

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

MÉMOIRE DE MASTER PROFESSIONNEL

En
Informatique
Option
Génie logiciel

Thème

Conception et réalisation d'une application
web interne avec messagerie professionnelle
cas : CHU de BEJAIA

Présenté par :

M^r :AMRANE Azzeddine

M^r :BOUREGHIT Rostom

Soutenu le 28 Juin 2017 devant le jury composé de :

Président	A. SEBAA	U. A/Mira Béjaïa.
Encadreur	S. AIT AHCENE	U. A/Mira Béjaïa.
Examineur	K. ADEL	U. A/Mira Béjaïa.
Examineur	M. CHEKRID	U. A/Mira Béjaïa.

** Remerciements **

*Nous tenons à remercier profondément notre promotrice M^{me}. AIT
AHCENE pour nous avoir dirigée tout au long de notre
travail durant toute cette année universitaire.*

*Nous remercions également notre encadreur de stage au CHU de
BEJAIA M^r. MEDAHE pour sa gentillesse et son aide précieuse,
également tout le personnel du bureau de l'information et de
l'audiovisuel.*

Ainsi que toutes les personnes qui nous ont aidé dans notre travail.

✧ *Dédicaces* ✧

Nous dédions ce travail à nos parents, nos familles et tous nos amis.

Table des matières

Table des matières	i
Table des figures	v
Liste des tableaux	vii
Notations et symboles	viii
Introduction générale	1
1 Étude et critique de l'existant	2
1.1 Introduction	2
1.2 Présentation de l'organisme d'accueil	2
1.2.1 Présentation du CHU de Bejaia	2
1.2.2 Les différentes directions de l'hôpital KHELIL AMRANE . . .	3
1.2.3 Organigramme de l'établissement	5
1.3 Critique de l'existant	5
1.4 Solution proposée	6
1.5 Analyse et spécification des besoins	6
1.5.1 Les besoins fonctionnels	7
1.5.2 Les besoins non fonctionnels	7
1.6 Méthodologie de développement	7
1.6.1 Présentation de UP (Unified process)	7
1.7 Pattern design MVC (Modèle Vue Contrôleur)	9
1.7.1 Couche modèle	9

1.7.2	Couche vue	9
1.7.3	Couche contrôle	10
1.7.4	Illustration du modèle MVC	11
1.8	Architecture client/serveur 3-tiers	11
1.9	Présentation d'UML	12
1.9.1	Définition	12
1.9.2	Les diagramme dans UML	12
1.10	Conclusion	13
2	Conception	14
2.1	Introduction	14
2.2	Diagrammes des cas d'utilisation	14
2.2.1	Identification des acteurs	14
2.2.2	Identification des cas d'utilisations	15
2.2.3	Diagramme de cas d'utilisation de l'administrateur	16
2.2.4	Diagramme de cas d'utilisation des membres	18
2.2.5	Diagramme de cas d'utilisation global	18
2.2.6	Description textuelle des cas d'utilisation :	20
2.3	Diagrammes de séquences	24
2.3.1	Élaboration des diagrammes de séquence	24
2.3.2	Diagramme de séquence envoyer un message	27
2.3.3	Diagramme de séquence Rechercher	29
2.3.4	Diagramme de séquence ajouter un membre	31
2.3.5	Diagramme de séquence modifier un membre	32
2.3.6	Diagramme de séquence supprimer un membre	35
2.4	Présentation du diagramme de classe	36
2.4.1	Présentation des classes et leurs attributs	37
2.4.2	Élaboration du diagramme de classes du système	39
2.5	Présentation du diagramme d'activité	40
2.5.1	Élaboration des diagrammes d'activité	41
2.6	Passage au modèle relationnel	44
2.7	Conclusion	46

3	Réalisation	47
3.1	Introduction	47
3.2	Langage de développement	47
3.2.1	PHP	47
3.2.2	SQL	48
3.2.3	JavaScript	48
3.3	Outils de développement	48
3.3.1	Wampserver	48
3.3.2	Mysql	49
3.3.3	Phpstorme	49
3.3.4	Sublimtext	49
3.3.5	Draw.io	49
3.3.6	Mozilla firefox	50
3.3.7	Git	50
3.4	Framework	50
3.4.1	Laravel	50
3.4.2	jQuery	51
3.4.3	Ajax	51
3.4.4	Semantic ui	52
3.5	Arborescence de l'application	52
3.5.1	Arborescence côté membre	52
3.5.2	Arborescence côté administrateur	53
3.6	Présentation des interfaces pour les membres	54
3.6.1	Interface d'authentification	54
3.6.2	Page d'accueil	55
3.6.3	Affichage des actualités	56
3.6.4	Calendrier des événements	57
3.6.5	Affichage d'un événement	58
3.6.6	Affichage des sondages	59
3.6.7	Résultat du sondage	59
3.6.8	Boite de réception	60
3.6.9	Nouveau message	60

3.7	Interfaces de l'administrateur	61
3.7.1	Page de gestion du contenu	61
3.7.2	Gestion des utilisateurs	62
3.7.3	Ajouter un membre	63
3.7.4	Modifier un membre	64
3.7.5	Supprimer un membre	65
3.8	Diagramme de déploiement	66
3.9	Conclusion	66
Conclusion et perspectives		67
Bibliographie		69

Table des figures

1.1	Organigramme de l'établissement	5
1.2	Schéma du pattern MVC [1]	11
1.3	Illustration d'une architecture 3-tiers [19]	12
2.1	Diagramme de cas d'utilisation de l'administrateur :	17
2.2	Diagramme de cas d'utilisation des membres :	18
2.3	Diagramme de cas d'utilisation global :	19
2.4	Diagramme de séquence du cas s'authentifier	25
2.5	Diagramme de séquence approfondie du cas s'authentifier.	26
2.6	Diagramme de séquence du cas envoyer un message.	27
2.7	Diagramme de séquence approfondie du cas envoyer un message.	28
2.8	Diagramme de séquence Rechercher.	29
2.9	Diagramme de approfondie séquence Rechercher.	30
2.10	Diagramme de séquence du cas ajouter un membre.	31
2.11	Diagramme de séquence approfondie du cas ajouter un membre.	32
2.12	Diagramme de séquence du cas modifier un membre.	33
2.13	Diagramme de séquence approfondie du cas modifier un membre.	34
2.14	Diagramme de séquence du cas supprimer un membre.	35
2.15	Diagramme de séquence approfondie du cas supprimer un membre.	36
2.16	Diagramme de classes du système.	39
2.17	Diagramme d'activité s'authentifier.	41
2.18	Diagramme d'activité ajouter membre.	42
2.19	Diagramme d'activité modifier membre.	43
2.20	Diagramme d'activité supprimer membre.	44

3.1	Arborescence côté membre.	52
3.2	Arborescence côté administrateur.	53
3.3	Interface d'authentification.	54
3.4	Page d'accueil.	55
3.5	Affichage d'une actualité.	56
3.6	Calendrier des événements.	57
3.7	Affichage d'un événement.	58
3.8	Sondage.	59
3.9	Résultat de sondage.	59
3.10	Boite de reception.	60
3.11	Nouveau message.	60
3.12	Page de gestion du contenu.	61
3.13	Page de gestion des membres.	62
3.14	Page ajout de membre.	63
3.15	Page modification de membre.	64
3.16	Page suppression de membre.	65
3.17	Diagramme de déploiement.	66

Liste des tableaux

2.1	Tableau des cas d'utilisation.	16
2.2	Description textuelle du cas d'utilisation N°1.	20
2.3	Description textuelle du cas d'utilisation N°2.	21
2.4	Description textuelle du cas d'utilisation N°6.	22
2.5	Description textuelle du cas d'utilisation N°7.	22
2.6	Description textuelle du cas d'utilisation N°8.	23
2.7	Description textuelle du cas d'utilisation N°10.	23
2.8	Description textuelle du cas d'utilisation N°11.	24
2.9	Description des classes et leurs attributs.	38

Notations et symboles

- **AJAX** :Asynchronous Javascript And Xml.
- **ANSI** :Agence Nationale de la Sécurité Informatique.
- **BDD** :Base De Données.
- **BLL** :Business Logic Layer.
- **CHU** :Centre Hospitalier Universitaire.
- **CSS** :Cascading Style Sheets.
- **DAL** :Data Acces Layer.
- **DOM** :Document Object Model.
- **GNU** :GNU's Not UNIX.
- **HTML** :HyperText Markup Language.
- **HTTP** :HyperText Transfer Protocol.
- **IDE** :Integrated Development Environment.
- **ISO** :Organisation Internationale de Normalisation.
- **LAMP** :Linux, Apache, Mysql/MariaDB, PHP/Phyton/Perl.
- **MVC** :Modèle Vue Contrôleur.
- **MySql** :My Structured Query Language.
- **OS** :Operating System.
- **PHP** :Personal Home Page.
- **SD** :Sous Direction.
- **SGBD** :System de Gestion de Base de Données.
- **SQL** :Structured Query Language.
- **UP** :Unified Process.
- **UI** :User Interface.

- **UML** :Unified Modeling Language.
- **XML** :Extensible Markup Language.

Introduction générale

L'apparition du courrier électronique dans les années 70 a changé radicalement le monde du travail. Dans les entreprises, l'usage des ordinateurs et du matériel mobile a fortement contribué à la modernisation du milieu professionnel. La messagerie électronique est considérée comme l'application par excellence pour la communication et l'échange d'informations. Facile et rapide d'utilisation, elle permet d'échanger des messages et des documents avec un vaste auditoire.

La communication actuelle entre les différents acteurs de la gestion du CHU de Béjaia se fait par simple courrier physique, ce qui engendre un retard dans la diffusion de l'information ainsi qu'un risque de perte ou de détérioration du courrier.

La réalisation d'une plate-forme de communication comblerait ces lacunes tout en améliorant la communication de la bonne information plus rapidement et efficacement.

Dans le premier chapitre nous présenterons et ferons une étude de l'organisme d'accueil.

Le deuxième chapitre portera sur la spécification des besoins et la conception du projet.

Enfin, le troisième chapitre sera consacré à la réalisation du système.

Étude et critique de l'existant

1.1 Introduction

Dans ce chapitre nous allons présenter l'organisme d'accueil dans lequel nous avons effectué notre stage, la méthode de développement que nous avons suivie ainsi que l'architecture qui sera utilisée et pour finir nous présenterons le processus unifier du formalisme UML et le pattern design MVC.

1.2 Présentation de l'organisme d'accueil

1.2.1 Présentation du CHU de Bejaia

Le CHU de Bejaia a été créé par le décret exécutif n° 09-319 du 17 chaoual 1430 correspondant au 6 octobre 2009 complétant la liste des centres hospitalo-universitaires annexée au décret exécutif n° 97-467 du 2 Chaàbane 1418 correspondant au 23 décembre 1997 fixant les règles de création, d'organisation et de fonctionnement des centres hospitalo-universitaires.

Le Centre hospitalo-universitaire de Bejaia est annexé en trois centres qui sont :

- Hôpital Khelil Amrane ;
- Hôpital Frantz Fanon ;

- Hôpital Targa Ouzemmour.

En 2011, l'hôpital Khellil Amrane est devenu le siège du Centre hospitalo-universitaire (CHU) de Bejaïa. La création de ce dernier est faite suite à l'inauguration de la faculté de médecine.

Il est placé sous la tutelle administrative du ministre chargé de la santé. La tutelle pédagogique est assurée par le ministre chargé de l'enseignement supérieur.

Le CHU est chargé, en relation avec l'établissement d'enseignement et/ou de formation supérieure en sciences médicales concerné, des missions : de diagnostic, d'exploration, de soins, de prévention, de formation, d'études et de recherche.[2]

1.2.2 Les différentes directions de l'hôpital KHELIL AM-RANE

Direction des activités médicales et paramédicales :

Assure principalement la gestion statutaire du personnel médicale et des internes, le suivi de la politique médicale et des formations (développement professionnel continu), notamment en termes de gestion prévisionnelle des emplois et des compétences médicales.

Constituée de quatre (4) sous-direction :

- Sous-direction des activités médicale ;
- Sous-direction des activités paramédicale ;
- Sous-direction de la gestion administrative du malade ;
- Sous-direction de la formation et documentation.

Direction des ressources humaines :

Constitué d'une seule sous-direction qui est celle du personnel qui a pour but de

conseiller et accompagner la direction générale dans ses orientations stratégiques en tenant compte du capital humain. Il définit et valide les grands axes de la politique des ressources humaines et prends souvent en charge les relations sociales.

Direction des finances et du contrôle :

A pour rôle l'élaboration de la stratégie financière pluriannuelle, du programme d'investissement et des prévisions d'exploitation, la gestion du budget annuel, le suivi mensuel de son exécution ainsi que la contribution à l'amélioration de la qualité et de la sincérité des comptes de l'établissement.

Constituée de deux (2) sous-direction :

- Sous-direction Du budget et comptabilité ;
- Sous-direction du calcul des coûts.

Direction des moyens matériels :

C'est la direction qui est responsables en matière d'investissement, elle décide des investissements de matériel et prépare les budgets d'investissement. Elle veille aussi au respect des commandes et contrôle et supervise l'entretien des parcs de matériel des établissements en mettant en place des plannings d'entretien.

Constituée de trois (3) sous-direction :

- Sous-direction économique ;
- Sous-direction des équipement, infrastructure et maintenance ;
- Sous-direction des produits pharmaceutique.

1.2.3 Organigramme de l'établissement

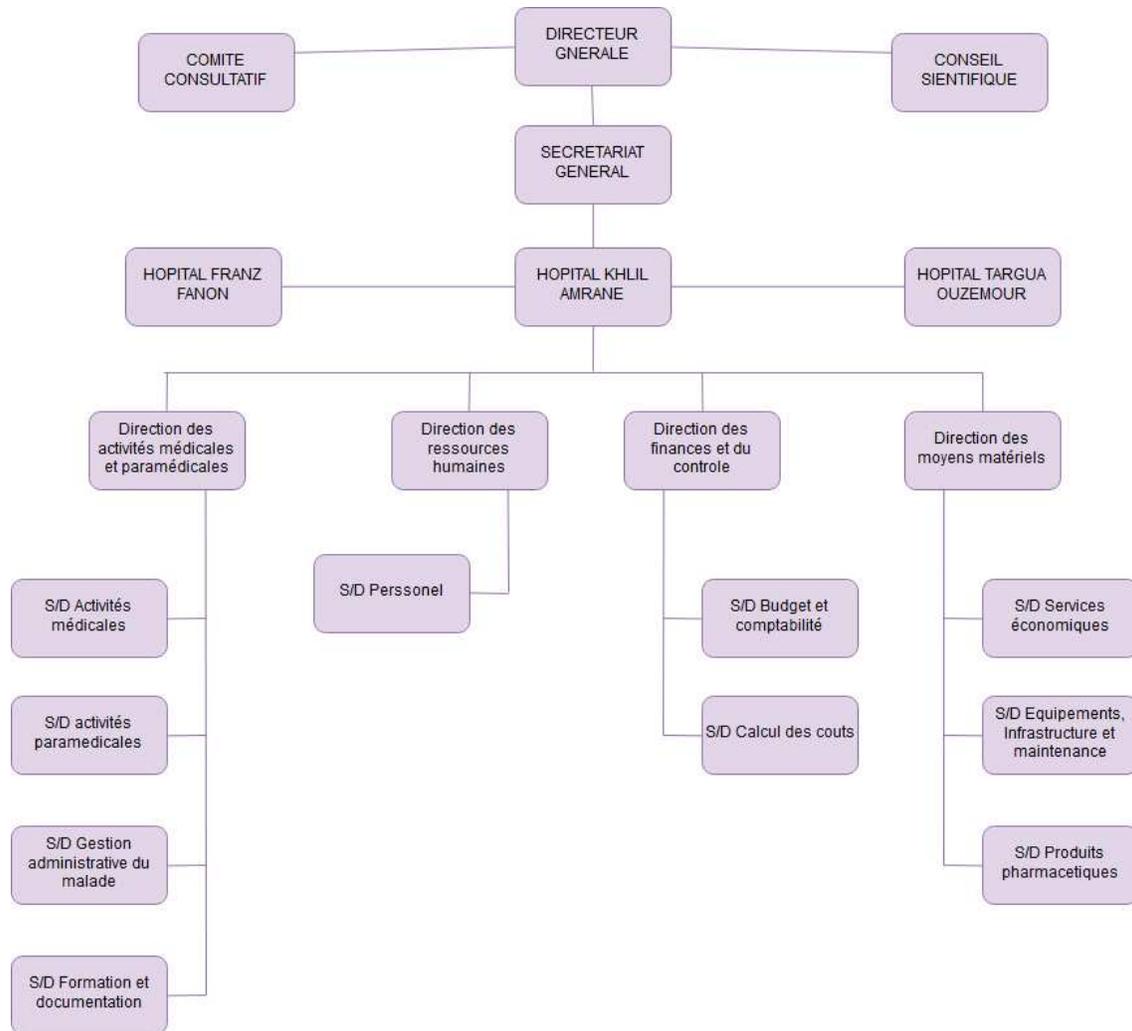


FIGURE 1.1 – Organigramme de l'établissement

1.3 Critique de l'existant

Les visites de l'établissement au courant de notre stage nous ont permis de détecter un certain nombre de lacunes, parmi elles nous avons constaté qu'aucun

outil d'administration permettant la communication entre les différents acteurs de la gestion du CHU n'est utilisé, cela se fait par simple courrier physique et implique un retard ou une perte des documents et plus de moyens pour leur livraison.

Aussi nous avons remarqué une absence totale de plateformes de communication cela rend plus difficile la gestion des courriers ainsi que la diffusion d'information ou de documents.

1.4 Solution proposée

Afin de combler les lacunes citées précédemment nous proposons de concevoir une application avec une interface conviviale et intuitif tout en étant la plus efficace possible, avec les fonctionnalités suivantes :

- Concevoir un système de communication entre les cadres des directions et structures hospitalières ;
- Un système qui va permettre la diffusion d'informations à une -ou plusieurs personnes concernées ;
- Garder une trace de tous les documents échangés ;
- Centraliser et unifier la plateforme de communication pour plus de sécurité ;
- Afficher les événements dans un calendrier (séminaire, réunions . . .) ;
- Offrir une interface permettant la gestion des comptes ainsi que tous les affichages d'événements et d'informations sur l'application.

1.5 Analyse et spécification des besoins

Avec le client, nous avons identifié les besoins de notre application, cela constituera la phase de départ pour concevoir les fonctionnalités de l'application. Nous allons définir les besoins fonctionnels qui représenteront les fonctionnalités principales exigées par le client et les besoins non fonctionnels qui décrivent des contraintes techniques auxquelles sera soumis le système.

1.5.1 Les besoins fonctionnels

- **Messagerie professionnelle interne** : Tous les membres inscrits dans l'application doivent pouvoir communiquer entre eux à travers la messagerie.
- **Diffusion d'informations** : Les responsables chargées de la communication au sein de l'établissement doivent pouvoir diffuser des circulaires, notes et diverses informations au sein des différentes structures de l'établissement.
- **Traçabilité des documents** : Toute trace de documents et messages échangée dans l'application doit être sauvegardée.
- **Gestion de membres** : la gestion des membres qui auront accès à l'application sera faite par les administrateurs.
- **Espace d'affichage** : Un espace d'affichage pour tout événement organisé.
- **L'authentification** : Tous les utilisateurs de l'application doivent s'authentifier pour accéder à leur compte.

1.5.2 Les besoins non fonctionnels

- **La disponibilité** : L'application doit être disponible partout et à tout moment par les utilisateurs.
- **Fiabilité** : Toutes les données de l'application doivent être fiables.
- **Ergonomie de l'interface graphique** : L'application doit fournir une interface ergonomique de façon à faciliter son utilisation.
- **Rapidité de traitement** : l'application doit assurer une rapidité de traitement proche d'une exécution en temps réel.

1.6 Méthodologie de développement

1.6.1 Présentation de UP (Unified process)

UP (Unified process) est une méthode générique de développement logiciel orienté objet, itératif et incrémental, piloté par les cas d'utilisation et centré sur l'architecture, il peut être adapté à une large classe de systèmes logiciels et à

différents domaines d'application. Il est nécessaire de l'adapter en fonction du contexte de l'équipe et de l'organisation.[20]

Les phases du processus unifié :

La méthode UP se base sur quatre phases :

- **Analyse des besoins** : Établir une vision globale du projet où on spécifie les besoins et on étudie la faisabilité du projet.
- **Élaboration** : On reprend les éléments de l'analyse des besoins et on développe une architecture de référence, les risques et la plupart des besoins sont identifiés.
- **Construction** : Finaliser l'analyse, la conception, l'implémentation et les tests puis transformer l'architecture de référence en produit exécutable tout en veillant à respecter son intégrité.
- **Transition** : Livraison du produit au client afin d'effectuer des essais pour détecter d'éventuelles anomalies.

Les activités du processus unifié :

Chaque phase est constituée d'une succession d'activités. Les activités de la méthode UP sont :

- **Expression des besoins** : Compréhension et expression des besoins et des exigences du client qu'elle soit fonctionnelle ou non fonctionnelle.
- **Analyse** : Permet d'accéder à une compréhension des besoins et des exigences du client et livrer des spécifications issues des cas d'utilisation.
- **Conception** : Permet d'acquérir une compréhension approfondie des contraintes liées aux outils de réalisation en prenant en compte le choix d'architecture technique retenu pour le développement et l'exploitation système.
- **Implémentation** : On implémente le système sous forme de composant, bibliothèque et de fichier. Elle a pour objectif de planifier l'intégration des

composants et de produire les classes et les sous-systèmes sous forme de codes sources.

- **Tests** : Permettent de vérifier les résultats de l'implémentation de toutes les exigences et de s'assurer de la bonne intégration de tous les composants dans le logiciel.
- **Déploiement** : Livraison et exploitation du produit.

