

CURRICULUM VITAE

ETAT CIVIL :

Nom : RABETO

Prénoms : Mac Carthy Durando

Date et lieu de naissance : 07 juillet 1998 à Mahajanga

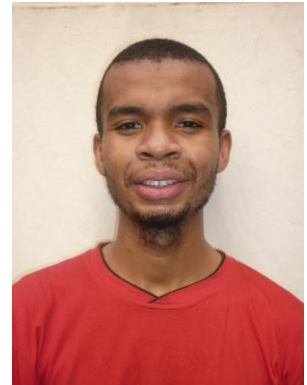
Adresse postale : Isada Fianarantsoa 301

Téléphone : +261 32 44 583 43 / +261 34 90 031 59

E-mail : mcrabeto@gmail.com

Situation matrimoniale : célibataire

Accroche : Développeur Junior



FORMATIONS ET DIPLOMES :

2019-2020 : Etudiant en 2^{ème} année de formation en Master professionnel à l'Ecole Nationale d'Informatique de l'Université de Fianarantsoa. Parcours : Informatique Générale

2018-2019 : Etudiant en 1^{ère} année de formation en Master professionnel à l'Ecole Nationale d'Informatique de l'Université de Fianarantsoa. Parcours : Informatique Générale

2019 : Obtention du diplôme de licence professionnelle en informatique à l'Ecole Nationale d'Informatique de l'Université de Fianarantsoa Parcours : Informatique Générale.

2015 : Obtention du diplôme de baccalauréat série C avec mention BIEN.

2014 – 2015 : Classe de terminale au Lycée Saint Gabriel à Mahajanga.

STAGES ET EXPERIENCES PROFESSIONNELLES :

Mars 2021 – Août 2021: Création d'un chatbot pour la gestion d'une foire aux questions au sein de l'entreprise YoungDev Antananarivo Madagascar.

Novembre 2019 - Janvier 2020: Création d'une application pour la gestion des résultats du concours d'entrée en 1^{ère} année en licence professionnelle en informatique à l'Ecole Nationale d'Informatique de l'Université de Fianarantsoa

Novembre 2018 - Janvier 2019 : Création d'une plateforme de paiement en ligne liée au mobile banking au sein de la société ASA Avaratr'Ankatso Antananarivo.

Novembre 2017 - Janvier 2018 : Développement d'une application CRM pour la gestion des réclamations des clients au sein de la société Orange Madagascar (notamment à la Tour Orange Ankorondrano).

Novembre 2016 - Janvier 2016: Développement d'une application pour la gestion bancaire de la BNI.

CONNAISSANCES EN INFORMATIQUE :

INFORMATIQUE ORIENTEE GENIE LOGICIEL

- Langages de programmation : C, C++, VB.NET, C#, Java
- Langages et technologies web : HTML5, CSS3 , Javascript, Node.js, JQuery, PHP, ASP,JSP
- Développement d'applications mobiles sous Android
- Frameworks: CodeIgniter, Ionic, Laravel, Django, Angular, VueJs, Flutter
- Méthodes : Merise, 2TUP
- Langage de modélisation : UML
- Systèmes de gestion de base de données : MySQL, PostgreSQL, Oracle, MongoDB
- Application CRM
- Langage SQL
- Logiciel de gestion de version : GIT
- Plateforme de gestion de projet : Trello
- Méthodologie agile : Scrum, XP

INFORMATIQUE ORIENTEE ADMINISTRATION SYSTEMES ET RESEAUX :

- Clonage et restauration système
- Réseaux TCP/IP
- Mise en place d'un réseau LAN
- Surveillance matériel et logiciel via le protocole SNMP.
- Administration systèmes et réseaux Microsoft : Active Directory, DHCP, DNS, IIS Serveur, GPO, Fail over (cluster de basculement), Load Balancing (partage de charge).
- Routage dynamique : OSPF, RIP, MPLS

- Routage statique
- Routeurs CISCO et MIKROTIK
- Outils de virtualisation machine : Virtual Box
- Outils de simulation réseau : GNS3
- Administration systèmes UNIX : Debian, CentOS
- Réseaux mobiles (GSM, GPRS, LTE)
- Cloud Computing avec OpenStack

INFORMATIQUE BUREAUTIQUE :

- Word, Excel, PowerPoint
- Internet

CONNAISSANCES LINGUISTIQUES :

	Compréhension à l'orale	Compréhension à l'écrit	Lire	Ecrire	Parler
Français	TB	B	TB	B	B
Anglais	AB	AB	AB	AB	AB

TB : Très Bien, B : Bien, AB : Assez Bien, P : Passable

ACTIVITES EXTRA-CURRICULAIRES :

- Sport pratique : basket-ball.
- Aime chanter et danser.
- Titulaire d'un diplôme de DELF (Diplôme d'Etudes en Langue Française niveau A2, B1, B2).
- Membre au sein de GBUF (Groupe Biblique Universitaire de Fianarantsoa).

SOMMAIRE

CURRICULUM VITAE	I
SOMMAIRE	IV
REMERCIEMENTS	VI
LISTE DES FIGURES	VII
LISTE DES TABLEAUX	IX
NOMENCLATURE	X
INTRODUCTION	1
PARTIE I : PRESENTATIONS	2
Chapitre 1. PRESENTATION DE L'ENI	3
1.1. Informations d'ordre général	3
1.2. Missions et historiques.....	3
1.3. Organigramme institutionnel de l'ENI	5
1.4. Domaines de spécialisation.....	6
1.5. Architecture des formations pédagogiques.....	7
1.6. Relations de l'ENI avec les entreprises et les organismes	10
1.7. Partenariat au niveau international.....	11
1.8. Débouchés Professionnels des Diplômés.....	13
1.9. Ressources Humaines	15
Chapitre 2. PRÉSENTATION DE YOUNGDEV	16
2.1. Historique et brève présentation de la société YoungDev	16
2.2. Fiche d'identification	16
2.3. Services	17
2.4. Organigramme	18
2.5. Valeurs	18
Chapitre 3. DESCRIPTION DU PROJET.....	19
3.1. Formulation.....	19
3.2. Objectif et besoins de l'utilisateur	19
3.3. Moyens nécessaires à la réalisation du projet	20
3.4. Résultats attendus.....	21
3.5. Chronogramme des travaux	21
PARTIE II : ANALYSE ET CONCEPTION	22

Chapitre 4. ANALYSE PREALABLE	23
4.1. Analyse de l'existant.....	23
4.2. Critique de l'existant.....	23
4.3. Conception avant-projet.....	24
Chapitre 5. ANALYSE CONCEPTUELLE	35
5.1. Dictionnaire des données	35
5.2. Règles de gestion	37
5.3. Représentation et spécification des besoins	37
5.4. Spécification des besoins techniques	58
5.5. Modélisation du domaine.....	59
Chapitre 6. CONCEPTION DETAILLEE.....	61
6.1. Architecture du système.....	61
6.2. Diagramme de séquence de conception pour chaque cas d'utilisation.....	62
6.3. Diagramme de classe de conception pour chaque cas d'utilisation.....	68
6.4. Diagramme de classe de conception global	72
6.5. Diagramme de paquetages	75
6.6. Diagramme de déploiement	78
PARTIE III : REALISATION	81
Chapitre 7. Mise en place de l'environnement de développement	82
7.1. Installation et configuration des outils.....	82
7.2. Architecture de l'application.....	87
Chapitre 8. DÉVELOPPEMENT DE L'APPLICATION	88
8.1. Création de la base de données	88
8.2. Codage de l'application	90
8.3. Présentation de l'application.....	93
CONCLUSION	100
BIBLIOGRAPHIE.....	XI
WEBOGRAPHIE	XII
GLOSSAIRE	XIII
TABLE DES MATIERES.....	XIV
RESUME	XVII
ABSTRACT.....	XVII

REMERCIEMENTS

Tout d'abord, je tiens à remercier Dieu pour sa bénédiction et pour sa grande protection pendant la réalisation de ce mémoire.

