où l'intervention de Firebase qui va nous permettre d'externaliser cette gestion.

5.3 Les langages de programmation

La présentation des langages de programmation sont des facteurs importants pour la réussite de la conception et la réalisation d'un logiciel. En effet, chaque langage a ses propres caractéristiques qui doivent être compatibles avec les contraintes et conditions exigées par le cahier des charges. Le choix doit être fait en fonction de leur souplesse, leur adaptation aux fonctions exigées et aux ressources matérielles disponibles.

5.3.1 De JSF à Angular

Au début nous avons utiliser le langage "Multi-end" : JSF (Java Server Faces), qui est un framework MVC (Modèle Vue Controleur) Java traditionnels à base d'actions et basé sur la notion de composants. Mais en allant plus loin dans les détails, nous pouvons en déduire des principaux inconvénients qui ont empêchés la progression du développement de l'application.

Deux inconvénients majeurs JSF :

1. Grande courbe d'apprentissage. JSF est complexe, c'est juste vrai.
2. Sa nature composante. Le framwork basé sur les composants essaye de cacher la vraie nature du Web, qui vient avec une énorme quantité de complications et de désastres (comme ne pas soutenir GET dans JSF). Chaque framework basé sur des composants ajoute de l'abstraction au développement Web, et cette abstraction se traduit par un surcoût inutile et une complexité plus élevée.

En bref, JSF essaient de cacher au programmeur la vrai requête/réponse et la nature sans état du web. C'est un désavantage majeur pour JSF, certe il peut convenir à certains types d'applications (intranet, formulaires intensifs), mais pour une application web réelle, ce n'est pas une bonne solution.

5.3.2 Angular

Après avoir été bloqué avec JSF, on a recommencer le développement avec Angular, un langage fessant partie de la nouvelle vague de frameworks JavaScript portée par Google. C'est un socle technique qui se veut extensible et qui pousse vers un développement

structuré. Il s'inscrit dans un mouvement d'innovation côté front-end, dont le but est d'éviter le chargement d'une nouvelle page à chaque action demandée [**?**].

Angular présente la particularité d'être totalement frontend (côté client). Pour utiliser Angular dans notre application, nous allons devoir utiliser un système backend (côté serveur), dans notre cas ce sera Firebase pour gérer la connexion à la base de données. En utilisant ce langage, on voit clairement la distinction de développement entre front et back.

5.3.3 Une navigation plus fluide pour le visiteur

L'utilisation d'Angular impose un développement selon la structure MVVM (Modèle-Vue-Vue-Modèle). Ce principe offre un avantage de taille, celui de diminuer considérablement la vitesse de chargement des pages. En effet, le nombre d'accès au serveur est fortement diminué car la communication se fait majoritairement en mode asynchrone. Autrement dit, l'interface visuelle est portée côté client. En conséquence, une importante partie des requêtes supportées en arrière-plan est ainsi supprimée, ce qui permet de concevoir des applications web plus légères. Ceci explique sa parfaite adaptation pour les applications web monopage (SPA) qui ne comportent qu'une seule et unique interface.

5.3.4 Une meilleure gestion de contenu dynamique

Le framework estampillé Google étend le langage HTML traditionnel pour enrichir davantage le contenu dynamique par le biais d'un couplage bidirectionnel (two-way data-binding). Derrière ce nom barbare se cache un concept très pratique : dès qu'une vue est modifiée, la donnée est envoyée au model associé qui rafraîchit à son tour la vue. Concrètement, si un internaute remplit un champ texte, la valeur saisie peut s'afficher à un autre endroit de la page et ce sans rechargement ni soumission au préalable de l'information. Il s'agit donc d'une synchronisation entre le modèle et la vue qui permet de créer des applications plus responsives [**?**].

5.3.5 Une plateforme extensible et modulaire

Pour pallier à la nature statique de la solution HTML, Angular introduit la notion de directives chargée d'associer un comportement JavaScript spécifique à chaque nouvel élément de ce langage balisé. Ces composants vont permettre de rendre le code extensible et modulable. Il devient alors facile d'ajouter, de modifier ou de supprimer des directives,

ce qui fait entre autre la popularité d'Angular. Celles-ci peuvent tout-à-fait être partagées et réutilisées de projet en projet pour éviter de réinventer la roue [**?**].