1.7 Pattern design MVC (Modèle Vue Contrôleur)

C'est un modèle d'architecture qui cherche à séparer nettement les couches de présentation (UI : User Interface), métier (BLL : Business Logic Layer) et d'accès aux données (DAL : Data Access Layer). Le but étant d'avoir une dépendance minimale entre les différentes couches de l'application ; ainsi les modifications effectuées sur n'importe quelle couche de l'application n'affectent pas les autres couches.[4]

1.7.1 Couche modèle

La couche modèle représente la partie de l'application qui exécute la logique métier. Cela signifie qu'elle est responsable de récupérer les données, de les convertir selon des concepts chargés de sens pour l'application, tel que le traitement, la validation, l'association et beaucoup d'autres tâches concernant la manipulation des données.

À première vue, l'objet Model peut être vu comme la première couche d'interaction avec n'importe quelle base de données de l'application. Mais plus globalement, il fait partie des concepts majeurs autour desquels l'application est exécutée.[5]

1.7.2 Couche vue

La vue retourne une présentation des données venant du modèle. Étant séparée par les Objets Model, elle est responsable de l'utilisation des informations dont elle

dispose pour produire une interface de présentation de l'application.

Par exemple, de la même manière que la couche Model retourne un ensemble de données, la Vue utilise ces données pour fournir une page HTML. Ou un résultat XML formaté.

La couche Vue n'est pas seulement limitée au HTML ou à la représentation en texte de données. Elle peut aussi être utilisée pour offrir une grande variété de formats en fonction de vos besoins, comme les vidéos, la musique, les documents et tout autre format auquel vous pouvez penser.[5]

1.7.3 Couche contrôle

La couche contrôle gère les requêtes des utilisateurs. Elle est responsable de retourner une réponse avec l'aide mutuelle des couches Model et Vue.

Les controllers peuvent être imaginés comme des managers qui ont pour mission de vérifier que toutes les ressources souhaitées pour accomplir une tâche soient déléguées aux travailleurs de façon appropriée. Il attend des requêtes des clients, vérifie leur validité selon l'authentification et les règles d'autorisation, délègue les données récupérées et traitées par le Model, et sélectionne les types de présentation correcte que le client accepte, pour finalement déléguer le processus d'affichage à la couche Vue.[5]

1.7.4 Illustration du modèle MVC

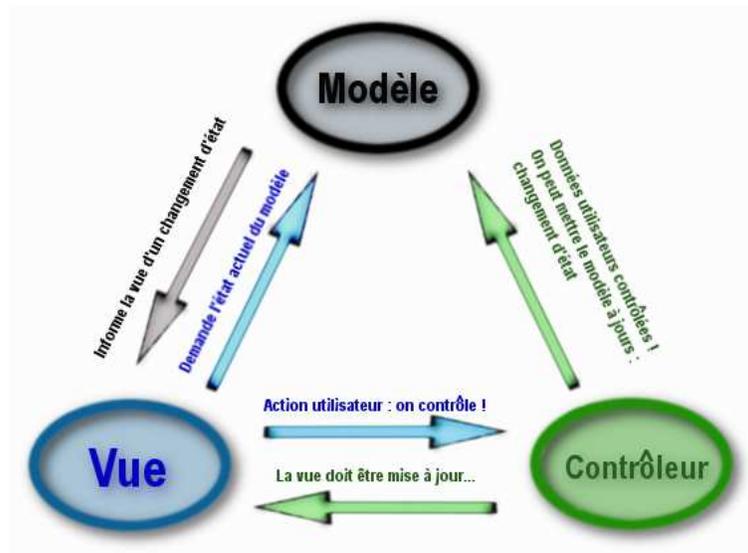


FIGURE 1.2 – Schéma du pattern MVC [1]

1.8 Architecture client/serveur 3-tiers

L'architecture à deux niveaux (aussi appelée architecture 2-tiers, tiers signifiant rangé en anglais) caractérise les systèmes clients/serveurs pour lesquels le client demande une ressource. Dans l'architecture à 3 niveaux il existe un niveau intermédiaire, c'est-à-dire que l'on a généralement une architecture partagée entre :

- Un client, qui représente l'ordinateur demandeur de ressources, il est équipé d'une interface utilisateur (généralement un navigateur web) chargée de la présentation ;
- Le serveur d'application (appelé également middleware), chargé de fournir la ressource, mais faisant appel à un autre serveur ;
- Le serveur de données, fournissant au serveur d'application les données dont il a besoin.[19]

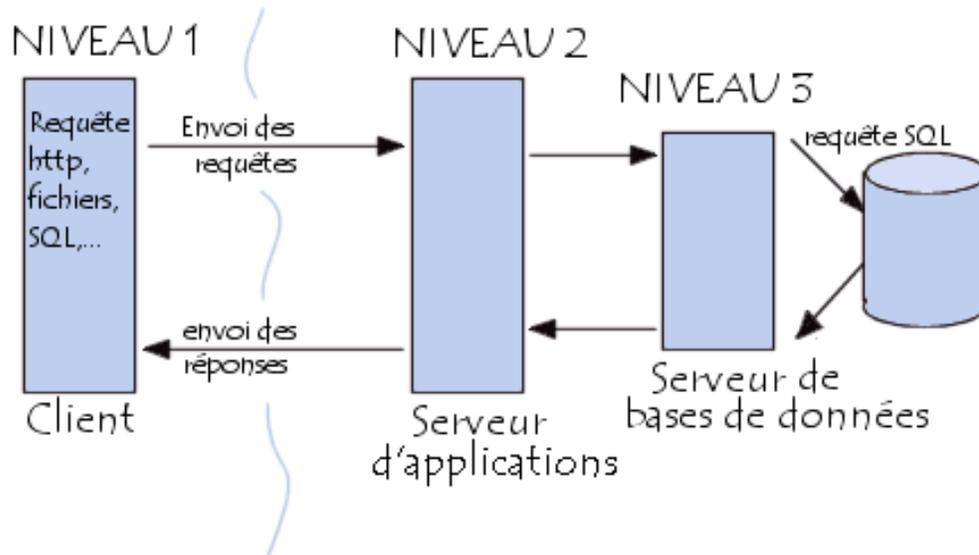


FIGURE 1.3 – Illustration d'une architecture 3-tiers [19]

1.9 Présentation d'UML

1.9.1 Définition

UML se définit comme un langage de modélisation graphique et textuel destiné à comprendre et décrire des besoins, spécifier et documenter des systèmes, esquisser des architectures logicielles, concevoir des solutions et communiquer des points de vue.[22]

1.9.2 Les diagramme dans UML

Dans sa version 2.0 UML propose treize types de diagrammes qui sont regroupés dans deux grands ensembles.

Diagrammes structurels :

On trouve six diagrammes qui permettent de représenter l'aspect statique du système.

- Diagramme de classe ;
- Diagramme d'objet ;
- Diagramme de composant ;
- Diagramme déploiement ;
- Diagramme de paquetage ;
- Diagramme de structure composite.

Diagrammes de comportement : Ces diagrammes permettent de représenter l'aspect dynamique du système, en mettant en évidence les réactions face aux événements et la production des résultats attendus. UML propose sept diagrammes :

- Diagramme de cas d'utilisation ;
- Diagramme d'activités ;
- Diagramme d'état transition ;
- Diagramme de séquence ;
- Diagramme de communication ;
- Diagramme global d'interaction ;
- Diagramme de temps.

1.10 Conclusion

Dans ce chapitre nous avons présenté l'organisme d'accueil, dans lequel nous avons constaté des lacunes au niveau de la communication, nous y avons proposé une solution, définit les besoins fonctionnel et non fonctionnel de l'application, la méthodologie de développement, le pattern design, l'architecture que nous allons adopté et pour finir une présentation du formalisme UML.

2.1 Introduction

Avant d'entamer tout projet, il est nécessaire de faire une modélisation du système, la conception constituera la base de départ de notre travail après avoir pris connaissance des problèmes rencontrés par l'établissement " CHU KHLIL AM-RANE " et identifier tous les besoins fonctionnels ainsi que non fonctionnels. Nous allons les projeter dans des diagrammes de cas d'utilisation, de séquences de classes et d'activités pour obtenir une vue claire des besoins du projet.

2.2 Diagrammes des cas d'utilisation

Un cas d'utilisation permet de décrire l'interaction entre les acteurs et le système. La description de l'interaction est réalisée suivant le point de vue de l'utilisateur.

2.2.1 Identification des acteurs

Un acteur est une entité externe qui interagit directement avec le système. Dans notre système il existe deux acteurs à savoir :

- **Administrateur** : C'est la personne chargée d'administrer l'environnement. Elle possède un compte de type administrateur, il peut y avoir plusieurs administrateurs.

- **Membre** :Les membres de l'établissement peuvent consulter et envoyer des messages et aussi avoir accès à diverses informations.

2.2.2 Identification des cas d'utilisations

L'étude de cas d'utilisation a pour objectif de déterminer ce que chaque utilisateur attend du système. La détermination du besoin est basée sur la représentation de l'interaction entre l'acteur et le système.

S'authentifier : permet d'identifier l'utilisateur et de donner accès au système avec les privilèges propices à chaque utilisateur.

Mettre à jour membre : permet à l'administrateur de gérer la liste des membres qui auront accès au système.

Mettre à jour événement : permet à l'administrateur de gérer tous les événements sur le système.

Mette à jour actualités : permet à l'administrateur de gérer les différentes informations sur le système.

Mettre à jour sondage : permet à l'administrateur de gérer les sondages.

Rechercher membre : permet d'effectuer une recherche dans la liste des membres.

Rechercher actualités : permet d'effectuer une recherche dans la liste des actualités.

Rechercher événement : permet d'effectuer une recherche dans la liste des événements.

Rechercher sondage : permet d'effectuer une recherche dans la liste des sondages.

Envoyer un message : permet à un utilisateur d'envoyer des messages.

Consulter un message : permet à un utilisateur de consulter les messages reçus dans sa messagerie.

Consulter événement : permet à un utilisateur de consulter les évènements affichés sur le système.

Consulter actualités : permet à un utilisateur de consulter les actualités publiées sur le système.

Répondre sondage : permet au membre de répondre aux sondages.

N°	Cas d'utilisation		Acteur
1	S'authentifier		- Administrateur - Membre
2	Mettre à jour membre	- Ajouter - Modifier - Supprimer	- Administrateur
3	Mettre à jour actualités	- Ajouter - Modifier - Supprimer	- Administrateur
4	Mettre à jour événement	- Ajouter - Modifier - Supprimer	- Administrateur
5	Mettre à jour sondage	- Ajouter - Modifier - Supprimer	- Administrateur
6	Envoyer un message		- Administrateur - Membre
7	Consulter message		- Administrateur - Membre
8	Consulter évènement		- Administrateur - Membre
9	Consulter actualités		- Administrateur - Membre
10	Répondre sondage		- Administrateur - Membre
11	Rechercher membres		- Administrateur
12	Rechercher actualités		- Administrateur
13	Rechercher événements		- Administrateur
14	Rechercher sondages		- Administrateur

TABLE 2.1 – Tableau des cas d'utilisation.

2.2.3 Diagramme de cas d'utilisation de l'administrateur

Le diagramme ci-dessous décrit les cas d'utilisations identifiés pour l'administrateur.

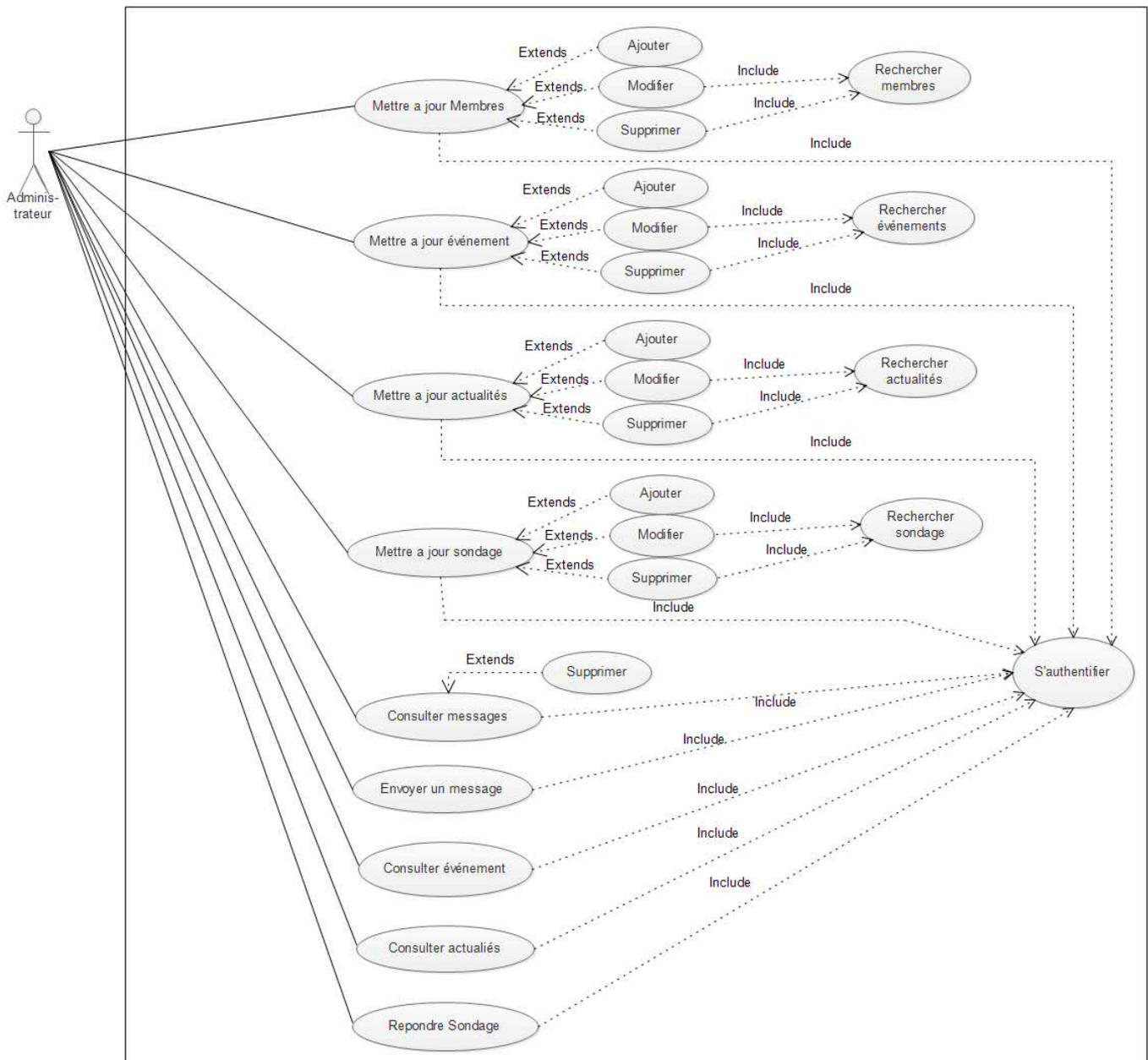


FIGURE 2.1 – Diagramme de cas d'utilisation de l'administrateur :

2.2.4 Diagramme de cas d'utilisation des membres

Le diagramme ci-dessous décrit les cas d'utilisations identifiés pour les membres.

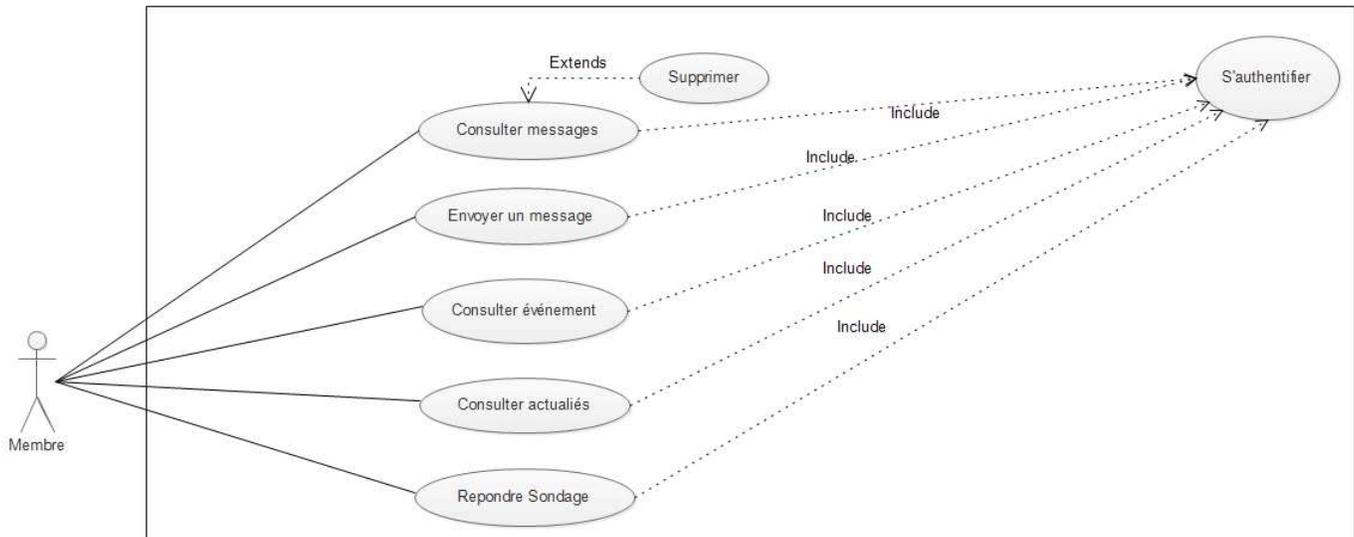


FIGURE 2.2 – Diagramme de cas d'utilisation des membres :

2.2.5 Diagramme de cas d'utilisation global

Ci-dessous le diagramme de cas d'utilisations global incluant l'administrateur et le membre avec les différentes interactions.

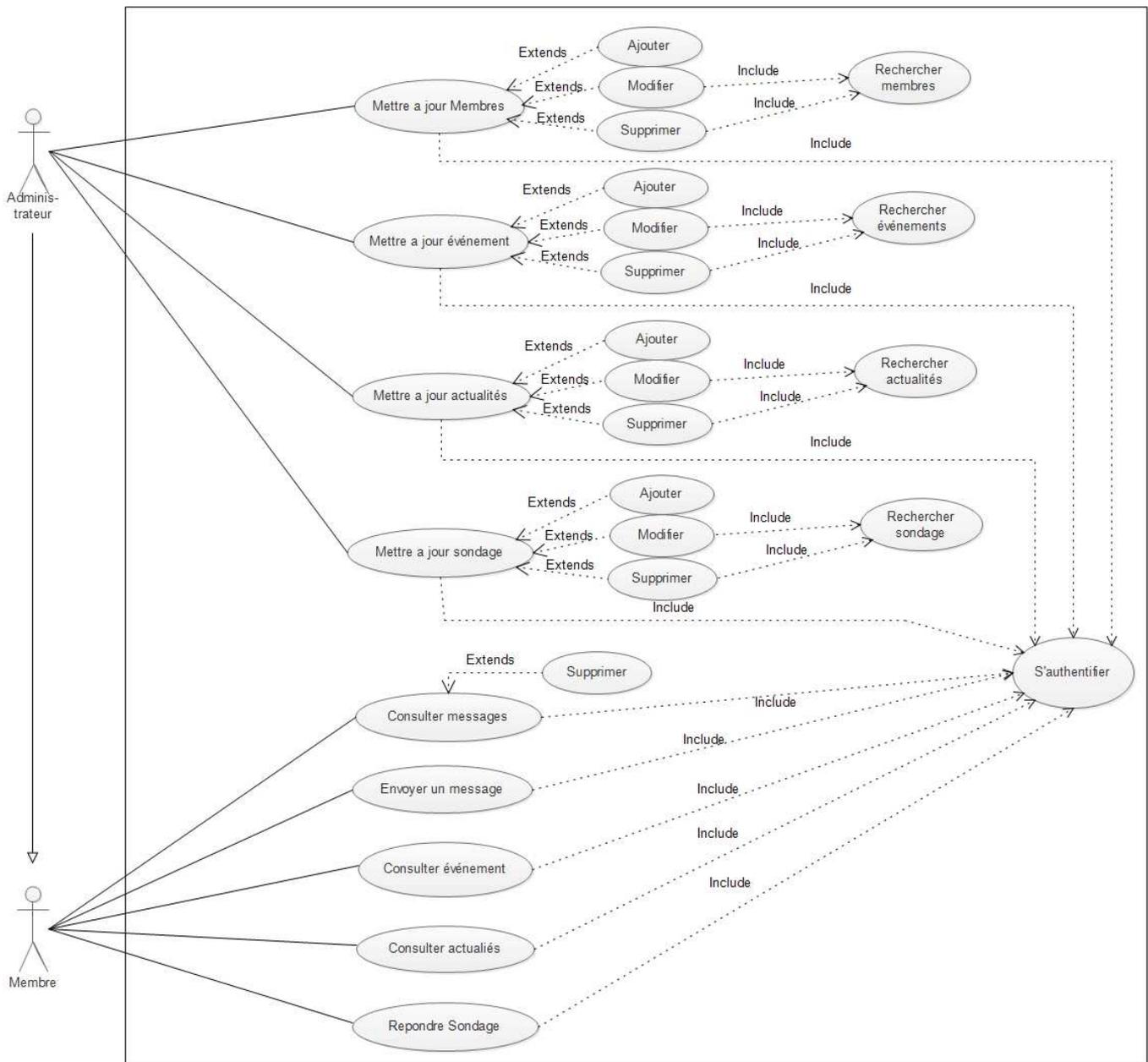


FIGURE 2.3 – Diagramme de cas d'utilisation global :

2.2.6 Description textuelle des cas d'utilisation :

À chaque cas d'utilisation doit être associée une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs [20].