Ensuite, je remercie chaleureusement :

- Monsieur **RAFAMANTANANTSOA Fontaine**, Professeur à l'Université de Fianarantsoa, Président de l'Université de Fianarantsoa, pour avoir assuré le bon fonctionnement de l'Université pendant cette année universitaire ;
- Monsieur **RAMAMONJISOA Andriantiana Bertin Olivier**, Professeur Titulaire à l'Université de Fianarantsoa, Directeur de l'Ecole Nationale d'Informatique qui m'a donné l'opportunité de réaliser ce stage ;
- Monsieur **RABETAFIKA Louis Haja**, Assistant d'Enseignement Supérieur et de Recherche à l'Université de Fianarantsoa, Responsable du mention « Informatique » pour avoir assuré le fonctionnement des formations pédagogiques au sein de l'Ecole Nationale d'Informatique.
- Monsieur **RABEARISON Tsiory Mamitiana**, Ingénieur en informatique, Directeur Général de la société YoungDev ainsi que mon encadreur professionnel de m'avoir accepté comme stagiaire au sein de son entreprise et aussi de m'avoir encadré durant la réalisation du stage ;
- Monsieur **GILANTE Gesazafy**, Assistant d'Enseignement Supérieur et de Recherche à l'Université de Fianarantsoa, Responsable du parcours « Informatique Générale » pour l'organisation des formations pratiques ;
- Monsieur **RALAIVAO Jean Christian**, Assistant d'Enseignement Supérieur et de Recherche à l'Université de Fianarantsoa, Responsable du parcours « Génie Logiciel et Base de données » pour l'organisation des formations théoriques et à la fois mon encadreur pédagogique pour m'avoir encadré lors de l'élaboration de ce livre de mémoire.
- Monsieur **RAKOTOASIMBAHOAKA Cyprien Robert**, Maître de Conférences étant mon examinateur
- Monsieur **MAHATODY Thomas**, Maître de Conférences étant le président du jury
- Tous les membres du jury pour l'attention qu'ils m'accordent durant la présentation de ce mémoire.
- Tous les Enseignants et Chercheurs de l'Ecole Nationale d'Informatique, qui ont participé à notre formation.
- Tout le personnel de la Société YoungDev pour ses remarques et suggestions pertinentes durant notre stage.

De plus, j'exprime ma vive gratitude :

A ma famille qui m'a soutenu moralement et financièrement durant la réalisation de ce mémoire.

A tous ceux, de près ou de loin, qui ont contribué à la réalisation de ce travail.

LISTE DES FIGURES

Figure 1 : Organigramme de l'ENI	6
Figure 2 : Architecture des études correspondant au système LMD	8
Figure 3 : Illustration des différents services fournis par Youngdev	17
Figure 4 : Organigramme de la société Youngdev	18
Figure 5 : Les valeurs de l'entreprise Youngdev	18
Figure 6 : Représentations d'un acteur	38
Figure 7 : Formalisme de base de représentation d'un cas d'utilisation	39
Figure 8 : Diagramme des cas d'utilisation globale	42
Figure 9 : Formalisme général du cadre d'un diagramme de séquence	50
Figure 10 : Diagramme de séquence système du cas d'utilisation « S'authentifier »	51
Figure 11 : Diagramme de séquence système du cas d'utilisation « Afficher statistique »	52
Figure 12 : Diagramme de séquence système du cas d'utilisation « Créer intention »	53
Figure 13 : Diagramme de séquence système du cas d'utilisation "Mettre à jour intention"	54
Figure 14 : Diagramme de séquence système du cas d'utilisation "Discuter avec l'internaute" ...	55
Figure 15 : Diagramme de séquence système du cas d'utilisation "Répondre aux requêtes non comprises par le chatbot"	56
Figure 16 : Diagramme de séquence système du cas d'utilisation "Poser question"	57
Figure 17 : Diagramme de séquence système du cas d'utilisation "Visualiser la liste des internautes"	58
Figure 18 : Modélisation du domaine	60
Figure 19 : Schéma de l'architecture MVC	61
Figure 20 : Diagramme de séquence de conception du cas d'utilisation "S'authentifier"	63
Figure 21 : Diagramme de séquence de conception du cas d'utilisation "Afficher statistique" ...	63
Figure 22 : Diagramme de séquence de conception du cas d'utilisation du cas d'utilisation "Créer intention"	64
Figure 23 : Diagramme de séquence de conception du cas d'utilisation "Mettre à jour intention"	65
Figure 24 : Diagramme de séquence de conception du cas d'utilisation "Discuter avec l'internaute"	65
Figure 25 : Diagramme de séquence de conception du cas d'utilisation "Répondre aux requêtes non comprises par le chatbot"	66
Figure 26 : Diagramme de séquence de conception du cas d'utilisation "Poser question"	67
Figure 27: Diagramme de séquence de conception du cas d'utilisation "Visualiser la liste des internautes"	67
Figure 28 : Diagramme de classe de conception du cas d'utilisation "S'authentifier"	68
Figure 29 : Diagramme de classe de conception du cas d'utilisation "Afficher statistique"	68
Figure 30: Diagramme de classe de conception du cas d'utilisation "Créer intention"	69
Figure 31 : Diagramme de classe de conception du cas d'utilisation « Mettre à jour intention ».	69
Figure 32 : Diagramme de classe de conception du cas d'utilisation "Discuter avec l'internaute"	70
Figure 33 : Diagramme de classe de conception du cas d'utilisation "Répondre aux requêtes non comprises par le chatbot"	71
Figure 34 : Diagramme de classe de conception du cas d'utilisation "Poser question"	71

Figure 35 : Diagramme de classe de conception du cas d'utilisation "Visualiser la liste des internautes"	72
Figure 36 : Formalisme général d'une classe et exemples	73
Figure 37 : Exemple de multiplicités	74
Figure 38 : Diagramme de classe de conception global	75
Figure 39 : Formalisme de représentation de paquetages	76
Figure 40 : Exemple de liens de dépendance entre paquetages	77
Figure 41 : Diagramme de paquetage du domaine d'étude	78
Figure 42 : Représentation de nœuds	79
Figure 43 : Diagramme de déploiement du domaine d'étude	80
Figure 44 : Installation de Node.js	83
Figure 45 : Installation d'Angular	83
Figure 46 : Installation de MongoDB	84
Figure 47 : Installation de Visual Studio Code	85
Figure 48 : Installation de Visual Paradigm for UM	86
Figure 49 : Architecture de l'application	87
Figure 50 : Page d'authentification	93
Figure 51 : Page tableau de bord	94
Figure 52 : Liste des intentions	95
Figure 53 : Page de création d'intention	96
Figure 54 : Page d'édition d'une intention	97
Figure 55 : Page de discussion instantanée	97
Figure 56 : Liste des sessions internautes	98
Figure 57 : Liste des messages non comprises par le chatbot	98
Figure 58 : Zone d'édition de l'intention d'un message non comprise	99
Figure 59 : Boîte de discussion de l'internaute avec le chatbot	99