5.3.6 Présentation en HTML5/CSS3/JS/jQuery

Pour afficher, mettre en forme et structurer les données à l'utilisateur, nous utilisons les langages HTML (Hypertext Markup Language), le CSS (Cascading Style Sheets) et éventuellement le Javascript (JS) pour l'interactivité de l'application. Ceux-ci ont une caractéristique commune importante : ils sont tous interprétés par le navigateur, directement sur la machine client.

L'utilisation de jQuery est aussi un atout, c'est une bibliothèque JavaScript libre et multi-plateforme créée pour faciliter l'écriture de scripts côté client dans le code HTML.
Cette bibliothèque va nous servir notamment au fonctionnalités suivantes :

- Manipulation des feuilles de style en cascade (ajout/suppression des classes, d'attributs...) ;
- Événements ;
- Effets visuels et animations ;

5.4 Conception

5.4.1 Création du projet

Nous devons avant tout installé Node et npm. Ensuite pour installer angular CLI, il suffit de taper la commande suivante dans votre bash :

```
npm install -g @angular/cli
```

Pour crée maintenant notre application, il suffit d'exécuter la commande suivante :

```
ng new g4sm
```

5.4.2 La structure des dossiers

Après l'installation et la création de votre projet avec angular-cli, nous allons voir la structure des dossiers et fichiers dans l'architecture d'angular-cli.

```
// Tout ce qui va concerner les tests end to end
|- e2e/
```

```
|----- app.e2e-spec.ts
|----- app.po.ts
|----- tsconfig.e2e.json

// les dépendances avec npm
|- node_modules/

// l'endroit où les fichiers de build seront mis
|- dist/

// Le dossier où vous allez modifier vos fichiers de code
//Là où va se trouver vos composants, services, etc..
|- src/
|----- app/
|----- app.component.css|html|spec.ts|ts
|----- app.module.ts
|----- assets/
|----- environments/
|----- environment.prod.ts|ts
|----- favicon.ico
|----- index.html
|----- main.ts
|----- polyfills.ts
|----- styles.css
|----- test.ts
|----- tsconfig.app.json
|----- tsconfig.spec.json
|----- typings.d.ts

// la configuration globale de votre application
|- .angular-cli.json  // the main configuration file
|- .editorconfig      // editorconfig which is used in some VS Code setups
|- .gitignore
```

```
|- karma.conf.js
|- package.json
|- protractor.conf.js
|- README.md
|- tsconfig.json
|- tslint.json
```

Nous allons quasiment passer tout notre temps dans le dossier *src/app*. Ce dossier contient presque tous les fichiers dont nous avons besoin pour coder notre application. Les fichiers contenus dans ce dossier sont ensuite compilés dans le dossier *dist*. Nous pouvons aussi installer des dépendances avec le gestionnaire de package de node : "npm". Ces dépendances seront installées dans le dossier "node modules".

5.4.3 Firebase Realtime Database, une base de données en temps réel

La base de données avec Firebase se présente sous la forme d'un arbre "infini", composé uniquement d'objets clé/valeur. Il s'agit d'une base de données cloud où les données sont stockées sous format JSON (JavaScript Object Notation) et synchronisées en temps réel avec chaque client connecté. Par exemple, si on souhaite stocker un ensemble d'utilisateur, cela devrait plus ou moins ressembler à cela :

```
{
   "utilisateurs": {
      "id1": {
         "nom": "Midonique",
         "email": "mido@gmail.com",
      },
      "id2": { ... },
      "id3": { ... }
   }
}
```

5.4.4 Sécurisation des données

Pour faire simple : l'accès aux données présentes dans la base est régi par un ensemble de règles. Établissons un simple fichier de règles pour nos utilisateurs :

```
{
   "rules": {
      ".read": "auth != null",
      ".write": "auth != null"
   }
}
```

C'est uniquement un utilisateur connecté pourra lire, enregistrer et modifer ces données. Il s'agit évidemment ici d'un exemple simple : nous pouvez établir des règles de sécurité bien plus complexes afin de protéger les données des utilisateurs.

»»»> add5a5e37f01a192b346b315d06b0689239d27ba

«««< HEAD
=======
»»»> add5a5e37f01a192b346b315d06b0689239d27ba