Cas d'utilisation N°1	S'authentifier
Objectif	Vérification de l'identité de l'administrateur ou du membre
Acteur	- Administrateur - Membre
Pré condition	Site accessible
Scénario nominal	[Début] 1. L'utilisateur demande l'accès à la page d'authentification ; 2. Le système affiche la page d'authentification ; 3. L'utilisateur saisit son nom d'utilisateur et son mot de passe ; 4. Le système vérifie la conformité des informations saisit. 5. le système redirige l'utilisateur authentifié vers la page d'accueil. [Fin]
Scénario alternatif	1. Les informations fournies sont incorrectes ; 2. Le système affiche un message d'erreur. 3. On reprend au point 2 du scénario nominal.

TABLE 2.2 – Description textuelle du cas d'utilisation N°1.

Le tableau suivant représente le cas d'utilisation N°2 qui est la mise à jour d'un membre, on note aussi que ce tableau est le même que les cas d'utilisations N°3, 4 et 5 respectivement actualités, évènement et sondage.

Cas d'utilisation N°2		Mettre à jour membre
Ajouter	Objectif	Ajouter un membre
	Acteur	- Administrateur
	Pré condition	Site accessible S'authentifier
	Scénario nominal	[Début] 1. L'administrateur demande l'accès à l'interface de gestion des membres ; 2. Le système affiche l'interface demandée ; 3. L'administrateur choisit d'ajouter un membre ; 4. Le système affiche le formulaire d'ajout ; 5. L'administrateur saisit les données à ajouter ; 6. Le système confirme l'ajout. [Fin]
	Scénario alternatif	. En cas de champs vide affichage d'un message d'erreur. . Si le membre est déjà inscrit retour au scénario nominal 2.
Modifier	Objectif	Modifier le profil d'un membre
	Acteur	- Administrateur
	Pré condition	Site accessible S'authentifier
	Scénario nominal	[Début] 1. L'administrateur choisit de modifier ; 2. Le système affiche le formulaire de modification ; 3. L'administrateur saisit les données à modifier ; 4. Le système confirme la modification. [Fin]
	Scénario alternatif	En cas de saisie de données non compatible aux champs le système affiche un message d'erreur et retourne au scénario nominal 2.
Supprimer	Objectif	Supprimer de la liste un membre
	Acteur	- Administrateur
	Pré condition	Site accessible S'authentifier
	Scénario nominal	[Début] 1. L'administrateur choisit de supprimer ; 2. Le système affiche la demande de confirmation ; 3. L'administrateur confirme ou décline la demande ; 4. Le système exécute le choix de l'administrateur. [Fin]
	Scénario alternatif	Aucun

TABLE 2.3 – Description textuelle du cas d'utilisation N°2.

Cas d'utilisation N°6	Envoyer un message
Objectif	Envoyer des messages entre les membres de l'établissement
Acteur	- Administrateur - Membre
Pré condition	Site accessible S'authentifier
Scénario nominal	[Début] 1. L'utilisateur demande l'accès à la messagerie ; 2. Le système affiche la page d'envoi ; 3. L'utilisateur saisit les destinataires, l'objet et le message ; 4. Le système vérifie l'existence des destinataires ; 5. Le système envoi le message et affiche un message de confirmation. [Fin]
Scénario alternatif	En cas de destinataire inexistant le système retourne au scénario nominal 3.

TABLE 2.4 – Description textuelle du cas d'utilisation N°6.

Cas d'utilisation N°7	Consulter un message
Objectif	Consulter les messages reçus
Acteur	- Administrateur - Membre
Pré condition	Site accessible S'authentifier
Scénario nominal	[Début] 1. L'utilisateur demande l'accès à la messagerie ; 2. Le système affiche la page ; 3. L'utilisateur consulte la liste des messages reçus et choisit le message à lire ; 4. Le système affiche le message. [Fin]
Scénario alternatif	Aucun.

TABLE 2.5 – Description textuelle du cas d'utilisation N°7.

Le tableau suivant représente le cas d'utilisation N°8 qui est consulter

événement, on note aussi que ce tableau est le même pour le cas d'utilisation N°9 consulter actualités.

Cas d'utilisation N°8	Consulter un événement
Objectif	Consulter les événements organisés
Acteur	- Administrateur - Membre
Pré condition	Site accessible S'authentifier
Scénario nominal	[Début] 1. L'utilisateur demande l'accès à la page événement ; 2. Le système affiche la page. [Fin]
Scénario alternatif	Aucun.

TABLE 2.6 – Description textuelle du cas d'utilisation N°8.

Cas d'utilisation N°10	Réponde sondage
Objectif	Répondre au sondage publié par l'administrateur
Acteur	- Administrateur - Membre
Pré condition	Site accessible S'authentifier
Scénario nominal	[Début] 1. L'utilisateur demande l'accès au sondage ; 2. Le système affiche la page ; 3. L'utilisateur choisit une réponse ; 4. Le système affiche un message de confirmation. [Fin]
Scénario alternatif	Aucun.

TABLE 2.7 – Description textuelle du cas d'utilisation N°10.

Le tableau suivant représente le cas d'utilisation N°11 qui est rechercher membres, on note aussi que ce tableau est le même pour le cas d'utilisation 12,13 et

14 qui sont rechercher actualités, événements et sondage.

Cas d'utilisation N°11	Rechercher membres
Objectif	Rechercher un membre
Acteur	- Administrateur
Pré condition	Site accessible S'authentifier
Scénario nominal	[Début] 1. L'administrateur demande la page de mise à jour ; 2. Le système affiche la page ; 3. L'administrateur effectue la recherche ; 4. Le système affiche le résultat. 5. l'administrateur sélectionne le membre. [Fin]
Scénario alternatif	En cas où la recherche ne donne aucun résultat affichage d'un message indiquant cela.

TABLE 2.8 – Description textuelle du cas d'utilisation N°11.

2.3 Diagrammes de séquences

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. Cette représentation peut se réaliser par cas d'utilisation en considérant les différents scénarios associés.[20]

2.3.1 Élaboration des diagrammes de séquence

Le diagramme de séquence suivant représente le cas d'utilisation s'authentifier :

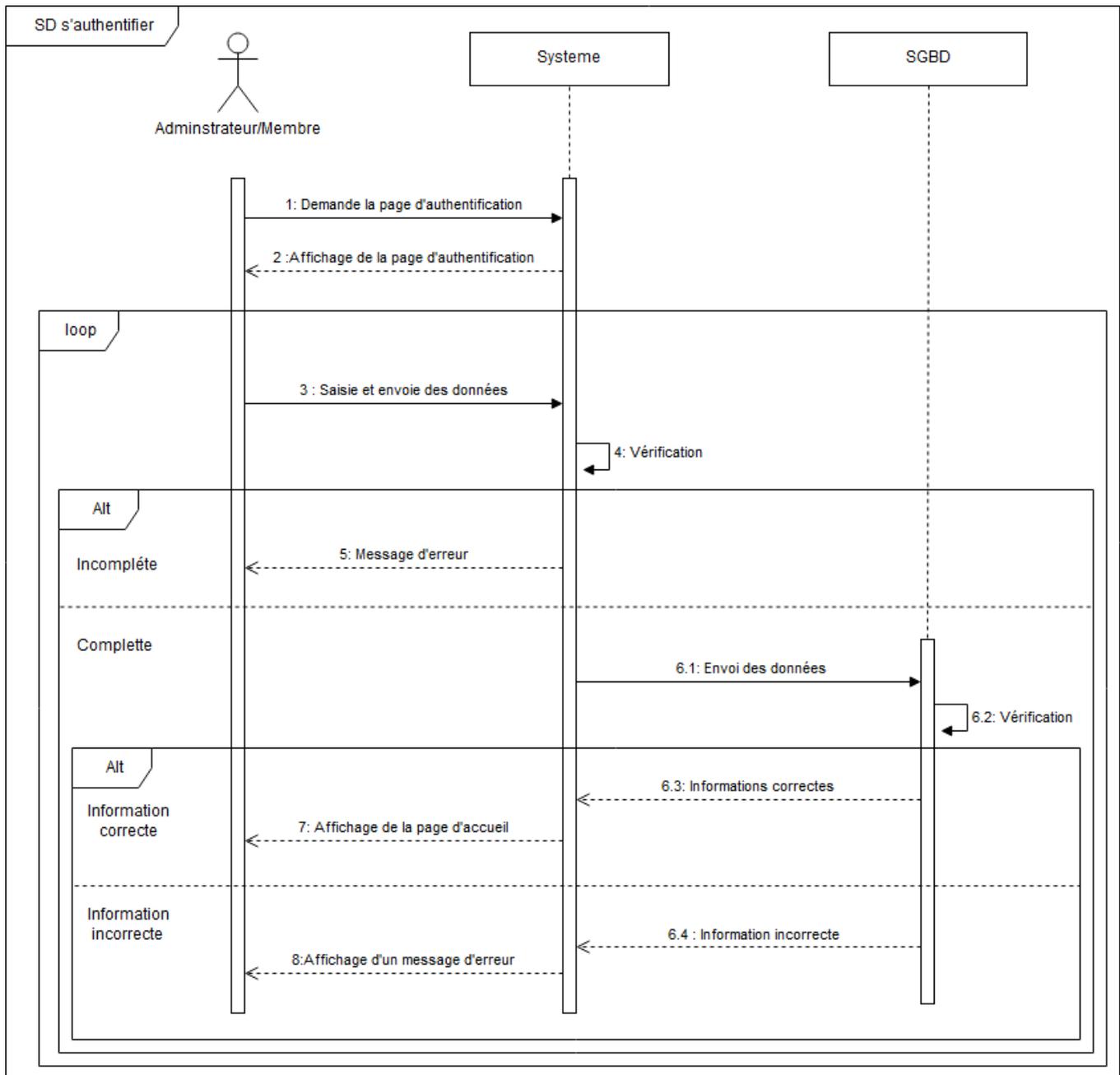


FIGURE 2.4 – Diagramme de séquence du cas s'authentifier

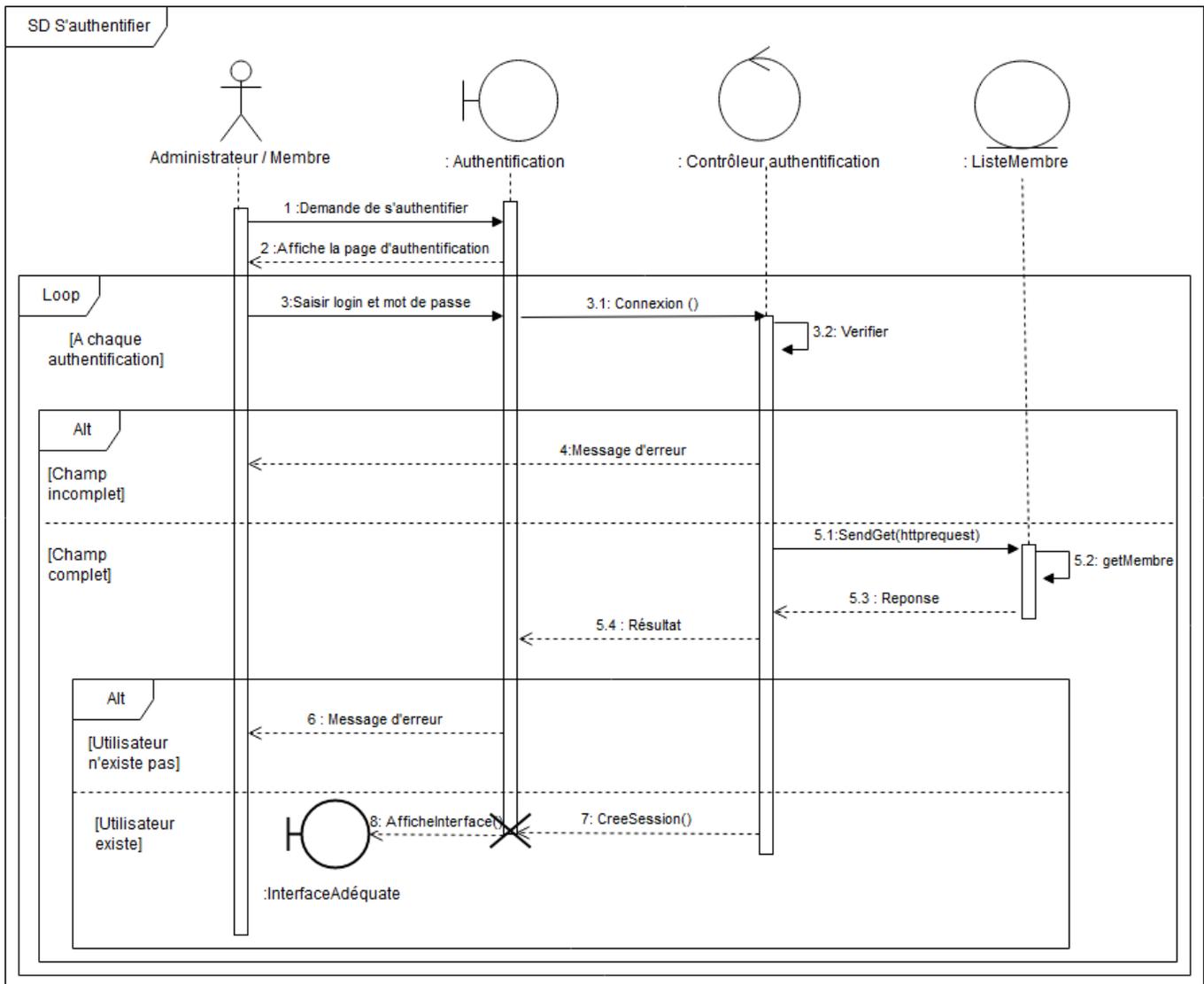


FIGURE 2.5 – Diagramme de séquence approfondie du cas s'authentifier.

2.3.2 Diagramme de séquence envoyer un message

Le diagramme de séquence suivant représente le cas d'utilisation envoyer un message :

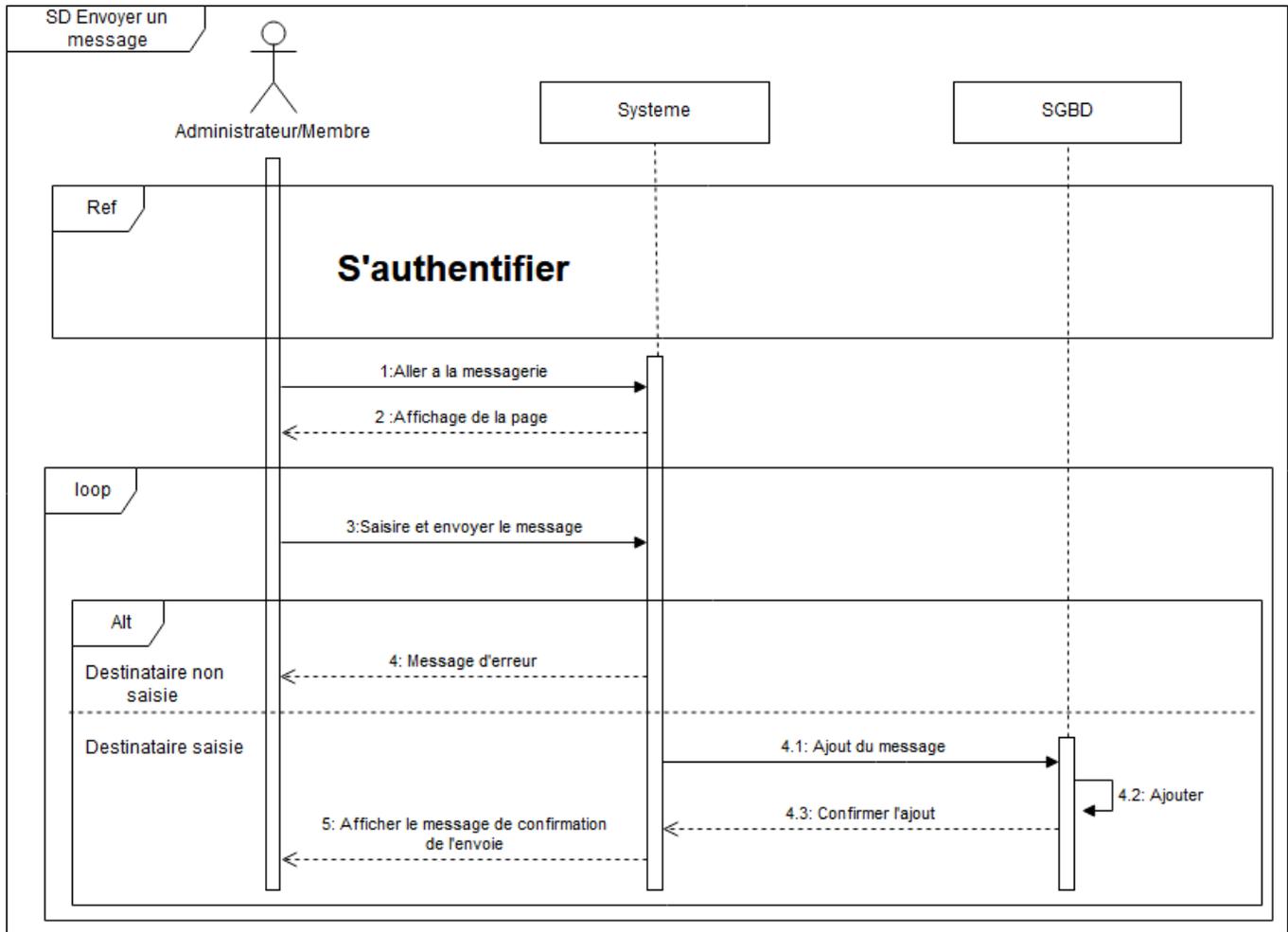


FIGURE 2.6 – Diagramme de séquence du cas envoyer un message.

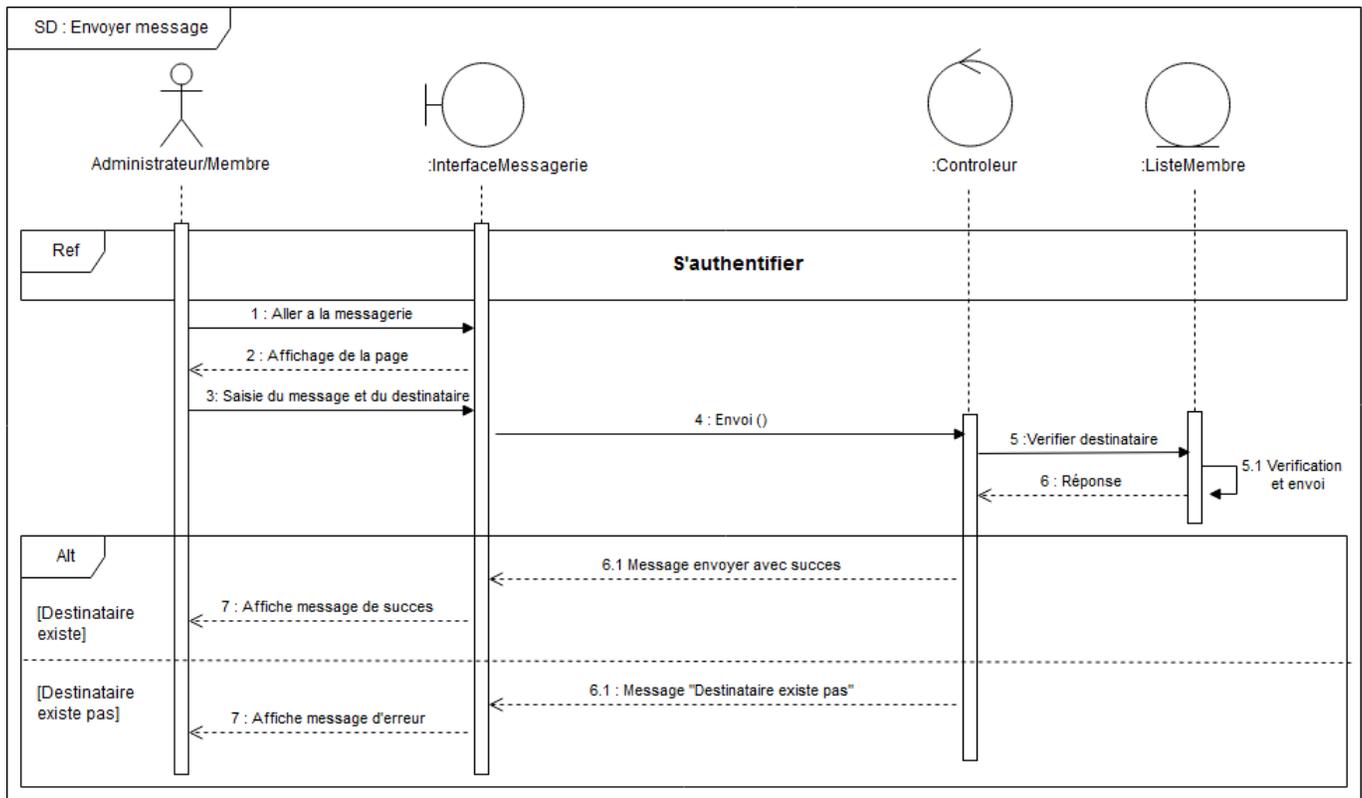


FIGURE 2.7 – Diagramme de séquence approfondie du cas envoyer un message.