LISTE DES TABLEAUX

Tableau 1 : Organisation du système de formation pédagogique de l'Ecole.....	7
Tableau 2 : Listes des formations existant à l'ENI	9
Tableau 3 : Débouchés professionnels éventuels des diplômés	14
Tableau 4 : Caractéristiques des matériels.....	20
Tableau 5 : Moyens logiciels utilisés.....	20
Tableau 6 : Chronogramme des travaux	21
Tableau 7 : Caractéristiques des matériels.....	23
Tableau 8 : Avantages et inconvénients de chaque solution	24
Tableau 9 : Comparaison de quelques framework Javascript.....	26
Tableau 10 : Comparaison entre NodeJS et PHP	28
Tableau 11 : Comparaison de quelques SGBD	29
Tableau 12 : Comparaison entre Visual Studio Code et Webstorm	30
Tableau 13 : Comparaison entre UML Designer et Visual Paradigm for UML.....	31
Tableau 14 : Comparaison entre un chatbot simple et un chatbot complexe	32
Tableau 15 : Dictionnaire des données	35
Tableau 16 : Informations détaillées de chaque acteur.....	39
Tableau 17 : Liste des cas d'utilisation	40
Tableau 18 : Priorité de chaque cas d'utilisation	49

NOMENCLATURE

2TUP: The Two Tracks Unified Process
A : Alternative
CEO: Chief Executive Officer
CSS : Cascade StyleSheet
CMS : Customer Management System
CPI : Chef de projet Interne
CPE : Chef de projet Externe
CTO : Chief Technical Officer
ENI : Ecole Nationale d'Informatique
E : Exceptionnelle
EI : Entreprise Individuel
ESN : Entreprise de Service Numérique
FAQ: Frequently Asked Questions
Go : Giga-octets
GHz : Giga Hertz
HTML: Hyper Text Markup Language
IA : Intelligence Artificiel
IDE: Integrated Development Environement
JDK : Java Development Kit
JS : JavaScript
JDBC: Java DataBase Connectivity
MVC: Model View Controller
MEAN: Mongo Express Angular Node.js
NLP : Natural Language Processing
NgRx : Reactive State for Angular
ODBC: Open DataBase Connectivity
OS : Operating System
PHP: Hypertext Preprocessor
RAM: Random Access Memory
RG : Règle de gestion
SDK : Software Development Kit
SGBD : Système de Gestion de Base de Donnée
SQL : Structured Query Language
SSD : Solid State Drive
UML: Unified Modeling Language
UP : Unified Process
VSC: Visual Studio Code
XML : eXtensible Markup Language

INTRODUCTION

L'avantage compétitif des entreprises repose sur leur capacité à répondre rapidement aux changements des besoins du marché et augmenter leurs bénéfices en produisant des biens ou des services qui satisfaits les besoins de la clientèle.

Dans un monde en constante évolution et un environnement concurrentiel, un avantage compétitif est de bénéficier de l'information voulu à tout moment.

Face à cette dernière affirmation, comment peut-on aider les clients à satisfaire leurs besoins à tout moment ? Comment peut-on aussi leur aider à bénéficier en permanence des informations concernant les services fournis par l'entreprise ?

Dans ce présent projet, on a développé au sein de l'entreprise YoungDev basé à Antananarivo un agent conversationnel (ou chatbot) permettant de répondre en temps réels aux requêtes envoyées par les visiteurs du site web de l'entreprise.

Pour mettre en place un tel système, nous allons utiliser une méthode de conception, puis un langage de programmation, et un SGBD et enfin un algorithme d'intelligence artificiel permettant de comprendre le langage naturel du visiteur.

Ainsi, l'entreprise attend qu'on puisse fournir aux clients un agent conversationnel permettant de répondre automatiquement aux requêtes envoyées par le visiteur du site web de l'entreprise. Dans le cas où le visiteur envoie des requêtes complexe, c'est un agent humain qui prend le relais.

Pour mieux entrer dans le vif détail du sujet, nous allons diviser notre travail en trois grandes parties, premièrement nous allons évoquer la partie « présentation » suivit de la partie « analyse et conception » et enfin nous allons voir la partie « réalisation ».

PARTIE I : PRESENTATIONS

Chapitre 1. PRESENTATION DE L'ENI

1.1. Informations d'ordre général

L'Ecole Nationale d'Informatique, en abrégé ENI, est un établissement d'enseignement supérieur rattaché académiquement et administrativement à l'Université de Fianarantsoa.

Le siège de l'Ecole se trouve à Tanambao- Antaninarenina à Fianarantsoa.

Les coordonnées pour la prise de contact avec l'Ecole sont les suivantes :

Ecole Nationale d'Informatique (ENI)
Antaninarenina Tanambao
BP 1487 FIANARANTSOA (301)
Téléphones : 034 05 733 36 ou 033 42 302 02
Adresse e-mail : eni@eni-univ-fianar.mg.
Site : www.eni-univ-fianar.mg

1.2. Missions et historiques

L'ENI se positionne sur l'échiquier socio-éducatif malgache comme étant le plus puissant secteur de diffusion et de vulgarisation des connaissances et des technologies informatiques.

Cette Ecole Supérieure peut être considérée aujourd'hui comme la vitrine et la pépinière des élites informaticiennes du pays.

L'Ecole s'est constituée de façon progressive au sein du Centre Universitaire Régional (CUR) de Fianarantsoa.

De façon formelle, l'ENI était constituée et créée au sein du (CUR) par le décret N° 83-185 du 24 Mai 1983, comme étant le seul établissement Universitaire Professionnalisé au niveau national, destiné à former des techniciens et des Ingénieurs de haut niveau, aptes à répondre aux besoins et exigences d'Informatisation des entreprises, des sociétés et des organes implantés à Madagascar.

L'ENI a par conséquent pour mission de former des spécialistes informaticiens compétents et opérationnels de différents niveaux notamment :

- En fournissant à des étudiants des connaissances de base en informatique ;
- En leur transmettant le savoir-faire requis, à travers la professionnalisation des formations dispensées et en essayant une meilleure adéquation des formations par rapport aux besoins évolutifs des sociétés et des entreprises.

- En initiant les étudiants aux activités de recherche dans les différents domaines des Technologies de l'information et de la communication (TIC).

L'implantation de cette Ecole Supérieure de technologie de pointe dans un pays en développement et dans une Province (ou Faritany) à tissu économique et industriel faiblement développé ne l'a pourtant pas défavorisée, ni empêchée de former des spécialistes informaticiens de bon niveau, qui sont recherchés par les entreprises, les sociétés et les organismes publics et privés sur le marché de l'emploi.

La filière de formation d'Analystes Programmeurs a été mise en place à l'Ecole en 1983, et a été gelée par la suite en 1996, tandis que la filière de formation d'ingénieurs a été ouverte à l'Ecole en 1986.

Dans le cadre du Programme de renforcement en l'Enseignement Supérieur (PRESUP), la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes des informatiques a été mise en place en 1986 grâce à l'appui matériel et financier de la Mission Française de coopération auprès de l'Ambassade de France à Madagascar.

Une formation pour l'obtention de la certification CCNA et / ou NETWORK + appelée « CISCO Networking Academy » a été créée à l'Ecole en 2002-2003 grâce au partenariat avec CISCO SYSTEM et l'Ecole Supérieure Polytechnique d'Antananarivo (ESPA). Cependant, cette formation n'avait pas duré longtemps.

Une formation de troisième cycle a été ouverte à l'Ecole depuis l'année 2003 – 2004 grâce à la coopération académique et scientifique entre l'Université de Fianarantsoa pour le compte de l'ENI et l'Université Paul Sabatier de Toulouse (UPST).

Cette filière avait pour objectif de former certains étudiants à la recherche dans les différents domaines de l'Informatique, et notamment pour préparer la relève des Enseignants-Chercheurs qui étaient en poste.

Pendant l'année 2007-2008, la formation en vue de l'obtention du diplôme de Licence Professionnelle en Informatique a été mise en place à l'ENI avec les deux options suivantes de formation :

- Génie Logiciel et base de Données.
- Administration des Systèmes et réseaux.