2.3.3 Diagramme de séquence Rechercher

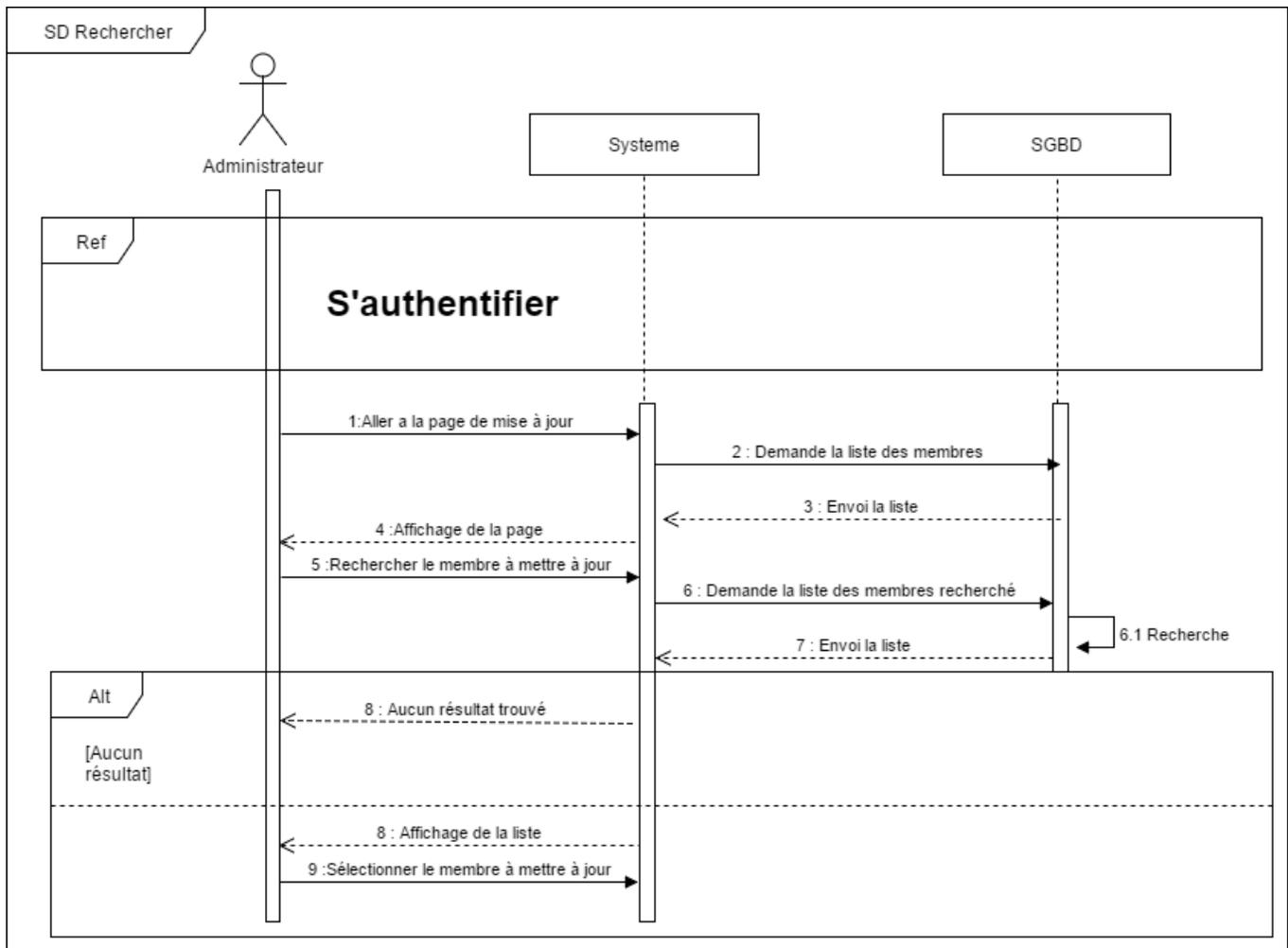


FIGURE 2.8 – Diagramme de séquence Rechercher.

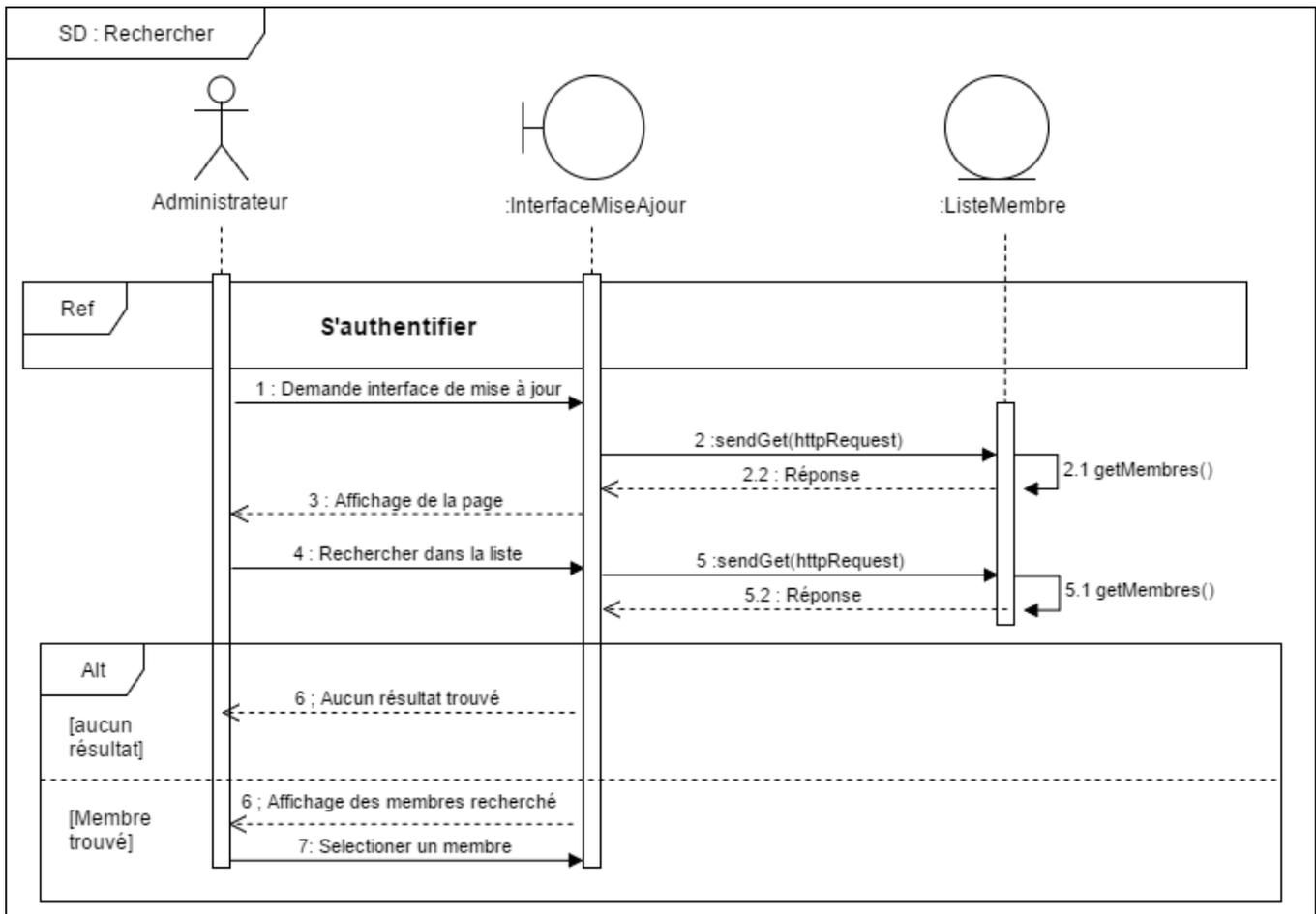


FIGURE 2.9 – Diagramme de approfondie séquence Rechercher.

2.3.4 Diagramme de séquence ajouter un membre

Le diagramme de séquence suivant représente le cas d'utilisation ajouter un membre :

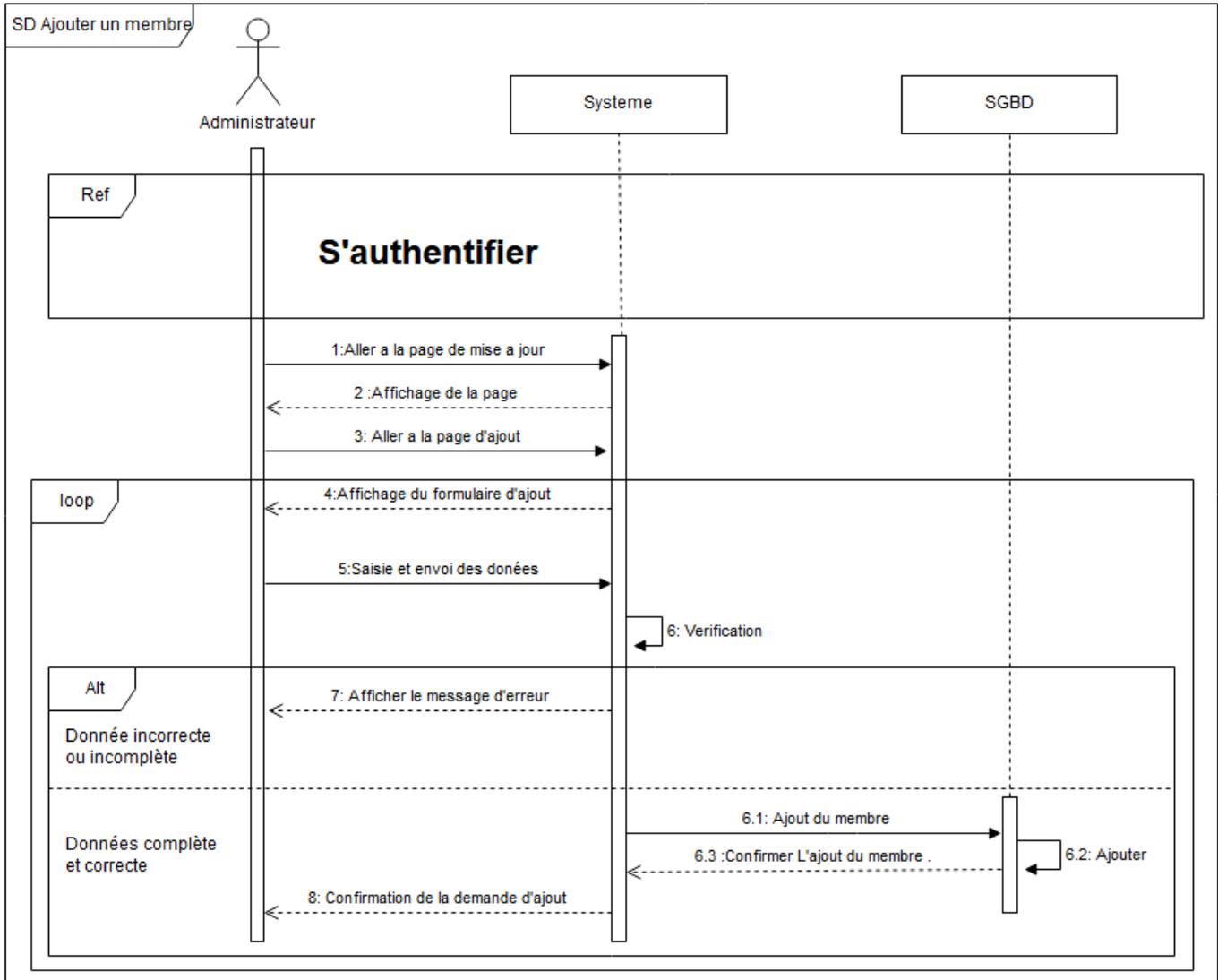


FIGURE 2.10 – Diagramme de séquence du cas ajouter un membre.

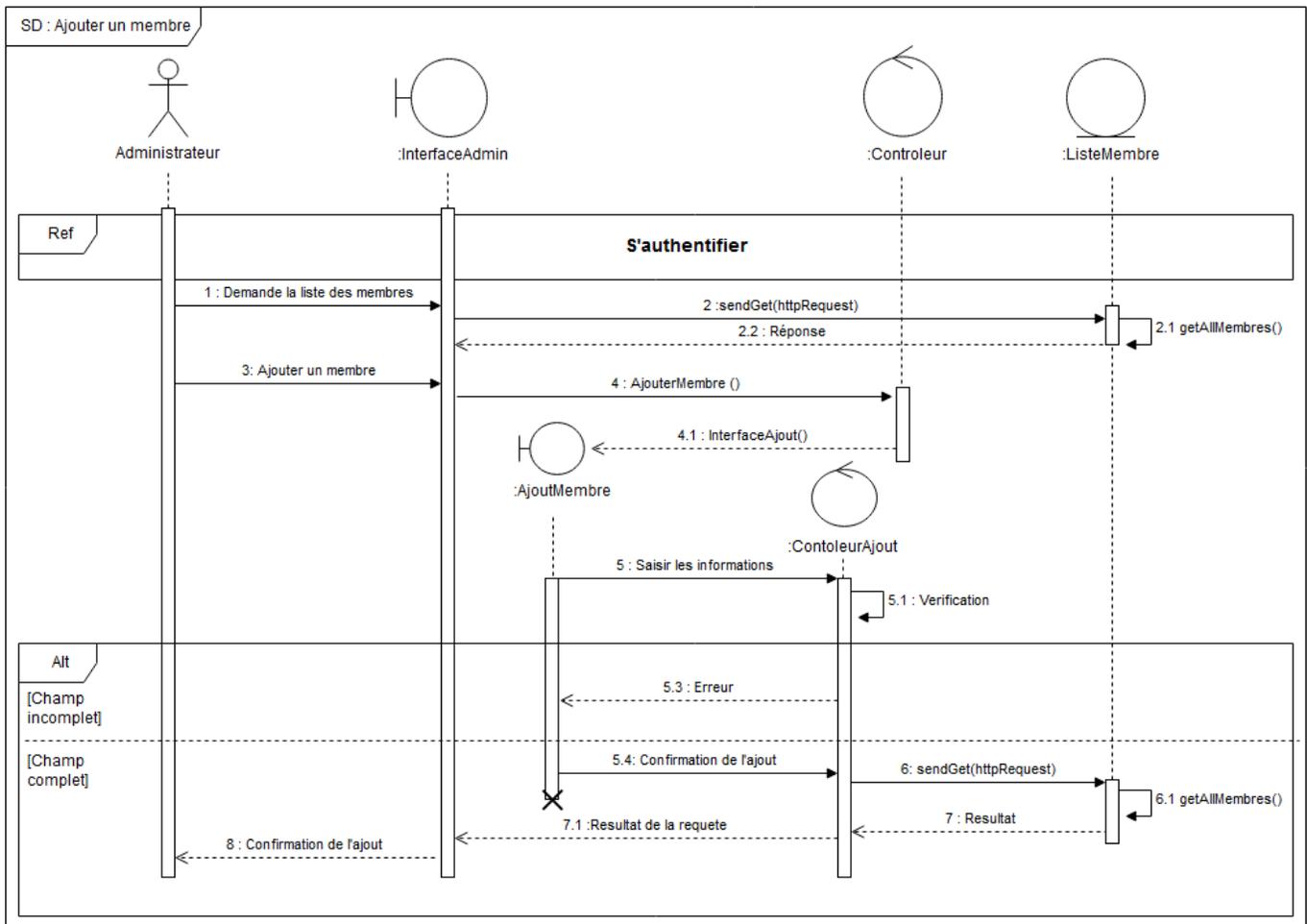


FIGURE 2.11 – Diagramme de séquence approfondie du cas ajouter un membre.

2.3.5 Diagramme de séquence modifier un membre

Le diagramme de séquence suivant représente le cas d'utilisation modifier un membre :

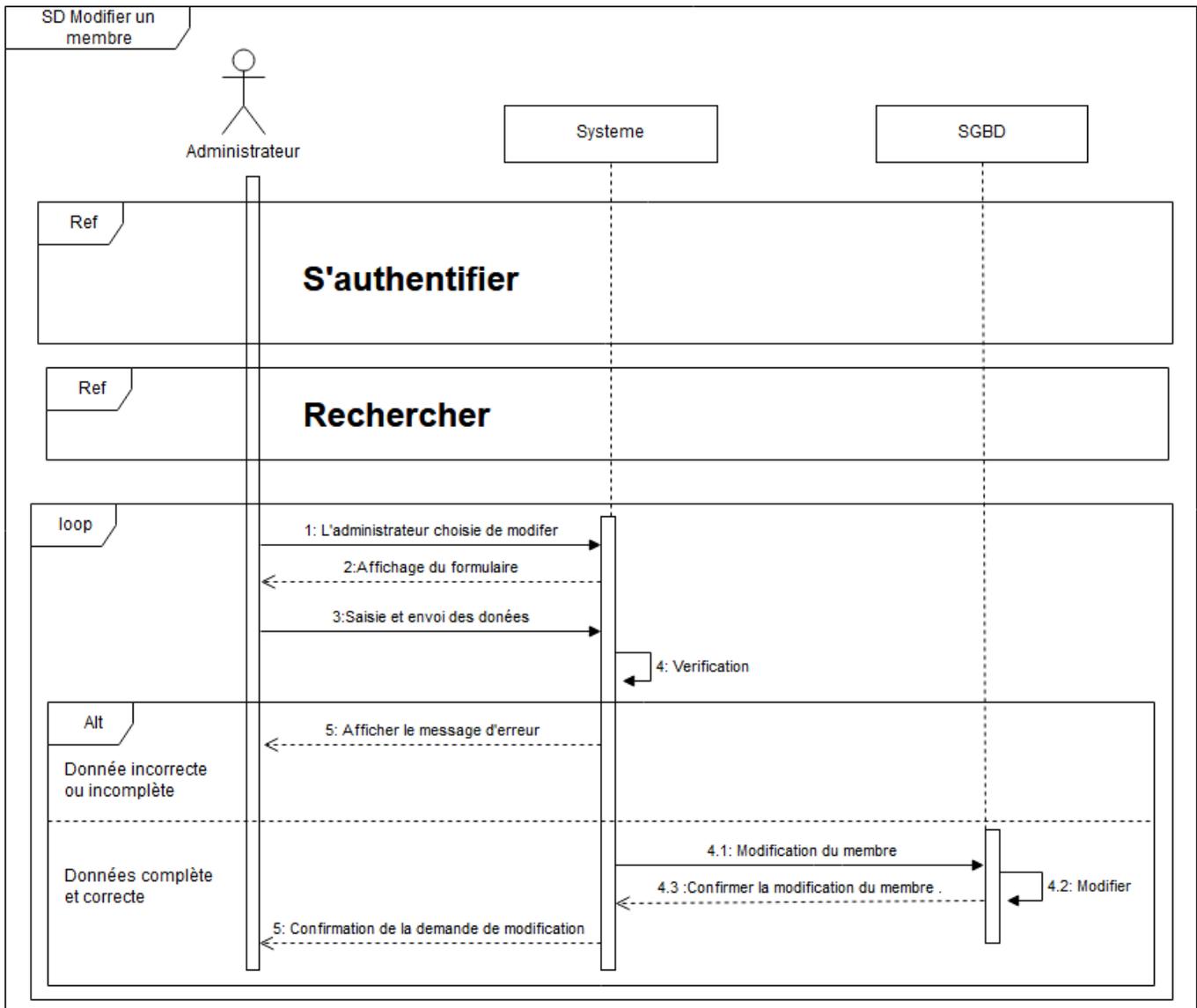


FIGURE 2.12 – Diagramme de séquence du cas modifier un membre.

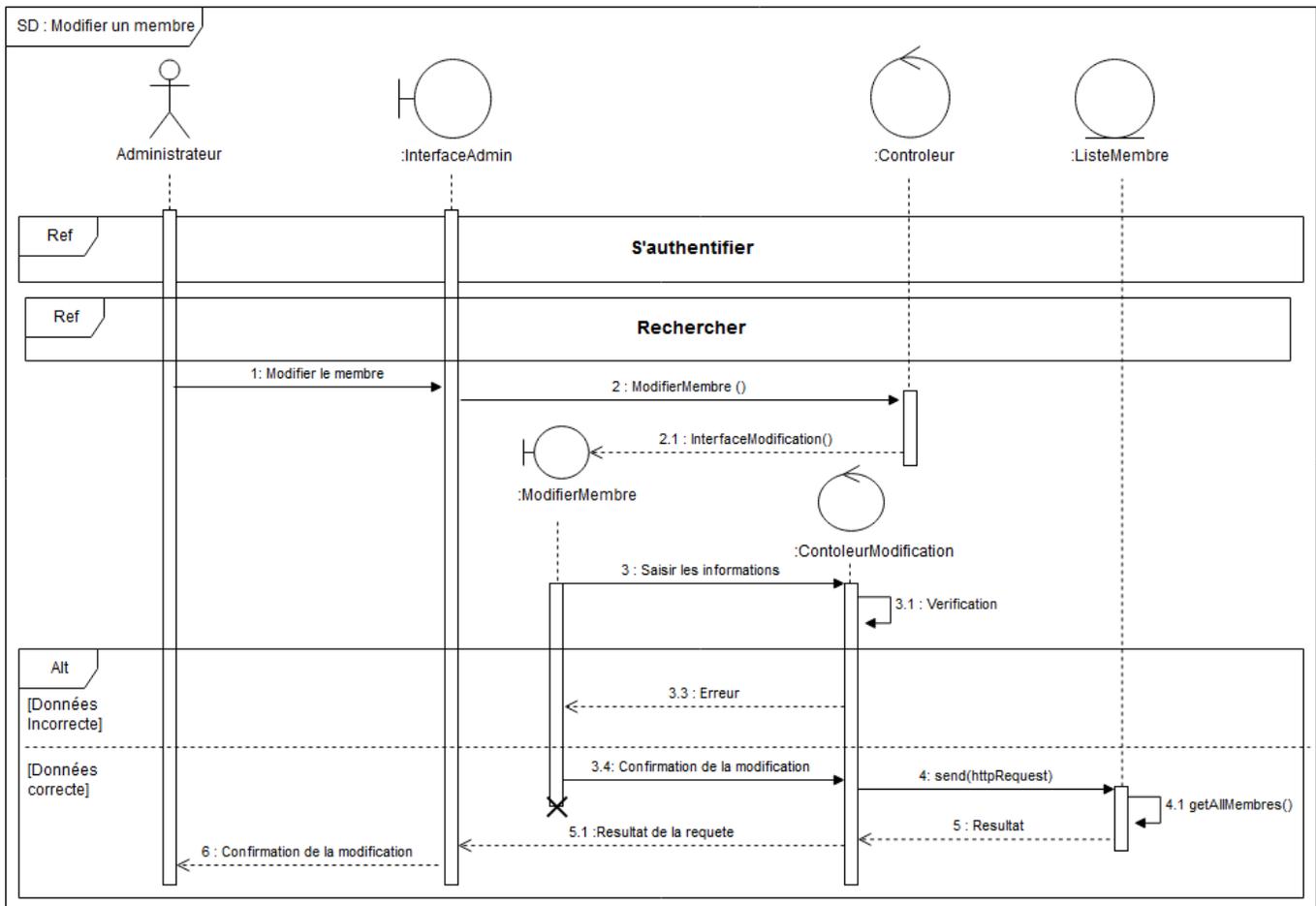


FIGURE 2.13 – Diagramme de séquence approfondie du cas modifier un membre.

2.3.6 Diagramme de séquence supprimer un membre

Le diagramme de séquence suivant représente le cas d'utilisation supprimer un membre :

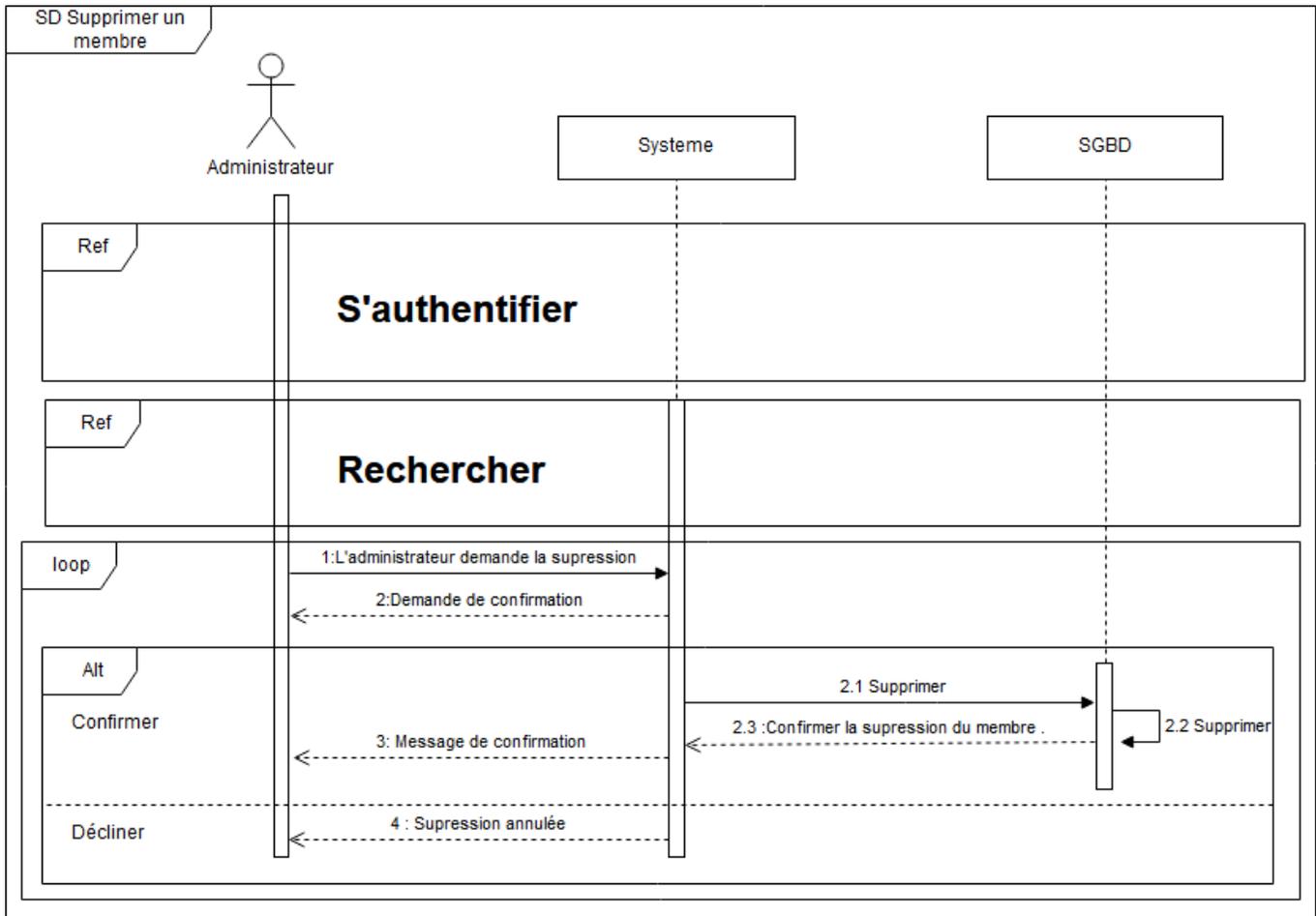


FIGURE 2.14 – Diagramme de séquence du cas supprimer un membre.

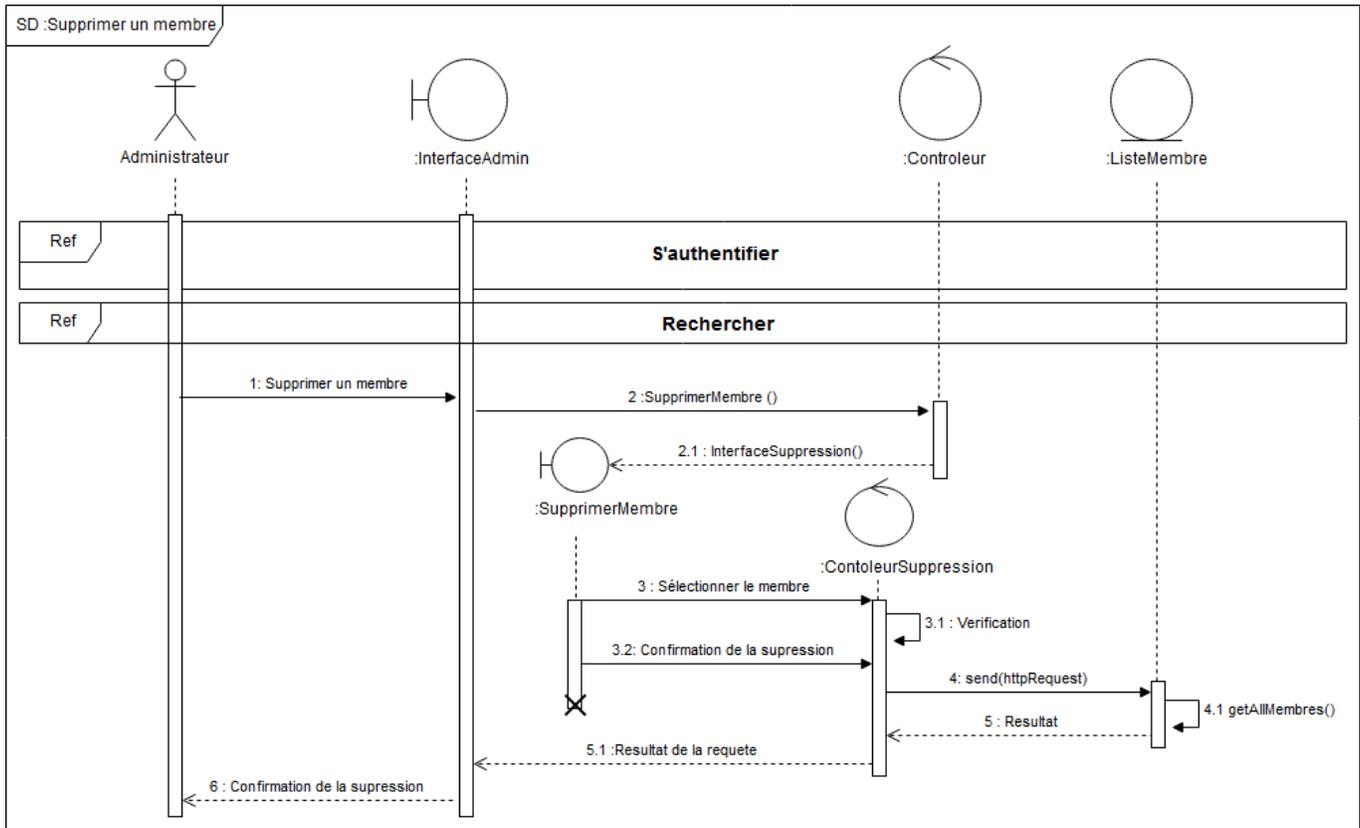


FIGURE 2.15 – Diagramme de séquence approfondie du cas supprimer un membre.

2.4 Présentation du diagramme de classe

Le diagramme de classe constitue l'un des pivots essentiels de la modélisation avec UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations.[20]

2.4.1 Présentation des classes et leurs attributs

Class	Attribut	Définition	Type	Taille
Users	-Id_user	-Identificateur	-Numérique	/
	-Nom	-Nom de l'utilisateur	-Alphabétique	30
	-Prénom	-Prénom de l'utilisateur	-Alphabétique	30
	-E-mail	-E-mail de l'utilisateur	-Alphanumérique	30
	-Password	-Mot de passe du membre	-Numérique	/
	-Adresse	-Adresse de l'utilisateur	-Alphanumérique	100
	-Fonction	-Fonction de l'utilisateur	-Alphabétique	30
Administrateur	-Id_admin	-Identificateur	-Numérique	/
Structure	-Id_struct	-Identificateur de la structure	-Numérique	/
	-Nom	-Nom de la structure	-Alphanumérique	20
Service	-Id_ser	-Identificateur du service	-Numérique	/
	-Nom	-Nom du service	-Alphabétique	20
Message	-Id_msg	-Identificateur du message	-Numérique	/
	-Date envoi	-Date d'envoi du message	-Date	/
	-Body	-Contenu du message	-Alphanumérique	/
Pièce jointe	-Id_pj	-Identificateur de la pièce jointe	-Numérique	/
	-Nom	-Nom du fichier	-Alphanumérique	/
	-File name	-Lien vers la pièce jointe	-Alphanumérique	/
Conversation	-Id_thread	-Identificateur	-Numérique	/
	-Subject	-Sujet	-Alphabétique	50
Actualité	-Id_actu	-Identificateur de la publication	-Numérique	/
	-Titre	-Titre de la publication	-Alphanumérique	/ 50
	-Body	-Contenu de la publication	-Alphanumérique	/
	-Date	-Date de la publication	-Date	/
	-image	-Lien vers l'image de la publication	-Alphanumérique	/
Catégorie	-Id_cat	-Identificateur de la catégorie	-Numérique	/
	-Nom	-Nom de la catégorie	-Alphanumérique	20
Evènement	-Id_event	-Identificateur de l'évènement	-Numérique	/
	-Titre	-Titre de l'évènement	-Alphanumérique	30
	-Body	-Descriptif de l'évènement	-Alphanumérique	/
	-Date_debut	-Date de debut de l'évènement	-Date	/
	-Date_fin	-Date de fin de l'évènement	-Date	/

Sondage	-Id_question	-Identificateur de la question	-Numérique	/
	-Question	-Question du sondage	-Alphanumérique	60
	-MaxCheck	-Nombre d'option	-Alphanumérique	/
	-Statut	-Sondage ouvert ou fermé	-Numérique	/
Option de réponse	-Id_option	-Identificateur de l'option	-Numérique	/
	-Choix	- Réponse du sondage	-Alphanumérique	20
	-nbrVote	-Nombre de fois ou la réponse a été choisie	-Numérique	/
Votes	-Id_vote	-Identificateur du vote	-Numérique	/

TABLE 2.9 – Description des classes et leurs attributs.

Liste des opérations et leur fonction :

- Authentifier :S'authentifier avant d'accéder au site.
- Ajouter : Pour effectuer des ajouts sur la table.
- Modifier : Effectuer des modifications sur la table.
- Supprimer : Supprimer des éléments de la table.
- Consulter : Consulter le contenu de la table.
- Envoyer : Envoyer un message.

2.4.2 Élaboration du diagramme de classes du système

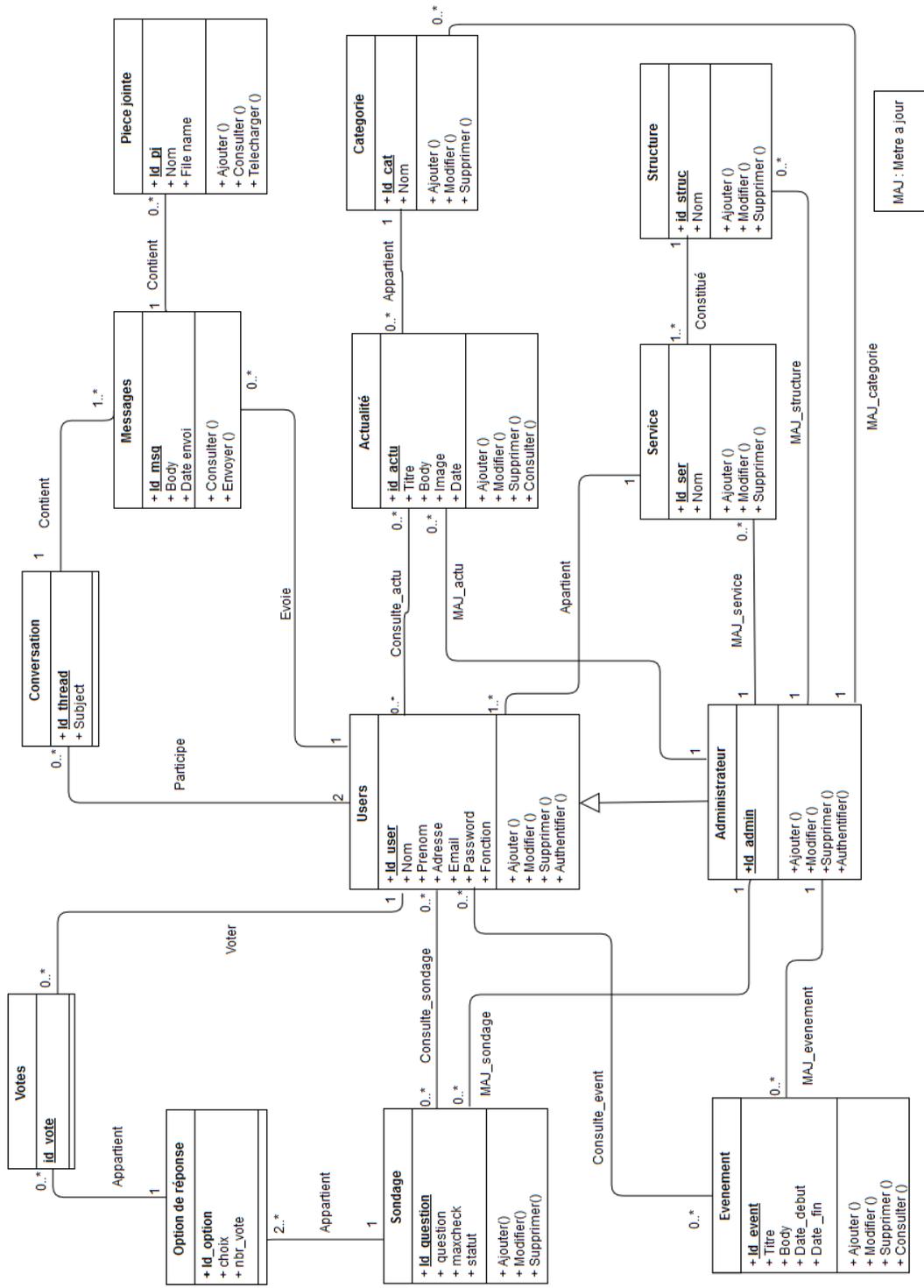


FIGURE 2.16 – Diagramme de classes du système.

2.5 Présentation du diagramme d'activité

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.[6]

2.5.1 Élaboration des diagrammes d'activité

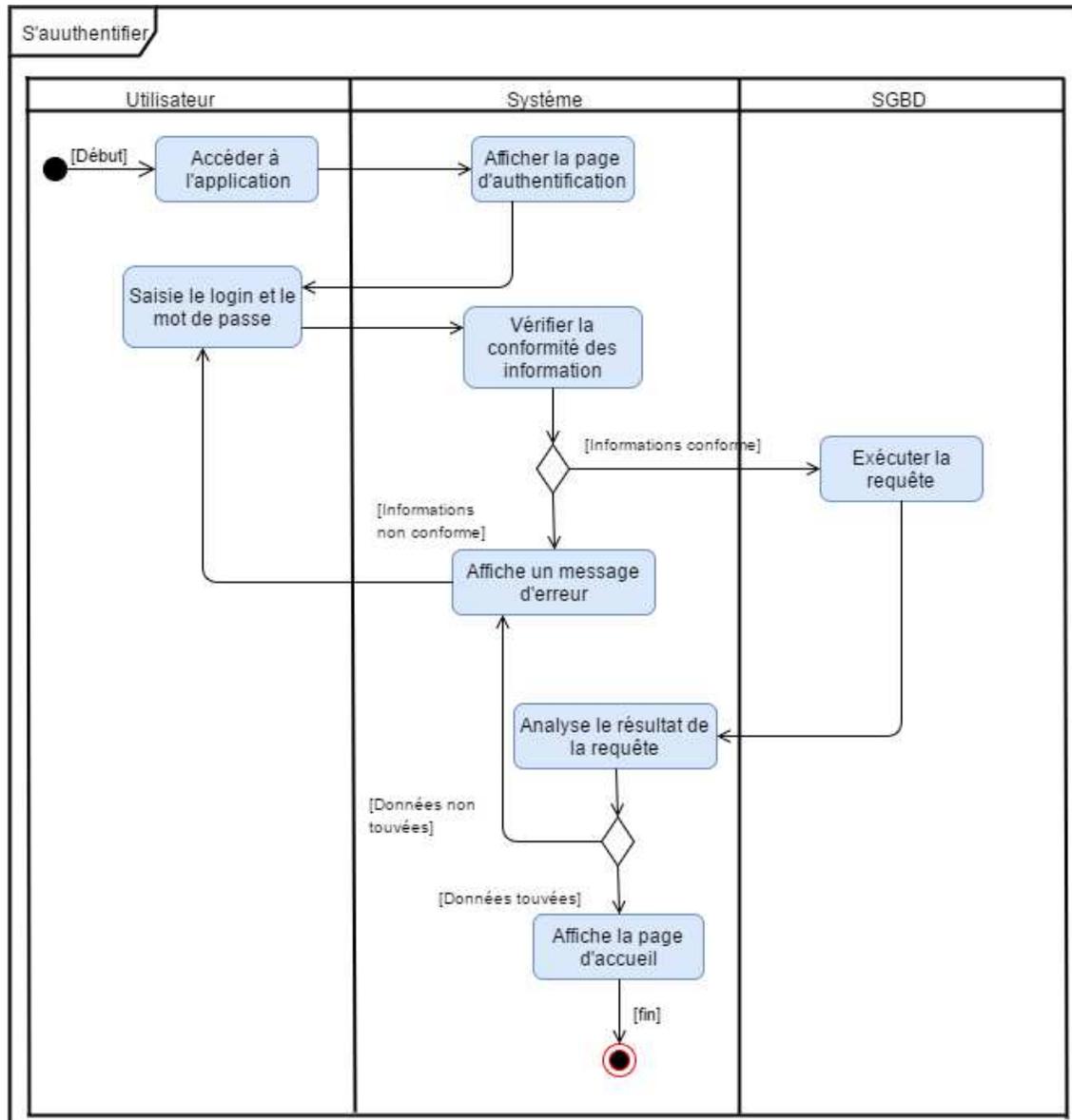


FIGURE 2.17 – Diagramme d'activité s'authentifier.

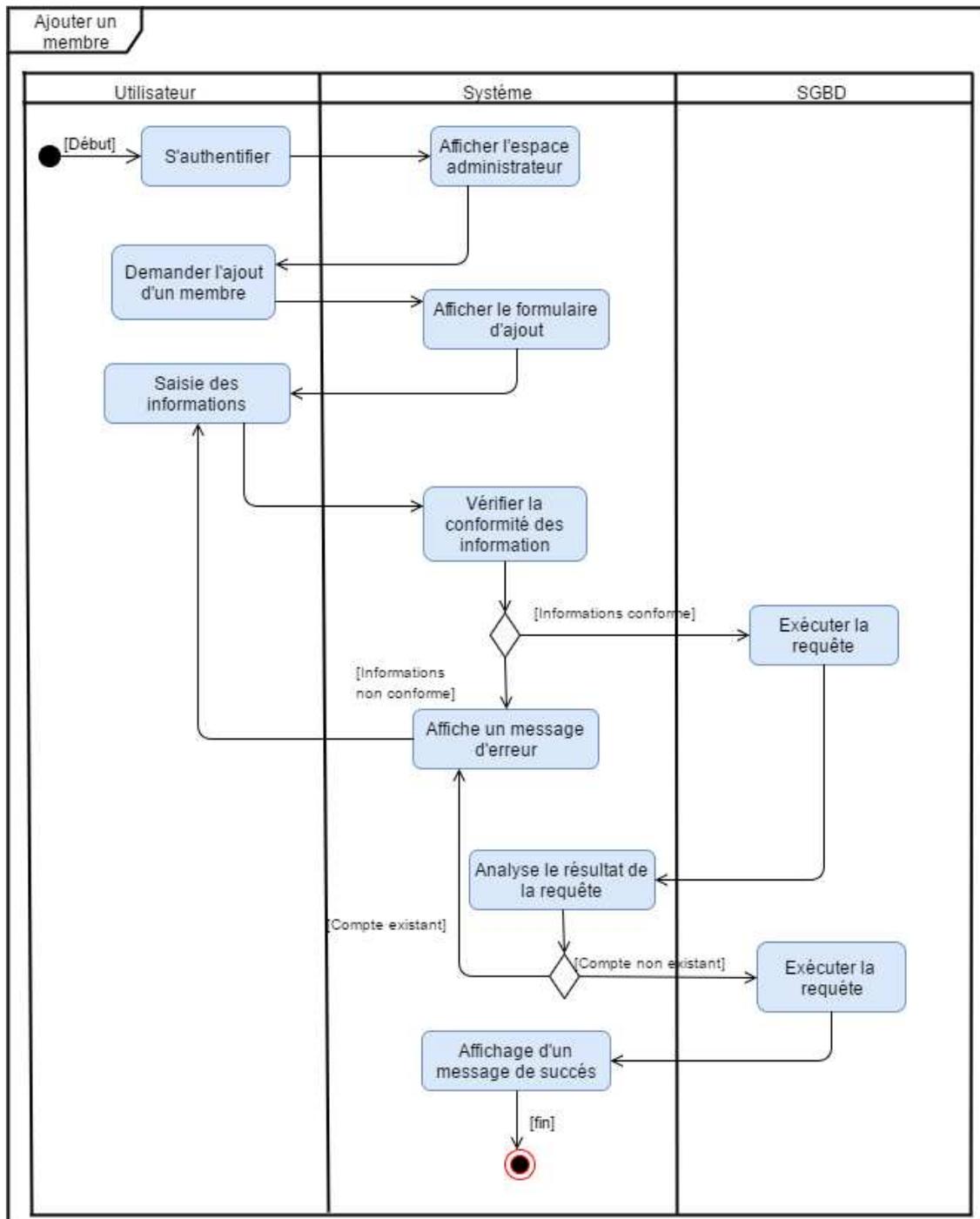


FIGURE 2.18 – Diagramme d'activité ajouter membre.