La mise en place à l'Ecole de ces deux options de formation devait répondre au besoin de basculement vers le système Licence – Master – Doctorat (LMD).

Mais la filière de formation des Techniciens Supérieurs en Maintenance des Systèmes Informatiques a été gelée en 2009.

En vue de surmonter les difficultés de limitation de l'effectif des étudiants accueillis à l'Ecole, notamment à cause du manque d'infrastructures, un système de « Formation Hybride » a été mise en place à partir de l'année 2010. Il s'agit en effet d'un système de formation semi-présentielle et à distance avec l'utilisation de la visioconférence pour la formation à distance.

Le système de formation hybride a été ainsi créé à Fianarantsoa ainsi qu'Université de Toliara.

1.3. Organigramme institutionnel de l'ENI

Cet organigramme de l'Ecole est inspiré des dispositions du décret N° 83-185 du 24 Mai 1983.

Sur cet organigramme, l'Ecole placée sous la tutelle académique et administrative de l'Université de Fianarantsoa, est dirigée par un Directeur élu par les Enseignants – Chercheurs permanents de l'Etablissement et nommé par un décret pris en Conseil des Ministres pour un mandat de 3 ans.

Le Conseil de l'Ecole est l'organe délibérant de l'Ecole.

Le Collège des Enseignants regroupant tous les enseignants-chercheurs de l'Ecole est chargé de résoudre les problèmes liés à l'organisation pédagogique des enseignements. Il propose et coordonne les programmes d'activités pédagogiques.

Le Conseil Scientifique propose les orientations pédagogiques et scientifiques de l'établissement, en tenant compte notamment de l'évolution du marché de travail et de l'adéquation des formations dispensées par rapport aux besoins des entreprises. Il coordonne les programmes de recherche à mettre en œuvre à l'Ecole.

Le Secrétariat principal coordonne les activités des services administratifs (Scolarité, Comptabilité, et Intendance).

Conformément aux textes en vigueur régissant les Etablissements malgaches d'Enseignement Supérieur, qui sont barrés sur le système LMD, les Départements de Formation pédagogique ont été ainsi remplacés par des Mentions et des parcours. Et les chefs des Départements ont été ainsi remplacés par des responsables des mentions et les responsables des parcours.

La Figure 1 présente l'organigramme actuel de l'Ecole.

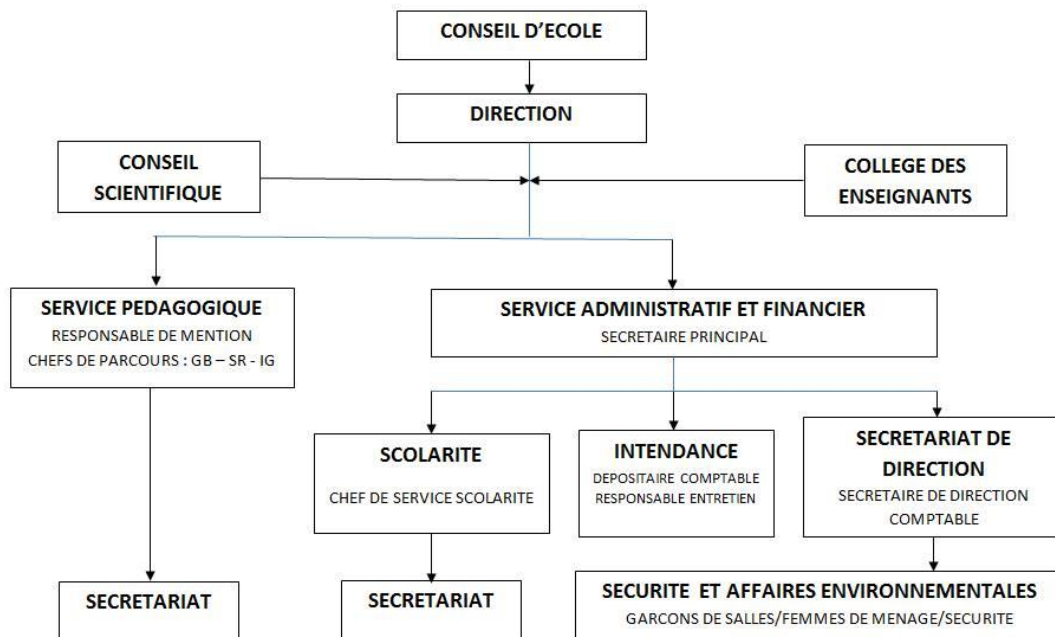


Figure 1 : Organigramme de l'ENI

1.4. Domaines de spécialisation

Les activités de formation et de recherche organisées à l'ENI portent sur les domaines suivants :

- Génie logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale,
- Modélisation informatique et mathématique des Systèmes complexes.

D'une manière plus générale, les programmes des formations sont basés sur l'informatique de gestion et sur l'informatique des Systèmes et Réseaux. Et les modules de formation intègrent aussi bien des éléments d'Informatique fondamentale que des éléments d'Informatique appliquée.

Le Tableau 1 décrit l'organisation du système de formation pédagogique de l'Ecole.

Tableau 1 : Organisation du système de formation pédagogique de l'Ecole.

Formation théorique	Formation pratique
<ul style="list-style-type: none">➤ Enseignement théorique➤ Travaux dirigés➤ Travaux pratiques➤ Conférences	<ul style="list-style-type: none">➤ Etude de cas➤ Travaux de réalisation➤ Projets / Projets tutorés➤ Voyage d'études➤ Stages en entreprises

1.5. Architecture des formations pédagogiques

Le recrutement des étudiants à l'ENI se fait uniquement par voie de concours d'envergure nationale en première année de Licence Professionnelle.

Les offres de formation organisées à l'Ecole ont été validées par la Commission Nationale d'Habilitation (CNH) auprès du Ministère de l'Enseignement Supérieur et de la Recherche Scientifique selon les dispositions de l'Arrêté N°31.174/2012-MENS en date du 05 Décembre 2012.

Au sein de l'ENI, il existe une seule mention (INFORMATIQUE) et trois parcours :

- Génie logiciel et Base de Données ;
- Administration des Systèmes et Réseaux ;
- Informatique Générale

L'architecture des études à trois niveaux conformément au système Licence- Master- Doctorat (LMD) permet les comparaisons et les équivalences académiques des diplômes au niveau international.

- L = Licence (Bac + 3) = L1, L2, L3 = 6 semestres S1 à S6
- M = Master (Bac + 5) = M1, M2 = 4 semestres S7 à S10

Le diplôme de Licence est obtenu en 3 années des études après Baccalauréat. Et le diplôme de Master est obtenu en 2 ans après obtenu du diplôme de LICENCE.

Le MASTER PROFESSIONNEL est un diplôme destiné à la recherche emploi au terme des études.

Le MASTER RECHERCHE est un diplôme qui remplace l'ancien Diplôme d'Etudes Approfondies (DEA), et qui permet de s'inscrire directement dans une Ecole Doctorale au terme des études.

- D = Doctorat (Bac +8)

Le Doctorat est un diplôme qu'on peut obtenir en 3 ans après l'obtention du diplôme de MASTER RECHERCHE.

La Figure 2 présente l'architecture des études correspondant au système LMD.