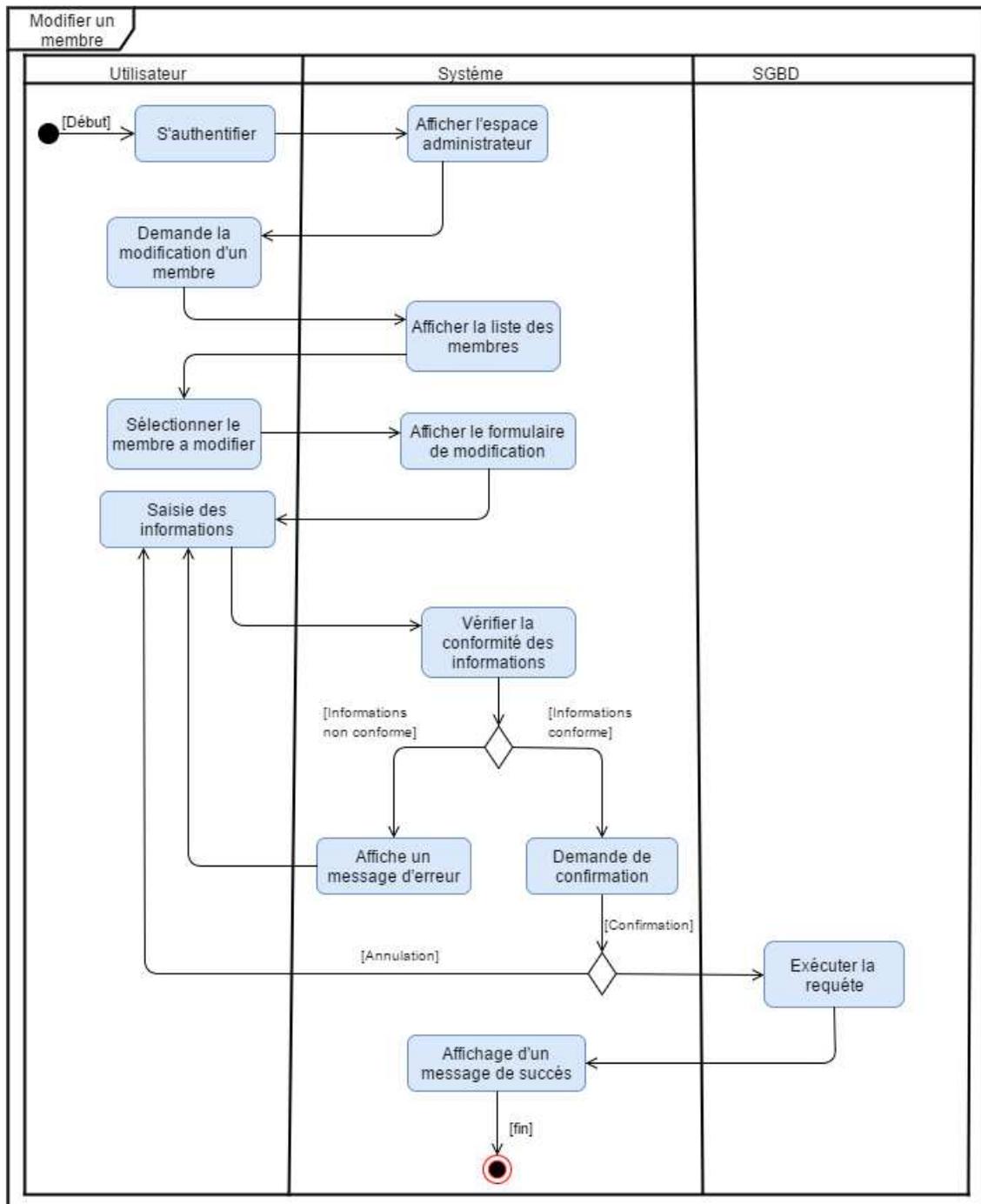


FIGURE 2.19 – Diagramme d’activité modifier membre.

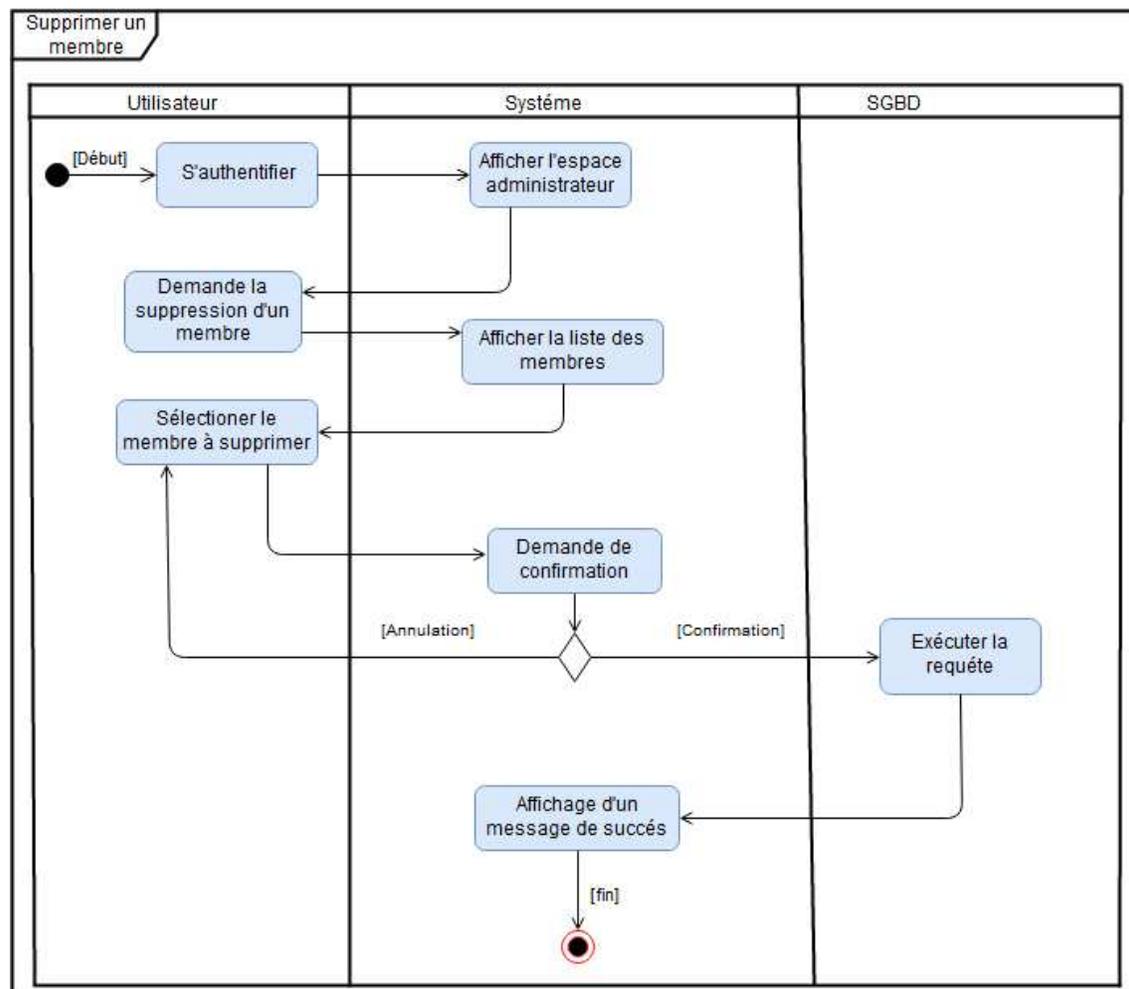


FIGURE 2.20 – Diagramme d'activité supprimer membre.

2.6 Passage au modèle relationnel

Pour passer au modèle relationnel nous allons suivre les trois (3) règles si dessous :

Règle 1 : toute entité devient une relation ayant pour clé primaire son identifiant. Chaque propriété se transforme en attribut.

Règle 2 : toute association hiérarchique (de type $[1, n]$) se traduit par une clé

étrangère. La clé primaire correspondant à l'entité père migre comme clé étrangère dans la relation correspondant à l'entité fils.

Règle 3 : toute association non hiérarchique (de type $[n, n]$ ou de dimension > 2) devient une relation. La clé primaire est formée par la concaténation (juxtaposition) de l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.

Après avoir appliqué les trois règles de passage cité précédemment, nous avons abouti au schéma relationnel de la base de données suivant :

- **Users**(Id_user, Nom, Prenom, Adresse, E-mail, Password, Fonction, #Id_ser),
- **Administrateur**(Id_admin, #Id_user),
- **Message**(Id_msg, Body, Date_envoi, #Id_user, #Id_thread),
- **Piece jointe**(Id_pj, Nom, File_name, #Id_msg),
- **Service**(Id_ser, Nom, #Id_struct, #Id_admin),
- **Structure**(Id_struct, Nom, #Id_admin),
- **Conversation**(Id_thread, Subject, #Id_user),
- **Evenement**(Id_event, Titre, Body, Date_debut, Date_fin, #Id_admin),
- **Actualité**(Id_actu, Titre, Body, Image, #Id_admin, #Id_cat),
- **Categorie**(Id_cat, Nom, #Id_admin),
- **Sondage**(Id_question, question, Maxcheck, statut, #Id_admin),
- **Option de réponse**(Id_option, choix, nbr_vote, #Id_question),
- **Votes**(Id_vote, #Id_option, #Id_user),
- **Consulte sondage**(#Id_user, #Id_question),
- **Consulte actu**(#Id_user, #Id_actu),
- **Consulte event**(#Id_user, #Id_event),

2.7 Conclusion

L'étape de la conception nous a permis de maîtriser les différents diagrammes UML dont nous avons eu recours, avec les quel nous avons pu exprimer les objectifs attendus dans le système à réaliser, ce qui va nous permettre de bien entamer l'étape de la réalisation.

3.1 Introduction

Après la conception de notre système, nous allons passer à sa réalisation en utilisant différents outils de développement, PHP objet comme langage de programmation avec laravel comme Framework, et nous allons présenter une arborescence de l'application ainsi que des interfaces graphiques. Cette étape sera l'étape finale avant le déploiement de l'application.

3.2 Langage de développement

3.2.1 PHP

PHP signifie Personal Home Page, c'est un langage incrusté au HTML et interprété ou compilé côté serveur. Il dérive du C et du Perl dont il reprend la syntaxe. Il est extensible grâce à de nombreux modules et son code source est ouvert. Comme il supporte tous les standards du web et qu'il est gratuit, il s'est rapidement répandu sur la toile.

PHP peut être installé sur les principaux serveurs web du marché. Il permet de générer des pages au format HTML à la demande. Ceci permet d'inclure dans ces pages de l'information issue de bases de données et du système de fichiers. Il permet également récupérer les informations envoyées par le navigateur (méthodes GET et

POST), informations qui peuvent simplement être utilisées par le script PHP ou être stockées pour une utilisation ultérieure.[7]

3.2.2 SQL

Le langage SQL (Structured Query Language) peut être considéré comme le langage d'accès normalisé aux bases de données. Il est aujourd'hui supporté par la plupart des produits commerciaux que ce soit par les systèmes de gestion de bases de données micro tel que Access ou par les produits plus professionnels tels que Oracle. Il a fait l'objet de plusieurs normes ANSI/ISO dont la plus répandue aujourd'hui est la norme SQL2 qui a été définie en 1992.

Le succès du langage SQL est dû essentiellement à sa simplicité et au fait qu'il s'appuie sur le schéma conceptuel pour énoncer des requêtes en laissant le SGBD responsable de la stratégie d'exécution. Le langage SQL propose un langage de requêtes ensembliste et assertionnel.[8]

3.2.3 JavaScript

Langage de développement, utilisant les objets, utilisé dans la conception de pages Web. Javascript a été créé en 1995 par Brendan Eich pour la société Netscape et fut intégré pour la première fois dans le navigateur Netscape Navigator 2.0. Le code Javascript peut être directement intégré dans une page web et est exécuté par le navigateur du côté client.[3]

3.3 Outils de développement

3.3.1 Wampserver

WAMP est une variante de LAMP pour les systèmes Windows et est souvent installé comme un lot de logiciels (Apache, MySQL et PHP).

La partie la plus importante du paquet WAMP est Apache (ou "Serveur HTTP Apache") qui est utilisé pour exécuter le serveur Web. En exécutant un serveur Web

Apache local sur une machine Windows, un développeur web peut tester des pages Web dans un navigateur Web sans les publier directement sur Internet.[9]

3.3.2 Mysql

MySQL est une base de données relationnelle libre qui a vu le jour en 1995 et très employée sur le Web, souvent en association avec PHP (langage) et Apache (serveur web). MySQL fonctionne indifféremment sur tous les systèmes d'exploitation (Windows, Linux, Mac OS notamment).

Le principe d'une base de données relationnelle est d'enregistrer les informations dans des tables, qui représentent des regroupements de données par sujets (table des clients, table des fournisseurs, table des produits par exemple). Les tables sont reliées entre elles par des relations.[10]

3.3.3 Phpstorme

PhpStorm est un environnement de développement (IDE pour integrated development environment) conçu pour PHP, développé par la société JetBrains s.r.o. depuis 2009.

Écrit en Java, il permet d'éditer du code en PHP, en HTML et en JavaScript. Il est compatible aussi bien sur Windows que sur Mac et sur Linux, et il est essentiellement dédié aux codeurs travaillant en PHP.[11]

3.3.4 Sublimtext

Sublime Text est un éditeur de texte générique codé en C++ et Python, disponible sur Windows, Mac et Linux.[12]

3.3.5 Draw.io

Draw.io est une application gratuite en ligne, accessible via son navigateur (protocole https) qui permet de dessiner des diagrammes ou des organigrammes. Cet

outil vous propose de concevoir toutes sortes de diagrammes, de dessins vectoriels, de les enregistrer au format XML puis de les exporter.[13]

3.3.6 Mozilla firefox

Mozilla Firefox est un navigateur web libre et gratuit, développé et distribué par la Mozilla Foundation avec l'aide de milliers de bénévoles grâce aux méthodes de développement du logiciel libre/open source et à la liberté du code source.[14]

3.3.7 Git

Git est un logiciel de gestion de versions décentralisé. C'est un logiciel libre créé par Linus Torvalds, auteur du noyau Linux, et distribué selon les termes de la licence publique générale GNU version 2. En 2016, il s'agit du logiciel de gestion de versions le plus populaire qui est utilisé par plus de douze millions de personnes.[18]

3.4 Framework

Un framework c'est tout simplement un ensemble d'outils venant à faciliter le travail du développeur lui permettant d'obtenir des gains en productivité. Il constitue une base cohérente et regroupe en général les fondations d'un logiciel informatique ou d'une application web.[21]

3.4.1 Laravel

Laravel est un framework PHP qui été créé par Taylor Otwel, qui initie alors une nouvelle façon de concevoir un framework en utilisant ce qui existe de mieux pour chaque fonctionnalité. De fait, il embarque des briques logiciel testé et approuvé par une grande communauté permettant d'améliorer la rapidité des développements et de la robustesse de l'application.[15]

Laravel s'inspire de ce qui existe de mieux, il reprend le système de routage de Symfony en l'étendant de manière efficace, il utilise la bibliothèque SwiftMailer pour l'envoi de mail, et bien d'autres encore. Tout en ajoutant ces propres composants originaux pour former au final Laravel.

Laravel nous offre :

- un système d'envoi de mail,
- un système de routage performant,
- un moteur de template efficace,
- un constructeur de formulaire très pratique,
- un système d'authentification pour les connexions,
- un créateur de requêtes SQL et un ORM performants.[21]

3.4.2 jQuery

jQuery est un framework JavaScript sous licence libre qui permet de faciliter des fonctionnalités communes de JavaScript.

L'utilisation de cette bibliothèque permet de gagner du temps de développement lors de l'interaction sur le code HTML d'une page web, ou la gestion des évènements. jQuery possède par la même occasion l'avantage d'être utilisable sur plusieurs navigateurs web.[16]

3.4.3 Ajax

Asynchronous JavaScript And Xml (AJAX) : désigne un nouveau type de conception de pages Web permettant l'actualisation de certaines données d'une page sans procéder au rechargement total de cette page.

Cette méthode de conception repose sur la combinaison de technologies déjà existantes : HTML/CSS, Javascript/DOM, XML et les requêtes HTTP.

Concrètement, un objet JavaScript permet d'envoyer une requête HTTP au serveur et de récupérer le résultat. La conséquence directe est que l'on peut s'affranchir du rechargement d'une page pour récupérer une information sur le serveur.[17]

3.4.4 Semantic ui

Semantic ui est un framework proposant un ensemble cohérent de composants d'interface. Il permet de simplifier le codage et de rapidement développer des interfaces d'applications, on peut aussi parler de framework front-end.

3.5 Arborescence de l'application

3.5.1 Arborescence côté membre

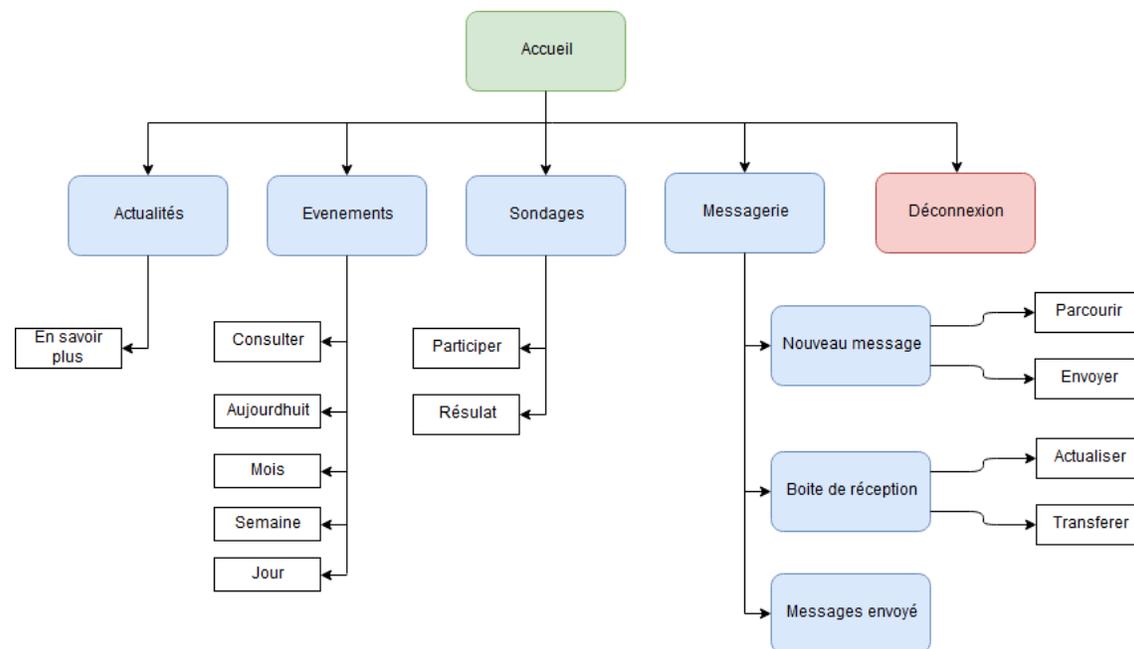


FIGURE 3.1 – Arborescence côté membre.