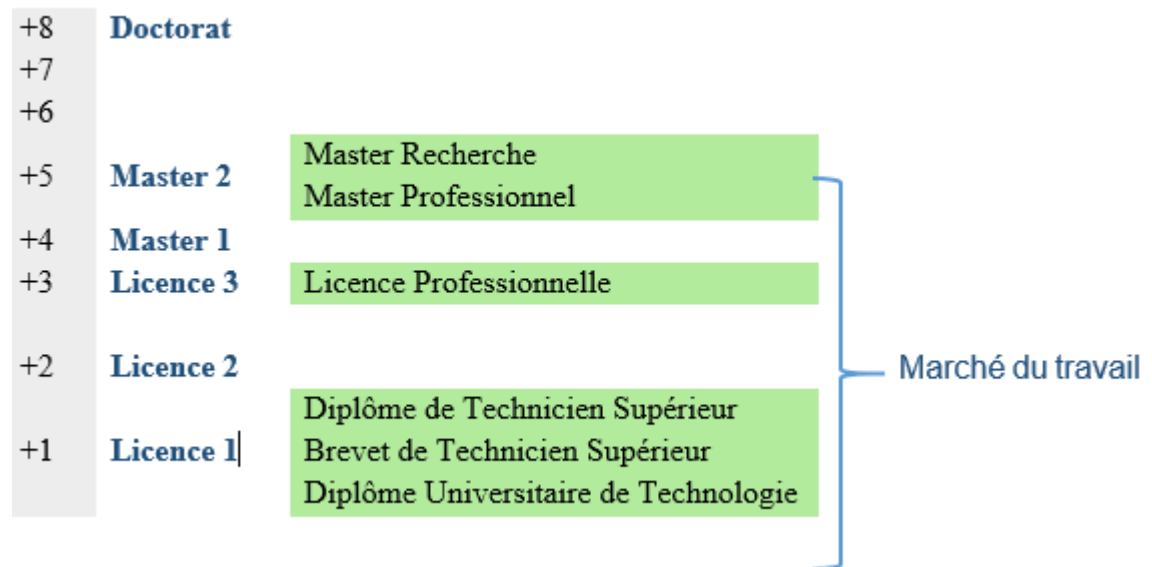


Figure 2 : Architecture des études correspondant au système LMD

La licence peut avoir une vocation générale ou professionnelle.

Le master peut avoir une vocation professionnelle ou de recherche.

Le Tableau 2 comprend la liste des formations existantes à l'ENI.

Tableau 2 : Listes des formations existant à l'ENI

	FORMATION EN	
	LICENCE PROFESSIONNELLE ET HYBRIDE	MASTER
Condition d'admission	Par voie de concours Formation Professionnelle : 100 candidats Formation généraliste : 150 candidats	
Condition d'accès	Bac de série C, D ou Technique	Etre titulaire de Licence Professionnelle
Durée de formation	3 ans	2 ans
Diplôme à délivrer	Diplôme de Licence Professionnelle en Informatique	Diplôme de Master Professionnel Diplôme de Master Recherche

L'accès en première année de formation en MASTER se fait automatiquement pour les étudiants de l'Ecole qui ont obtenu le diplôme de Licence Professionnelle.

Le Master Recherche permet à son titulaire de poursuivre directement des études en doctorat et de s'inscrire directement dans une Ecole Doctorale.

Les Ecoles Doctorales jouissent d'une autonomie de gestion par rapport aux Etablissements de formation universitaire.

Il convient de signaler que par arrêté ministériel N° 21.626/2012 – MESupRES publié le 9 Août 2012 par la Commission Nationale d'Habilitation (CNH), l'Ecole Doctorale « Modélisation – Informatique » a été habilitée pour l'Université de Fianarantsoa.

Depuis l'année universitaire 2010-2011, l'ENI s'est mise à organiser des formations hybrides en informatique dans les différentes régions (Fianarantsoa, Toliara) en raison de l'insuffisance de la capacité d'accueil des infrastructures logistiques. En effet, le système de formation hybride semi - présentielle utilise la visioconférence pour la formation à distance.

Bien qu'il n'existe pas encore au niveau international de reconnaissance écrite et formelle des diplômes délivrés par l'ENI, les étudiants diplômés de l'Ecole sont plutôt bien accueillis dans les instituts universitaires étrangères (CANADA, Suisse, France...).

1.6. Relations de l'ENI avec les entreprises et les organismes

Les stages effectués chaque année par les étudiants mettent l'Ecole en rapport permanent avec plus de 300 entreprises et organismes publics, semi-publics et privés, nationaux et internationaux.

L'Ecole dispose ainsi d'un réseau d'entreprises, de sociétés et d'organismes publics et privés qui sont des partenaires par l'accueil en stage de ses étudiants, et éventuellement pour le recrutement après l'obtention des diplômes par ces derniers.

Les compétences que l'Ecole cherche à développer chez ses étudiants sont l'adaptabilité, le sens de la responsabilité, du travail en équipe, le goût de l'expérimentation et l'innovation.

En effet, la vocation de l'ENI est de former des techniciens supérieurs de niveau LICENCE et des ingénieurs de type généraliste de niveau MASTER avec des qualités scientifiques, techniques et humaines reconnues, capables d'évoluer professionnellement dans des secteurs d'activité variés intégrant l'informatique.

Les stages en milieu professionnel permettent de favoriser une meilleure adéquation entre les formations à l'Ecole et les besoins évolutifs du marché de l'emploi.

Les principaux débouchés professionnels des diplômés de l'Ecole concernent les domaines suivants :

- ✓ L'informatique de gestion d'entreprise
- ✓ Les technologies de l'information et de la communication (TIC)
- ✓ La sécurité informatique des réseaux
- ✓ L'administration des réseaux et des systèmes
- ✓ Les services bancaires et financiers, notamment le Mobile Banking
- ✓ Les télécommunications et la téléphonie mobile
- ✓ Les Big Data
- ✓ Le commerce, la vente et l'achat, le Marketing
- ✓ L'ingénierie informatique appliquée
- ✓ L'écologie et le développement durable

Parmi les sociétés, entreprises et organismes partenaires de l'Ecole, on peut citer : ACCENTURE Mauritius, Air Madagascar, Ambre Associates, Airtel, Agence

Universitaire de la Francophonie (AUF) , B2B, Banque Centrale, BFG-SG, BIANCO, BLUELINE, Bureau national de gestion des Risques et des catastrophes (BNGRC), CEDII Fianarantsoa, Data Consulting, Central Test, Centre National Antiacridien, CNRE, CHU, CNRIT, COLAS, Direction Générale des Douanes, DLC, FID, FTM, GNOSYS, IBONIA, INGENOSIA, INSTAT, JIRAMA, JOUVE, MADADEV, MEF, MEN, MESupRES, MFB, MIC, Min des postes/Télécommunications et du Développement Numérique, NEOV MAD, Ny Havana, Madagascar National Parks, OMNITEC, ORANGE, OTME, PRACCESS, QMM Fort-Dauphin, SMMC, SNEDADRS Antsirabe, Sénat, Société d'Exploitation du Port de Toamasina (SEPT), SOFTWELL, Strategy Consulting, TELMA, VIVETEC, Société LAZAN'I BETSILEO, WWF, YoungDev ...

L'organisation de stages en entreprise contribue non seulement à renforcer la professionnalisation des formations dispensées, mais elle aide surtout à accroître de façon exceptionnelle les opportunités d'embauche pour les diplômés de l'Ecole.

1.7. Partenariat au niveau international

Entre 1996 et 1999, l'ENI avait bénéficié de l'assistance technique et financière de la Mission Française de Coopération et d'action culturelle dans le cadre du Programme de Renforcement de l'Enseignement Supérieur (PRESUP) consacré à l'Ecole a notamment porté sur :

- Une dotation en logiciels, micro-ordinateurs, équipements de laboratoire de maintenance et de matériels didactiques,
- La réactualisation des programmes de formation assortie du renouvellement du fonds de la bibliothèque,
- L'appui à la formation des formateurs,
- L'affectation à l'Ecole d'Assistants techniques français.

De 2000 à 2004, l'ENI avait fait partie des membres du bureau de la Conférence Internationale des Ecoles de formation d'Ingénieurs et Technicien d'Expression Française (CITEF).

Les Enseignants-Chercheurs de l'Ecole participent régulièrement aux activités organisées dans le cadre du Colloque Africain sur la Recherche en Informatique (CARI).

L'ENI avait également signé un accord de coopération inter-universitaire avec l'Institut de Recherche en Mathématiques et Informatique Appliquées (IREMIA) de l'Université de la Réunion, l'Université de Rennes 1, l'INSA de Rennes, l'Institut National Polytechnique de Grenoble (INPG).

A partir du mois de Juillet 2001, l'ENI avait abrité le Centre de Réseau Opérationnel (Network Operating Center) du point d'accès à Internet de l'Ecole ainsi que de l'Université de Fianarantsoa. Grâce à ce projet américain qui a été financé par l'USAID Madagascar, l'ENI de l'Université de Fianarantsoa avait été dotées d'une ligne spécialisée d'accès permanent au réseau Internet.

L'ENI avait de même noué des relations de coopération avec l'Institut de Recherche pour le Développement (IRD).

L'objet du projet de coopération avait porté sur la modélisation environnementale du Corridor forestier de Fandriana jusqu'à Vondrozo (COFAV). Dans ce cadre, un atelier scientifique international avait été organisé à l'ENI en Septembre 2008. Cet atelier scientifique avait eu pour thème de modélisation des paysages.

Et dans le cadre du programme scientifique PARRUR, l'IRD avait financé depuis 2010 le projet intitulé « Forêts, Parcs et Pauvreté dans le Sud de Madagascar (FPPSM). Des étudiants en DEA et des Doctorants issus de l'ENI avaient participé à ce Programme.

Par ailleurs, depuis toujours la même année 2010, l'ENI de Fianarantsoa avait été sélectionnée pour faire partie des organismes partenaires de l'Université de Savoie dans le cadre du projet TICEVAL relatif à la certification des compétences en TIC ;

Le projet TICEVAL avait été financé par le Fonds Francophone des Inforoutes pour la période allant de 2010 à 2012, et il avait eu pour objectif de généraliser la certification des compétences en Informatique et Internet du type C2i2e et C2imi.

Dans le cadre du projet TICEVAL, une convention de coopération avec l'Université de Savoie avait été signée par les deux parties concernées. La mise en œuvre de la Convention de Coopération avait permis d'envoyer des étudiants de l'ENI à Chambéry pour poursuivre des études supérieures en Informatique.

Enfin et non des moindres, l'ENI avait signé en Septembre 2009 un protocole de collaboration scientifique avec l'ESIROI – STIM de l'Université de la Réunion.

Comme l'ENI constitue une pépinière incubatrice de technologie de pointe, d'emplois et d'entreprises, elle peut très bien servir d'instrument efficace pour renforcer la croissance économique du pays, et pour lutter contre la Pauvreté.

De même que le statut de l'Ecole devrait permettre de renforcer la position concurrentielle de la Grande Ile sur l'orbite de la mondialisation grâce au développement des nouvelles technologies.

1.8. Débouchés Professionnels des Diplômés

Le chômage des jeunes diplômés universitaires fait partie des maux qui gangrènent Madagascar. L'environnement socio-politique du pays depuis 2008 jusqu' à ce jour a fait que le chômage des diplômés est devenu massif par rapport aux établissements de formation supérieure existants.

Cependant, les formations proposées par l'Ecole permettent aux diplômés d'être immédiatement opérationnels sur le marché du travail avec la connaissance d'un métier complet lié à l'informatique aux TIC.

L'Ecole apporte à ses étudiants un savoir-faire et un savoir-être qui les accompagnent tout au long de leur vie professionnelle. Elle a une vocation professionnalisante.

Les diplômés en LICENCE et en MASTER issus de l'ENI peuvent faire carrière dans différents secteurs liés à l'Informatique et aux TIC.

L'Ecole bénéficie aujourd'hui de 38 années d'expériences pédagogiques et de reconnaissance auprès des sociétés, des entreprises et des organismes. C'est une Ecole Supérieure de référence en matière informatique.

Par conséquent, en raison de fait que l'équipe pédagogique de l'Ecole est expérimentée, les enseignants-chercheurs et les autres formateurs de l'Ecole sont dotés d'une grande expérience dans l'enseignement et dans le milieu professionnel.

L'Ecole est fière de collaborer de façon régulière avec un nombre croissant d'entreprises, de sociétés et d'organismes publics et privés à travers les stages des étudiants. Les formations dispensées à l'Ecole sont ainsi orientées vers le besoin et les attentes des entreprises et des sociétés.

L'Ecole fournit à ses étudiants de niveau LICENCE et MASTER des compétences professionnelles et métiers indispensables pour les intégrer sur le marché du travail.

L'Ecole s'efforce de proposer à ses étudiants une double compétence à la fois technologique et managériale combinant l'informatique de gestion ainsi que l'administration des réseaux et systèmes.

D'une manière générale, les diplômés de l'ENI n'éprouvent pas de difficultés particulières à être recrutés au terme de leurs études. Cependant, l'ENI recommande à ses diplômés de promouvoir l'entrepreneuriat en TIC et de créer des cybercafés, des SSII ou des bureaux d'études, en vue de la mise en œuvre de la politique d'employabilité des diplômés.

La liste des débouchés professionnels éventuels des diplômés se trouve dans le Tableau 3.

Tableau 3 : Débouchés professionnels éventuels des diplômés

LICENCE	<ul style="list-style-type: none">➤ Analyste➤ Programmeur➤ Administrateur de site web/de portail web➤ Assistant Informatique et internet➤ Chef de projet web ou multimédia➤ Développeur Informatique ou multimédia➤ Intégrateur web ou web designer➤ Hot liner/Hébergeur Internet➤ Agent de référencement ou Référenceur➤ Technicien/Supérieur de help desk sur Informatique➤ Responsable de sécurité web➤ Administrateur de réseau➤ Administrateur de cybercafé
MASTER	<ul style="list-style-type: none">➤ Administrateur de Base de Données➤ Administrateur de réseau et système➤ Architecte de système d'information➤ Développeur d'application /web /java/Python/ IOS /Android➤ Ingénieur réseau➤ Webmaster /web designer➤ Concepteur Réalisateur d'applications➤ Directeur du système d'information➤ Directeur de projet informatique➤ Chef de projet informatique➤ Responsable de sécurité informatique➤ Consultant fonctionnel ou freelance➤ Administrateur de cybercafé

1.9. Ressources Humaines

Les ressources humaines au sein de l'ENI sont les suivantes :

- Directeur de l'Ecole : Professeur RAMAMONJISOA Bertin Olivier
- Responsable de Mention : Monsieur RABETAFIKA Louis Haja
- Responsable de Parcours « Génie Logiciel et Base de Données » : Monsieur RALAIVAO Jean Christian
- Responsable de Parcours « Administration Systèmes et Réseaux » : Monsieur SIAKA
- Responsable de Parcours « Informatique Générale » : Monsieur GILANTE Gesazafy
- Nombre d'Enseignants permanents : 12 dont deux (01) Professeur Titulaire, 1 (un) Professeur, cinq (05) Maîtres de Conférences et cinq (05) Assistants d'Enseignement Supérieur et de Recherche
- Nombre d'Enseignants vacataires : 10
- Personnel Administratif : 23

Chapitre 2. PRÉSENTATION DE YOUNGDEV

Dans ce chapitre, nous allons présenter brièvement l'entreprise Youngdev.