3.5.2 Arborescence côté administrateur

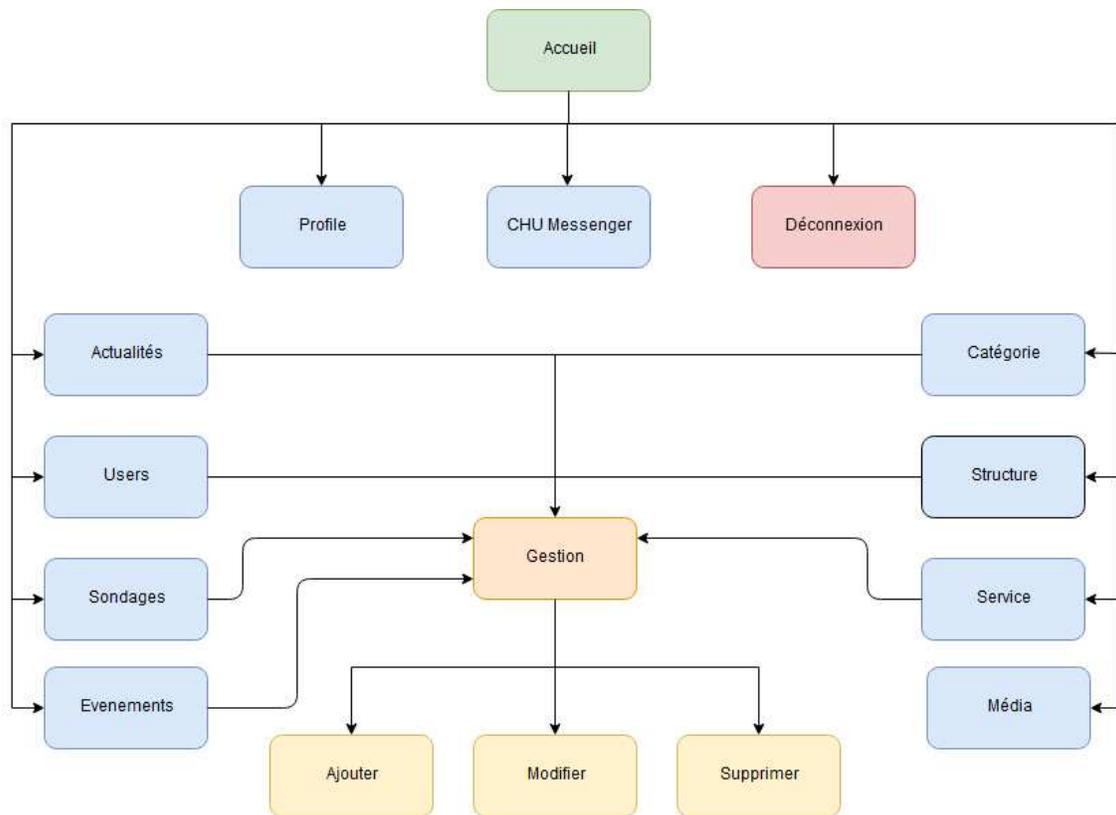
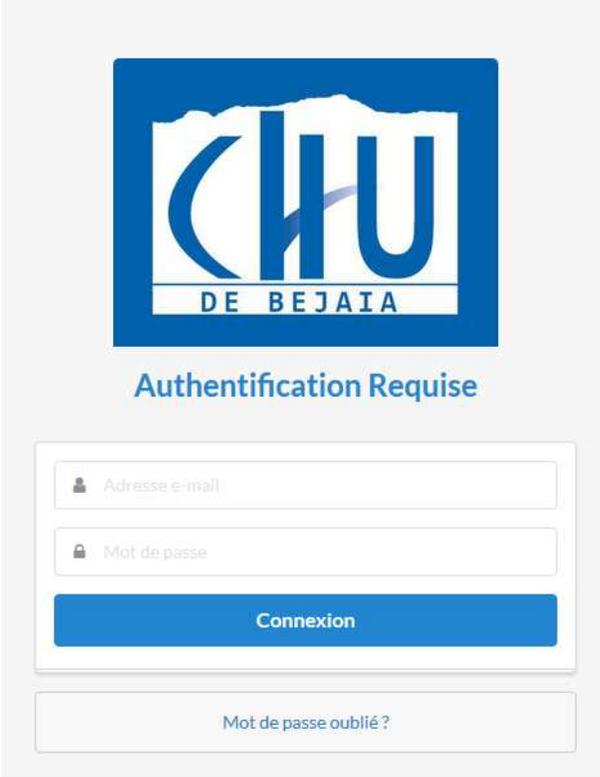


FIGURE 3.2 – Arborescence côté administrateur.

3.6 Présentation des interfaces pour les membres

3.6.1 Interface d'authentification

Avant d'avoir accès à l'application, chaque membre devra s'authentifier à partir de cette page en introduisant son adresse e-mail et son mot de passe.



The image shows a web-based authentication interface. At the top center is the logo for CHU DE BEJAIA, which consists of a blue square containing the letters 'CHU' in a stylized white font, with 'DE BEJAIA' written in smaller white letters below it. Below the logo, the text 'Authentification Requite' is displayed in blue. Underneath, there is a white form with two input fields. The first field is labeled 'Adresse e-mail' and has a person icon on the left. The second field is labeled 'Mot de passe' and has a lock icon on the left. Below these fields is a prominent blue button with the white text 'Connexion'. At the bottom of the form area, there is a link that says 'Mot de passe oublié?' in blue text.

FIGURE 3.3 – Interface d'authentification.

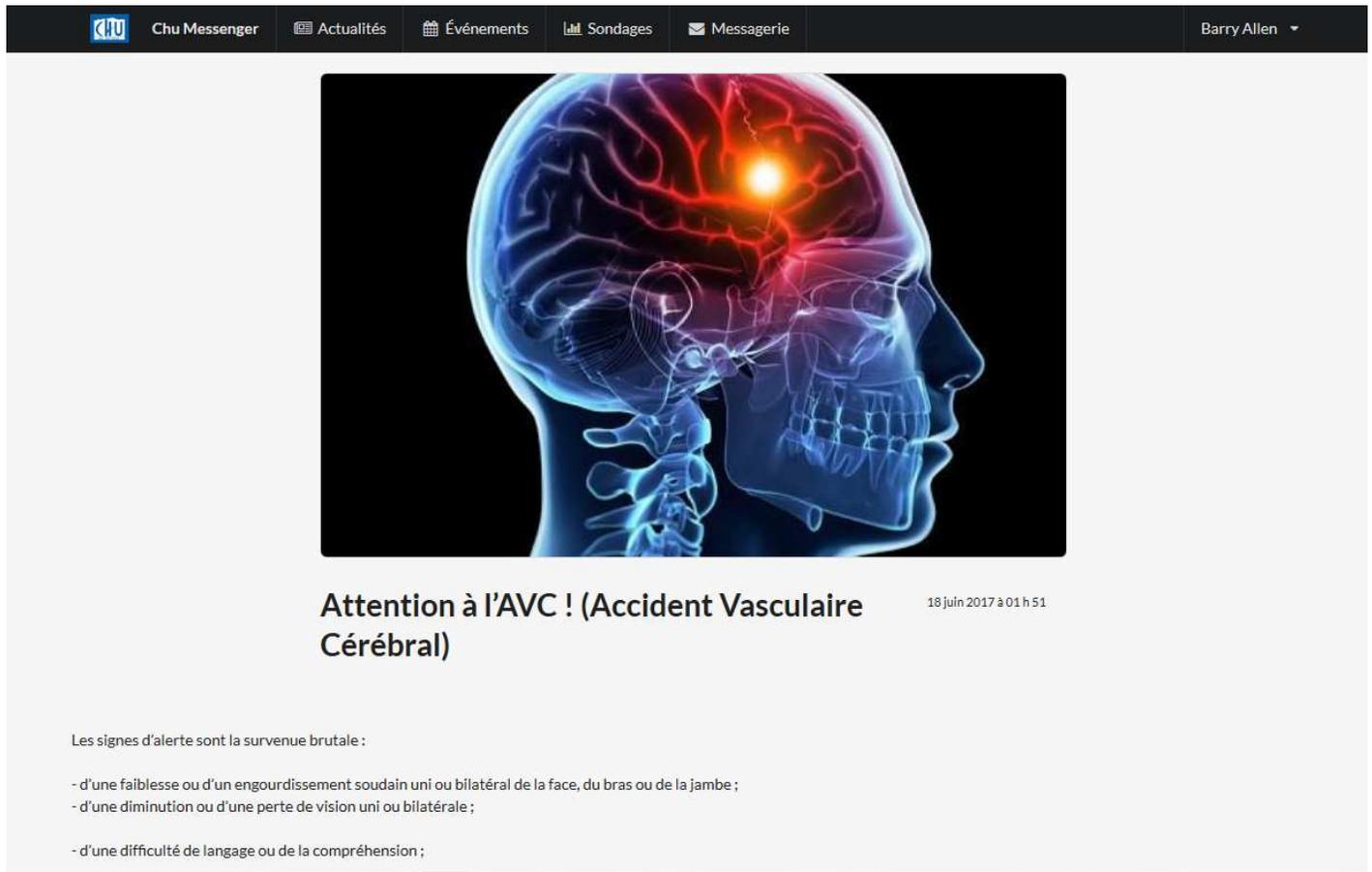
3.6.2 Page d'accueil

Après s'être authentifié le membre sera dirigé vers la page d'accueil où il aura accès au différentes pages de l'application :

FIGURE 3.4 – Page d'accueil.

3.6.3 Affichage des actualités

En cliquant sur le bouton en savoirs plus les membres pourront lire le contenu des actualités comme ci-dessous :



The screenshot shows a web application interface with a dark navigation bar at the top. The navigation bar contains several menu items: 'Chu Messenger' with a logo, 'Actualités' with a calendar icon, 'Événements' with a calendar icon, 'Sondages' with a bar chart icon, and 'Messagerie' with an envelope icon. On the right side of the navigation bar, the user's name 'Barry Allen' is displayed with a dropdown arrow. Below the navigation bar, the main content area features a large image of a human head in profile, rendered in a blue and red wireframe style. The brain is highlighted in red, and a bright orange and yellow glow is visible in the center of the brain, indicating a stroke. Below the image, the title of the article is 'Attention à l'AVC ! (Accident Vasculaire Cérébral)' in bold black text. To the right of the title, the date and time '18 juin 2017 à 01 h 51' are displayed. Below the title, there is a list of symptoms under the heading 'Les signes d'alerte sont la survenue brutale :'. The symptoms listed are: '- d'une faiblesse ou d'un engourdissement soudain uni ou bilatéral de la face, du bras ou de la jambe ;', '- d'une diminution ou d'une perte de vision uni ou bilatérale ;', and '- d'une difficulté de langage ou de la compréhension ;'.

Chu Messenger Actualités Événements Sondages Messagerie Barry Allen



Attention à l'AVC ! (Accident Vasculaire Cérébral) 18 juin 2017 à 01 h 51

Les signes d'alerte sont la survenue brutale :

- d'une faiblesse ou d'un engourdissement soudain uni ou bilatéral de la face, du bras ou de la jambe ;
- d'une diminution ou d'une perte de vision uni ou bilatérale ;
- d'une difficulté de langage ou de la compréhension ;

FIGURE 3.5 – Affichage d'une actualité.

3.6.4 Calendrier des événements

Sur cette page se trouve un calendrier qui contient les événements organisés par le CHU de BEJAIA.

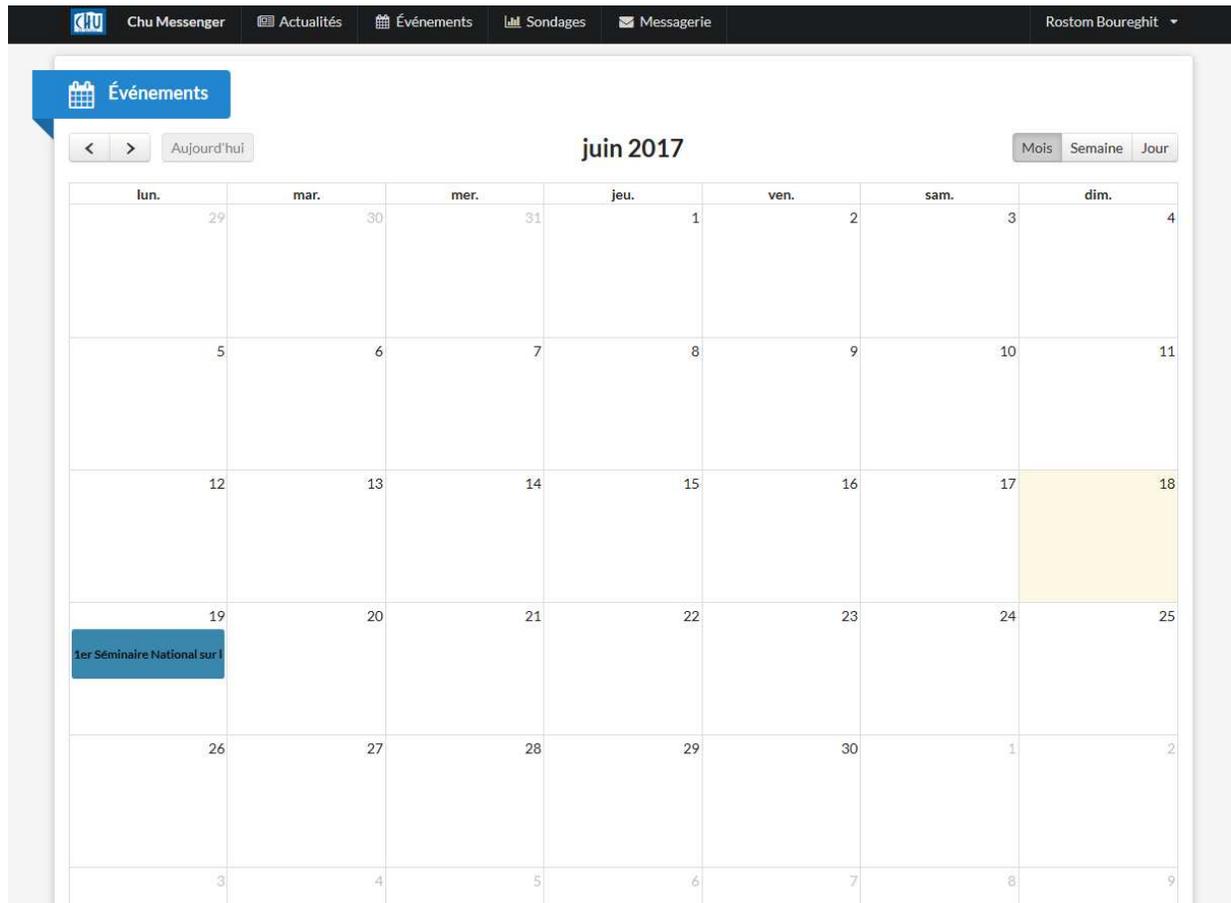


FIGURE 3.6 – Calendrier des événements.

3.6.5 Affichage d'un événement

Après avoir cliqué sur un événement dans le calendrier les informations sur celui-ci s'affichent comme suit :

1er Séminaire National sur l'environnement et l'hygiène dans les établissements de santé.

Sous le Haut Patronage de Monsieur le Ministre de la Santé, de la Population et de la Réforme Hospitalière
Le Centre Hospitalo-Universitaire de Bejaia
Organise

Le **1er** Séminaire National sur l'Environnement et l'hygiène dans les établissements de santé
Le Dimanche 19 février 2017
Auditorium Aboudaou

« Ensemble pour un environnement sain »

Tel : (034) 12.32.36 Email : comchu06@gmail.com
Tel : (034) 12.32.37 CHU de Bejaia@facebook.fr
Fax : (034) 12.33.61 Site : www.chubejaia.dz

Dans le cadre de la mise en œuvre des directives nationales relatives à l'hygiène de l'environnement dans les établissements de santé publics et privés, tracées par la Direction Générale de la Prévention et de la Promotion de la Santé, sous le Haut Patronage de Monsieur le Ministre de la Santé, de la Population et de la Réforme Hospitalière, le CHU de Bejaia abritera le 1er Séminaire National sur le renforcement de l'hygiène de l'environnement dans les établissements de santé publics et privés, le 19 février 2017 au niveau de l'Auditorium Aboudaou- Université de Bejaia.

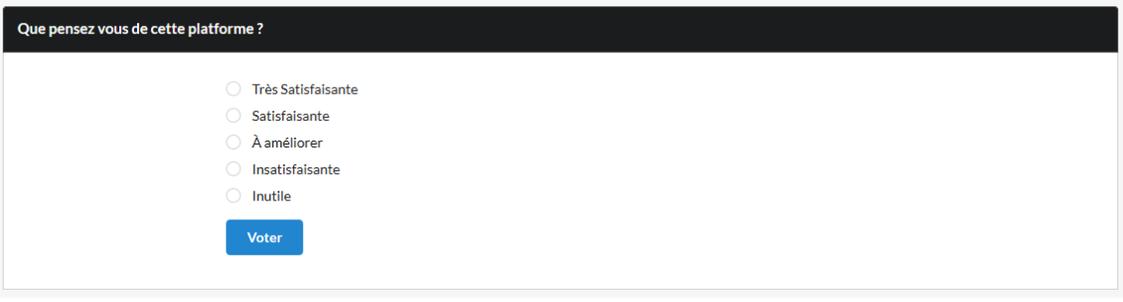
Cette manifestation scientifique vise à sensibiliser et informer l'ensemble du personnel de santé (tout corps confondus) sur les mesures figurant sur le référentiel portant sur l'hygiène de l'environnement dans les établissements de santé publics et privés.

Elle verra la participation d'éminents experts nationaux dans la prévention et la lutte contre les infections nosocomiales.

FIGURE 3.7 – Affichage d'un événement.

3.6.6 Affichage des sondages

En bas de la page d'accueil se trouve le dernier sondage publié avec ces propositions de réponses.



Que pensez vous de cette plateforme ?

- Très Satisfaisante
- Satisfaisante
- À améliorer
- Insatisfaisante
- Inutile

Voter

FIGURE 3.8 – Sondage.

3.6.7 Résultat du sondage

Après avoir répondu au sondage, un résultat s'affiche comme ci-dessous :

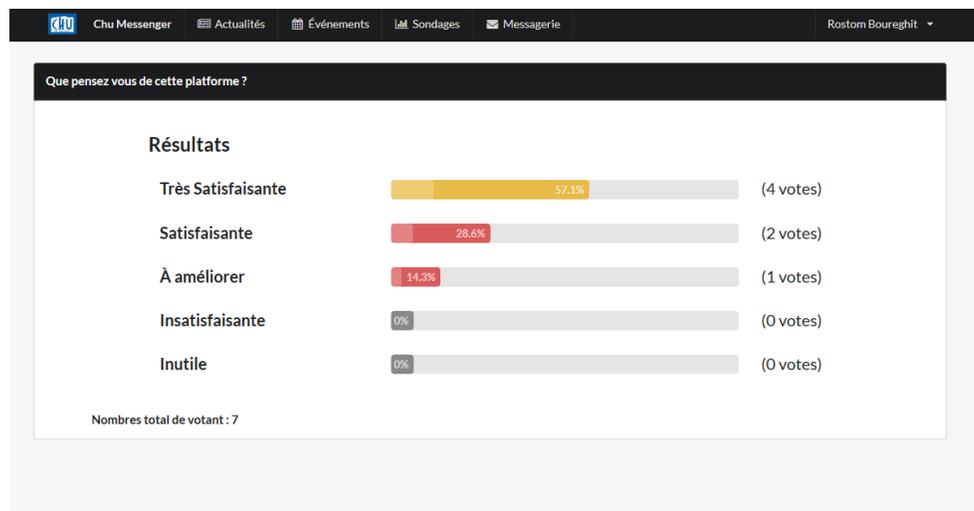


FIGURE 3.9 – Résultat de sondage.

3.6.8 Boite de réception

La boite de réception contient tous les messages reçus.

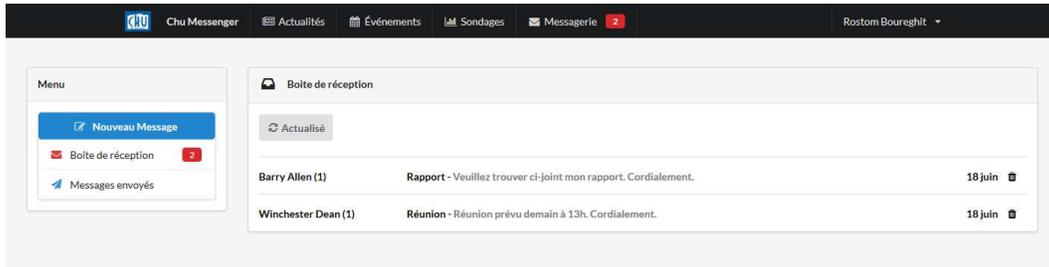


FIGURE 3.10 – Boite de reception.

3.6.9 Nouveau message

Dans cette page on rédige et envoie les messages aux autres membres.

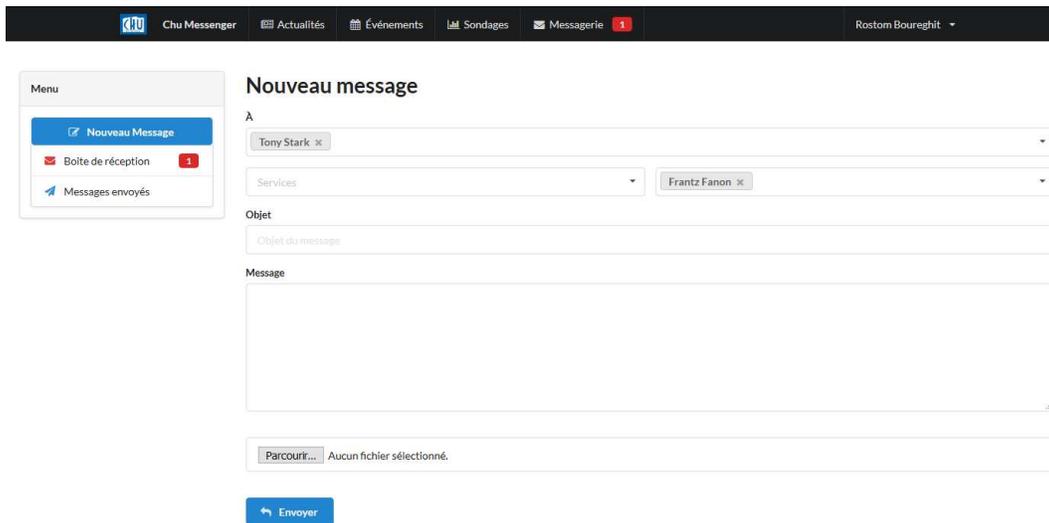


FIGURE 3.11 – Nouveau message.

3.7 Interfaces de l'administrateur

3.7.1 Page de gestion du contenu

Sur cette interface l'administrateur a accès aux pages de gestions de tout le contenu de l'application, de la gestion des membres jusqu'à la publication des sondages.