2.1. Historique et brève présentation de la société YoungDev

YoungDev est une entreprise de service numérique (ESN) dont le principal service tourne autour de la **Technologie de Gestion et de l'automatisation**. Elle a été créée en 2017 par les deux principaux fondateurs RABEARISON Tsiory et TOKINIAINA Eddy. Son siège social se trouve à Ankazomanga Antananarivo 101 Madagascar. Cette entreprise regroupe une équipe jeune et dynamique ayant pour but de contribuer à l'évolution technologique de la Grande île.

2.2. Fiche d'identification

Nous allons présenter les informations détaillées concernant l'entreprise Youngdev

2.2.1 Statut juridique

- ❖ Raison sociale : **YoungDev**
- ❖ Statut : **EI (Entreprise Individuel)**
- ❖ Date de création : **2017**
- ❖ Numéro d'Identification Fiscale : **3003470254**
- ❖ Numéro statistique : **63122 11 2019 0 04999**
- ❖ RCS : **RCS Antananarivo 2020 A 00195**

2.2.2 Direction

- ❖ Gérant : Monsieur RABEARISON Tsiory Mamitiana

2.2.3 Localisation et contacts

- ❖ Adresse : Lot III E 22 Ankazomanga Sud
- ❖ Téléphone : +261 34 84 332 26
- ❖ E-mail personnel : tsiorymamitiana@gmail.com
- ❖ E-mail professionnel : tsiory@youngdev.fr
- ❖ Site web: <https://youngdev.mg/>

2.3. Services

Spécialisée dans les nouvelles technologies, le génie logiciel, l'intelligence artificielle le design et ergonomie web, Youngdev se caractérise par ses compétences techniques de maîtrise d'œuvre ainsi que par ses équipes jeunes et dynamiques.

Youngdev propose des offres de service en termes de technologie informatique (conception, développement et la maintenance des sites web et applications, ergonomie et design web). Elle ambitionne d'accompagner ses clients dans la réussite de leurs projets, grâce à son expertise en la matière et à ses prestataires qualifiés.

La morale de Youngdev est axée sur « l'automatisation des tâches » afin de faire profiter des services innovants au grand public.

La figure 3 indique les différents services fournis par l'entreprise YoungDev.

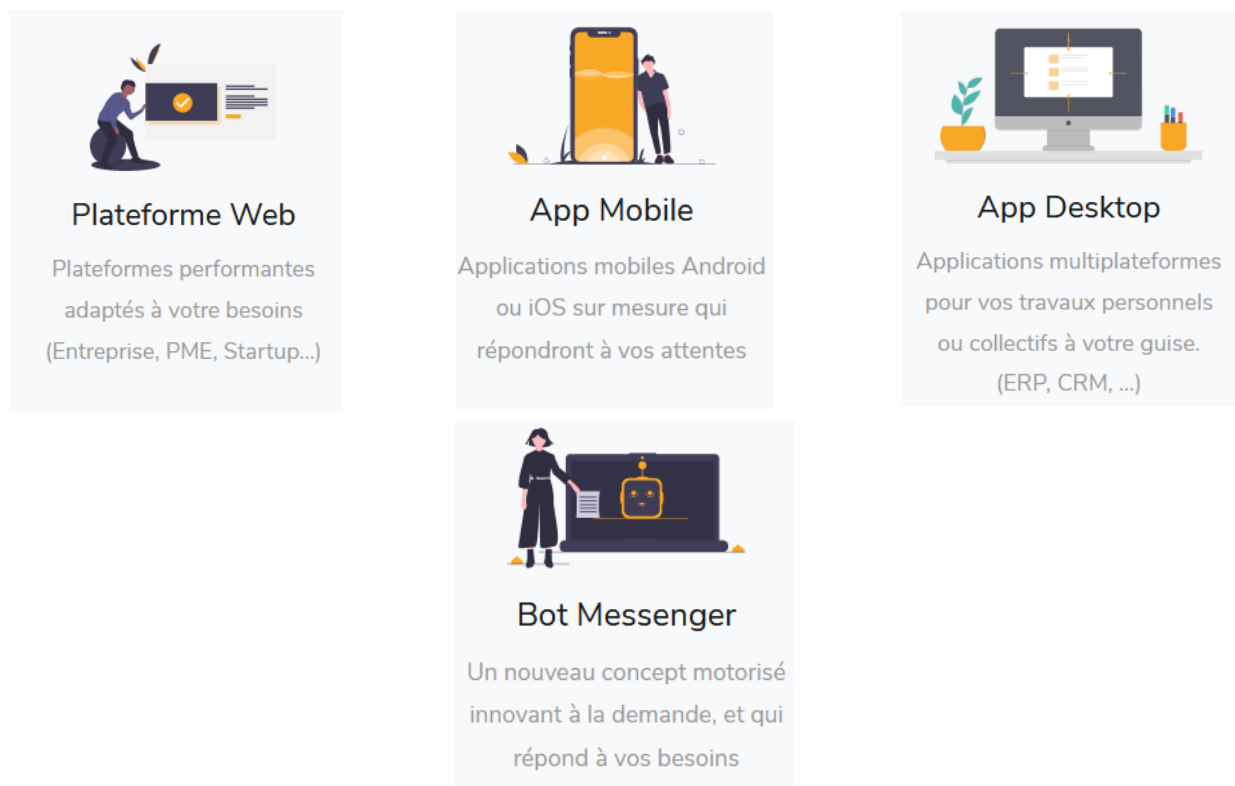


Figure 3 : Illustration des différents services fournis par Youngdev

2.4. Organigramme

La figure 4 présente l'organigramme de l'entreprise YoungDev :

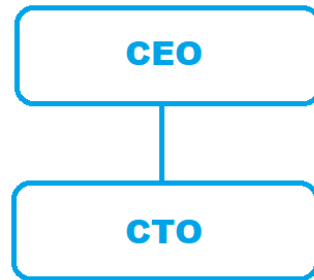


Figure 4 : Organigramme de la société Youngdev

2.5. Valeurs

L'entreprise Youngdev s'appuie ces quatre valeurs pour assurer une meilleure productivité sur les services fournis. La figure 5 fournit les différentes valeurs de l'entreprise.