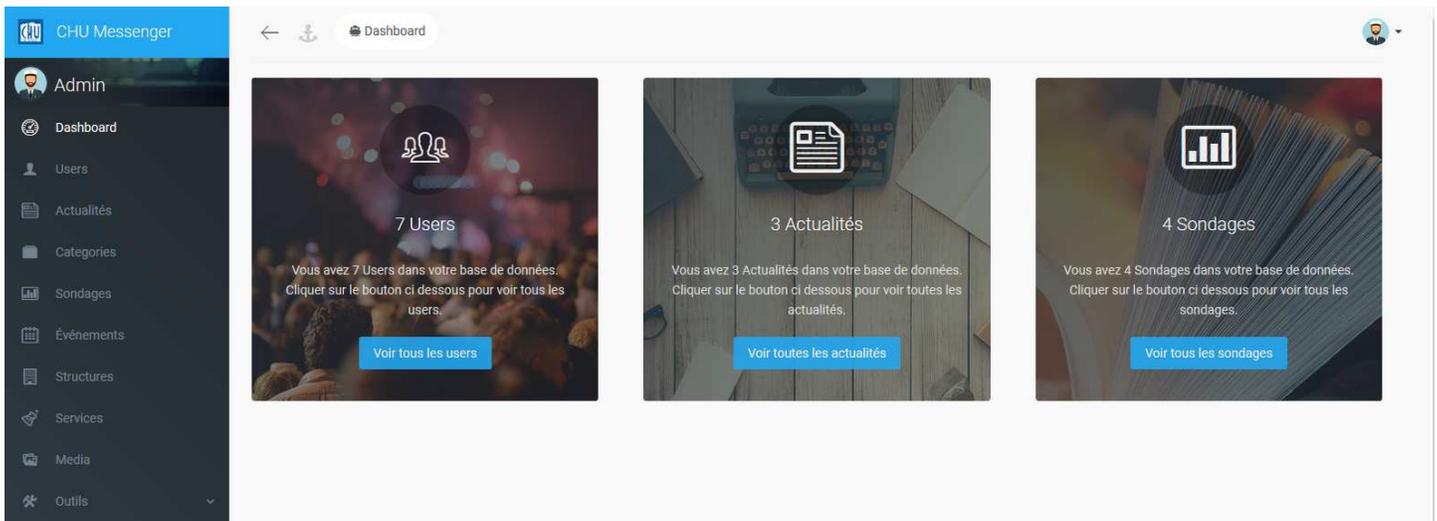


FIGURE 3.12 – Page de gestion du contenu.

3.7.2 Gestion des utilisateurs

Dans la page de gestion des utilisateurs, l'administrateur a la possibilité d'effectuer toutes les modifications sur la liste des membres qui auront accès à l'application.

Users [Ajouter Nouveau](#)

Show 10 entries Search:

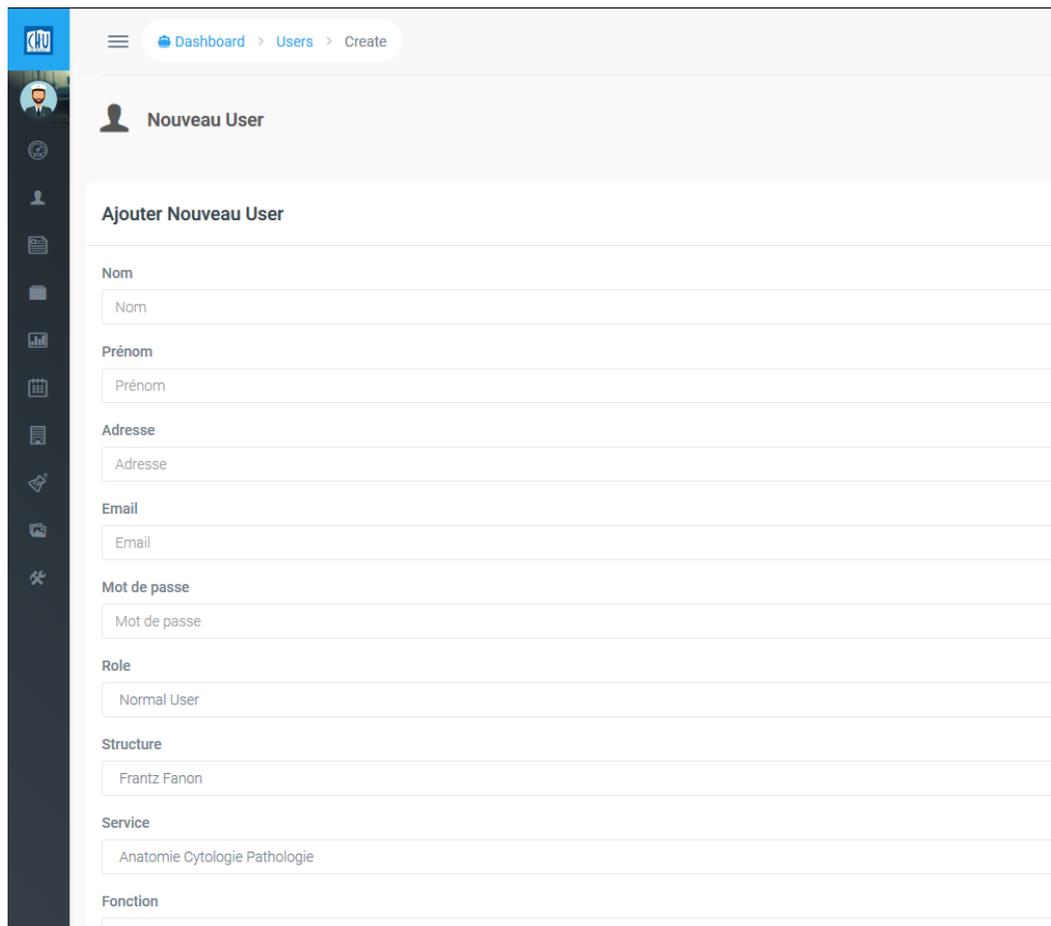
Nom	Prénom	Email	Fonction	Service	Structure	Actions
Boureghit	Rostom	vegapunk93@gmail.com	Développeur	Informatique	Khellil Amrane	Voir Modifier Supprimer
Stark	Tony	stark@gmail.com	Docteur	Chirurgie générale	Khellil Amrane	Voir Modifier Supprimer
Prince	Diana	diana@gmail.com	Docteur	Chirurgie Maxillo Faciale	Frantz Fanon	Voir Modifier Supprimer
Allen	Barry	barry@gmail.com	Docteur	Urgences Medicaux Chirurgicale	Khellil Amrane	Voir Modifier Supprimer
Wayne	Bruce	wayne@gmail.com	Professeur	Bloc opératoire central	Khellil Amrane	Voir Modifier Supprimer
Dean	Winchester	dean@gmail.com	Directeur	Cellule d'accueil et d'orientation des cancéreux	Khellil Amrane	Voir Modifier Supprimer
Admin		admin@admin.com				Voir Modifier Supprimer

Showing 1 to 7 of 7 entries [Previous](#) [1](#) [Next](#)

FIGURE 3.13 – Page de gestion des membres.

3.7.3 Ajouter un membre

Cette interface permet d'inscrire un membre sur l'application en y introduisant toutes les informations nécessaires.



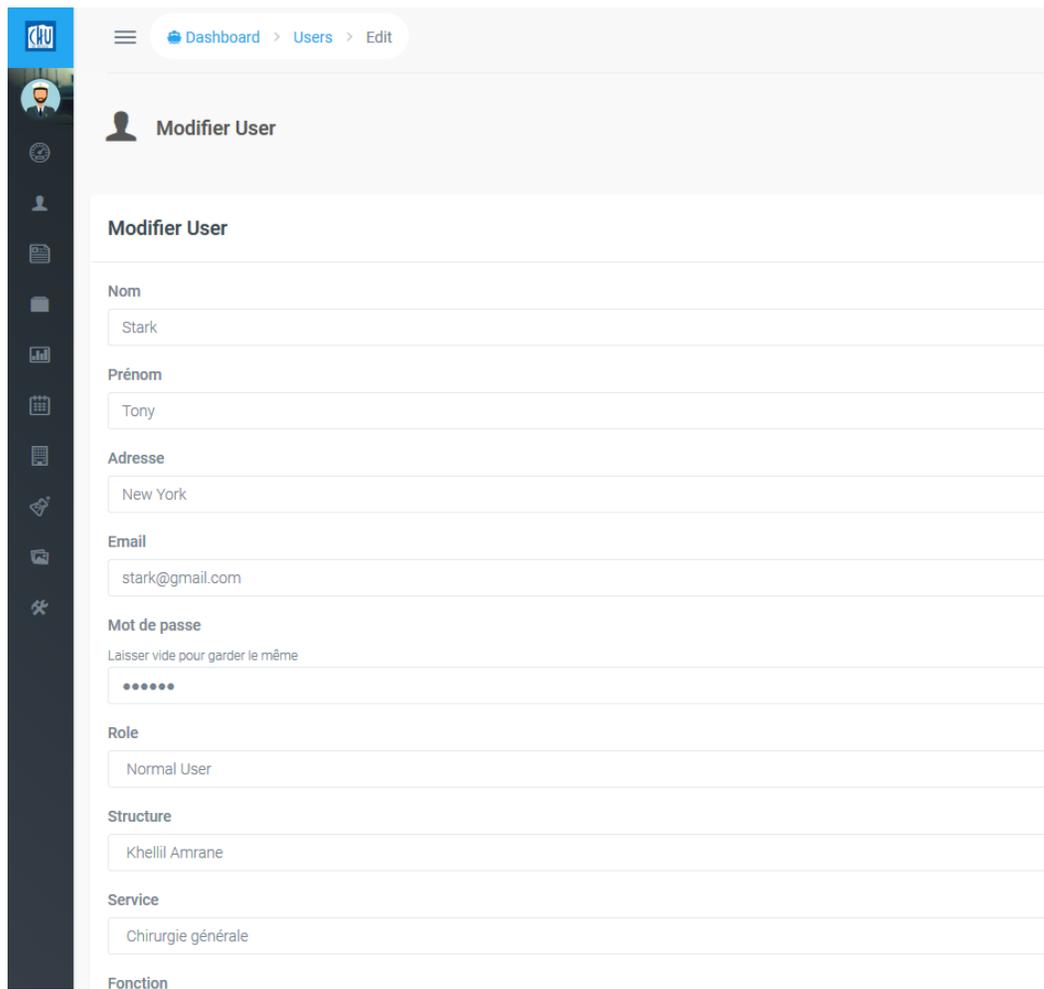
The screenshot shows a web application interface for adding a new user. The breadcrumb navigation at the top reads 'Dashboard > Users > Create'. The page title is 'Nouveau User'. The main heading is 'Ajouter Nouveau User'. The form contains the following fields:

- Nom**: Input field with the placeholder text 'Nom'.
- Prénom**: Input field with the placeholder text 'Prénom'.
- Adresse**: Input field with the placeholder text 'Adresse'.
- Email**: Input field with the placeholder text 'Email'.
- Mot de passe**: Input field with the placeholder text 'Mot de passe'.
- Role**: Input field with the placeholder text 'Normal User'.
- Structure**: Input field with the placeholder text 'Frantz Fanon'.
- Service**: Input field with the placeholder text 'Anatomie Cytologie Pathologie'.
- Fonction**: Input field with the placeholder text '...'.

FIGURE 3.14 – Page ajout de membre.

3.7.4 Modifier un membre

La modification des informations concernant un membre se fait à travers l'interface ci-dessous :



The screenshot displays a web interface for editing a user. On the left is a dark sidebar with a logo and navigation icons. The main content area has a breadcrumb trail: Dashboard > Users > Edit. Below this is a header 'Modifier User' with a user icon. The form contains the following fields:

- Nom:** Stark
- Prénom:** Tony
- Adresse:** New York
- Email:** stark@gmail.com
- Mot de passe:** Laisser vide pour garder le même (displayed as six dots)
- Role:** Normal User
- Structure:** Khellil Amrane
- Service:** Chirurgie générale
- Fonction:** (empty field)

FIGURE 3.15 – Page modification de membre.

3.7.5 Supprimer un membre

La suppression se fait à partir de la liste des membres en cliquant sur le bouton supprimer, un message de confirmation s'affiche comme suit :

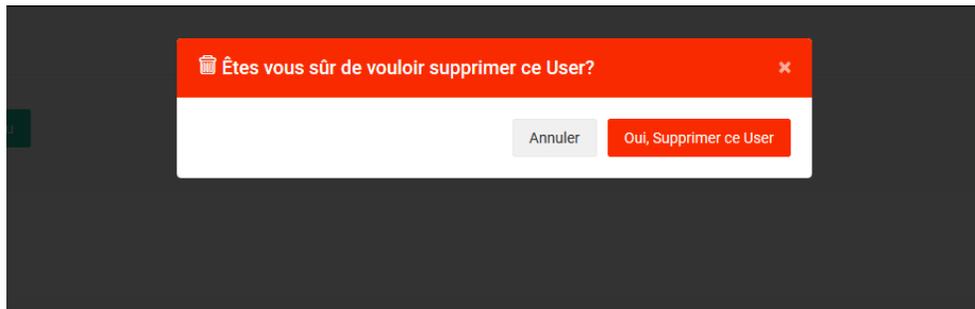


FIGURE 3.16 – Page suppression de membre.

3.8 Diagramme de déploiement

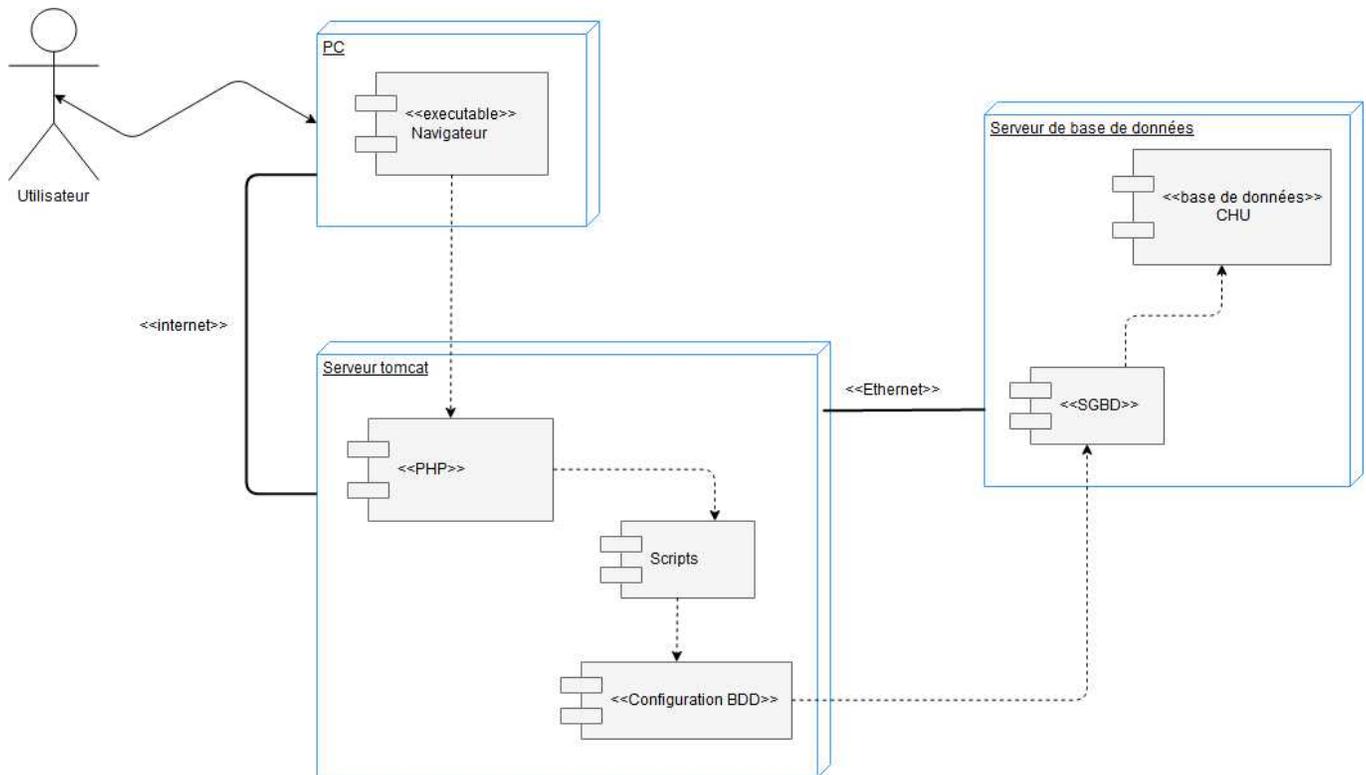


FIGURE 3.17 – Diagramme de déploiement.

3.9 Conclusion

Dans cette phase du cycle de vie de l'application, nous sommes passées de la théorie à la pratique, elle nous a permis d'atteindre nos objectifs décrits dans les étapes précédentes et d'aboutir a un produit utilisable. Nous avons présenté les outils qui nous ont permis de réaliser notre application ainsi qu'une présentation des arborescences et des interfaces de l'application pour bien illustrer le travail qui a été fait.

Conclusion et perspectives

Notre travail a consisté en premier lieu à faire une étude et critique de l'existant dans laquelle nous avons présenté l'organisme d'accueil et effectué une critique de l'existant pour ensuite y proposer une solution, et présenter la méthode de développement le pattern design et l'architecture 3-tiers que nous avons adoptés dans notre projet.

Dans le deuxième chapitre se trouve la conception du projet où nous avons réalisé les diagrammes UML nécessaire pour exprimer clairement les besoins du projet, nous avons eu recours aux diagrammes de cas d'utilisation, de séquence, de classe et d'activités.

Dans le troisième chapitre, nous sommes passés de la théorie à la pratique. Ce chapitre contient tous les outils et langages utilisés pour la réalisation du système, son arborescence et une présentation des interfaces afin de visualiser le travail effectué et avoir une explication sur le fonctionnement de l'application.

Ce travail nous a permis de maîtriser les différentes étapes de la conception d'un logiciel, de comprendre la méthodologie de développement UP et de maîtriser le formalisme UML, nous avons appris à analyser des problèmes et proposer des solutions adéquates afin d'améliorer l'échange d'informations entre les cadres de la gestion du CHU de BEJAIA.

En perspective nous souhaitons développer une version mobile de l'application web afin de permettre une meilleure accessibilité, et aussi développer de nouvelles fonctionnalités et les intégrer dans cette solution tel qu'un système de recherche de message pour la messagerie et un système de notification en temps réel.

Bibliographie

- [1] <https://openclassrooms.com/courses/apprenez-a-programmer-en-java/mieux-structurer-son-code-le-pattern-mvc>, (consulté le 14/06/2017).
- [2] <http://www.chu-bejaia.com>, (consulté le 17/05/2017).
- [3] <http://www.dicofr.com/cgi-bin/n.pl/dicofr/definition/20010101002842>, (consulté le 21/06/2017).
- [4] <http://dico.developpez.com/>, (consulté le 23/05/2017).
- [5] <https://book.cakephp.org>, (consulté le 23/05/2017).
- [6] <http://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-activites>, (consulté le 29/05/2017).
- [7] <http://dico.developpez.com/html/278-Langages-PHP-PHP--Hypertext-Preprocessor.php>, (consulté le 29/05/2017).
- [8] <http://laurent-audibert.developpez.com/Cours-BD/?page=langage-sql>, (consulté le 29/05/2017).
- [9] <https://techterms.com/definition/wamp>, (consulté le 29/05/2017).
- [10] <http://www.mosaique-info.fr/glossaire-web-referencement-infographie-multimedia-informatique-et-multimedia/448-mysql-definition.html>, (consulté le 29/05/2017).
- [11] <https://www.video2brain.com/fr/phpstorm>, (consulté le 29/05/2017).
- [12] https://fr.wikipedia.org/wiki/Sublime_Text, (consulté le 29/05/2017).

-
- [13] <http://www.tice-education.fr/index.php/tous-les-articles-et-ressources/articles-internet/819-draw-io-un-outil-pour-dessiner-des-diagrammes-en-ligne>, (consulté le 29/05/2017).
- [14] https://fr.wikipedia.org/wiki/Mozilla_Firefox, (consulté le 29/05/2017).
- [15] <https://www.supinfo.com/articles/single/3343-presentation-framework-php-laravel>, (consulté le 29/05/2017).
- [16] <http://glossaire.infowebmaster.fr/jquery/>, (consulté le 29/05/2017).
- [17] <http://www.futura-sciences.com/tech/definitions/tech-ajax-3998/>, (consulté le 29/05/2017).
- [18] <https://git-scm.com/>, (consulté le 29/05/2017).
- [19] <http://www.commentcamarche.net/contents/221-reseaux-architecture-client-serveur-a> (consulté le 5/05/2017).
- [20] D. Gabay J. Gabay. *UML 2 Analyse et conception, mise en oeuvre guidée avec études de cas*. DUNOD, 2008.
- [21] M.Chavelli. *Découvrez le framework php laravel*. EYROLLES, 2016.
- [22] F.Vallée P.Roques. *Processus de développement UML par action de l'analyse des besoins a la conception*. EYROLLES, 2004.

RÉSUMÉ

Dans ce mémoire, nous avons conçu et réalisé une messagerie interne professionnelle pour le CHU de BEJAIA dans le but d'aider à améliorer la communication et l'échange d'informations via la plateforme de messagerie.

Pour ce travail nous avons utilisé la méthode de conception UP et UML comme langage de modélisation, nous avons choisi de programmer l'application avec le langage PHP Objet avec Laravel comme framework et MySQL comme SGBD, et Semantic ui pour réaliser les interfaces.

Mots clés : Messagerie, PHP, Mysql, UP, UML, Génie logiciel.

ABSTRACT

In this brief, we have designed and implemented a professional internal messaging system for BEJAIA's teaching hospital with the aim of helping to improve the communication and the exchange of information via the messaging platform.

For this work we used the UP and UML design method as a modeling language, we chose to program the application with the language PHP Object with Laravel as framework and MySQL as DBMS, and Semantic ui to realize the interfaces.

Key words : Messaging, PHP, MySQL, UML, Software engineering.