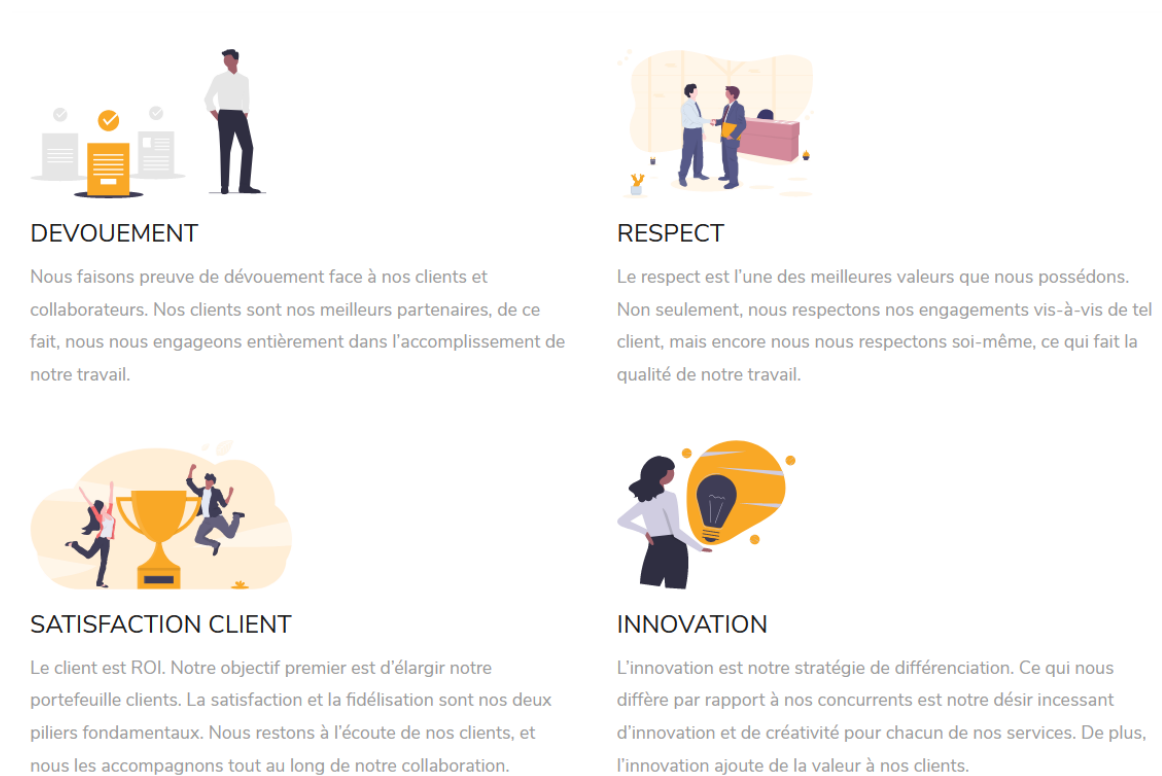


Figure 5 : Les valeurs de l'entreprise Youngdev

Chapitre 3. DESCRIPTION DU PROJET

Dans ce chapitre nous allons aborder une brève description du projet à effectuer, présenter les besoins de l'utilisateur, évoquer les moyens nécessaires à la réalisation du projet enfin préciser les résultats attendus du présent projet.

3.1. Formulation

Le projet consiste à créer un agent conversationnel (ou chatbot) permettant de répondre automatiquement aux questions posées par les visiteurs du site web de l'entreprise. Ces questions auxquelles le chatbot va répondre doivent tourner autour des produits et services fournis par l'entreprise. Dans le cas où il existe des questions que le chatbot ne peut pas traiter, l'intervention d'un agent humain sera nécessaire, et ce sera cet humain qui répondra à la place du chatbot.

3.2. Objectif et besoins de l'utilisateur

L'objectif de ce projet est de créer un chatbot (ou agent conversationnel) pour la gestion d'une foire aux questions.

Plus précisément, le projet consiste à créer un chatbot qui va fournir des réponses face aux questions envoyés par les visiteurs sur les produits et services de l'entreprise Youngdev.

Ainsi, les besoins de l'utilisateur sont :

- Gérer la base de connaissances du chatbot (en l'alimentant par les questions et réponses fréquents)
- Afficher la liste des sessions des internautes
- Envoie de réponse automatique provenant du chatbot dans le cas des requêtes simples envoyés par le visiteur
- Fournir des réponses sur les requêtes standard auxquelles le chatbot ne trouve pas de réponses
- Discuter directement avec le visiteur

En termes de besoins techniques, on veut un logiciel de qualité.

3.3. Moyens nécessaires à la réalisation du projet

Pendant la réalisation du projet, des moyens humains, matériels et logiciels ont été utiles afin de le mener à bien.

3.3.1 Moyens matériels

Le Tableau 4 illustre les caractéristiques des matériels utilisés.

Tableau 4 : Caractéristiques des matériels

Nom du PC	Marque	Processeur	Disque dur	RAM	Fréquence	Système d'exploitation
HP	HP	Intel Core i3 7th generation	500 Go	8 Go	2,3 GHz	Windows 10

3.3.2 Moyens humains

Les personnes qui contribuent au développement du projet sont :

- L'encadreur qui est l'initiateur ;
- Le stagiaire qui est à la fois concepteur et programmeur ;

3.3.3 Moyens logiciels

Les logiciels utilisés lors de la réalisation de ce projet sont présentés dans le tableau

5.

Tableau 5 : Moyens logiciels utilisés

LOGICIEL / LIBRAIRIE	VERSION	UTILITE
Angular	11	Framework de développement d'application web
NLP.js (node-nlp)	4.22.1	Algorithme d'intelligence artificielle utile pour la compréhension du langage naturel de l'homme
Socket	4.1.3	Assure en temps réel la mise à jour de donnée
NgRx	12.1.0	Librairie permettant de gérer l'état de l'application

Node.js	14.16.0	Langage utilisé pour le traitement côté back-end
MongoDB	1.26.1	Système de gestion de la base de données de notre application
Visual Paradigm	8.0	Outils de modélisation
Visual Studio Code	1.57.1	Editeur de code

3.4. Résultats attendus

Les résultats attendus de ce présent projet consistent à fournir aux visiteurs un agent conversationnel disponible 24h/24 pour traiter les simples requêtes envoyées par le visiteur sur les différents produits et services de l'entreprise. Dans le cas où les requêtes deviennent complexes, c'est un agent humain qui prend la place du chatbot pour traiter ces types de requêtes.

3.5. Chronogramme des travaux

Le tableau 6 représente le chronogramme des travaux.

Tableau 6 : Chronogramme des travaux

PERIODE	TÂCHES
18 Mars 2021 – 05 avril 2021	Analyse des besoins
06 avril 2021 – 30 Avril 2021	Documentation sur le projet à réaliser
1 ^{er} Mai 2021 – 19 Mai 2021	Conception et modélisation du système à développé
20 Mai 2021 – 25 août 2021	Développement de l'application
25 Août 2021 – 03 septembre 2021	Maintenance de l'application

PARTIE II : ANALYSE ET CONCEPTION

Chapitre 4. ANALYSE PREALABLE

Une étape essentielle de tout cycle de développement logiciel consiste à effectuer une analyse préalable. Le but de cette phase est de savoir l'organisation actuel du système afin de dégager ses points forts ainsi que ses faiblesses et proposer des solutions pour améliorer le système actuel. Nous allons ensuite choisir parmi plusieurs outils qui y seront proposés les outils nécessaires à la création du nouveau système.

4.1. Analyse de l'existant

4.1.1 Organisation actuelle

Pour poser une éventuelle question ou remarque au site web de l'entreprise, un système d'envoi de message est déjà mis en place en avance. Le message envoyé par le visiteur du site web va être reçu et vu par un agent humain dont ce dernier va répondre et puis va envoyer la réponse à l'email fourni par le visiteur en question.

4.1.2 Inventaire des moyens matériels

L'entreprise Youngdev à laquelle j'ai effectué mon stage utilise plusieurs ordinateurs, les caractéristiques de ces ordinateurs sont présentées dans le tableau 7.

Tableau 7 : Caractéristiques des matériels

Désignation	Marque	Disque dur	Processeur	Fréquence	OS	RAM
Ordinateur portable	Mac Book Pro	256 Go SSD	Corei5	2.5GHz	MacOS	16 Go
Ordinateur portable	Mac Book Pro	256 Go SSD	M1		MacOS	8 Go

4.2. Critique de l'existant

Comme évoqué précédemment (voir 4.1.1), un système de messagerie a été mise en place pour répondre aux requêtes des visiteurs, ce système est simple et efficace pour l'envoi de messages.

Cependant ce système présente des inconvénients comme le retard du temps de réponse provenant de l'agent humain puis viens l'indisponibilité pendant quelques temps de cet agent vu qu'il doit être toujours connecté pour discuter avec un visiteur. De plus, il faudra toujours l'intervention d'un agent humain pour répondre à toutes les requêtes envoyées par les visiteurs du site, ce qui peut être coûteux si l'agent doit traiter des tonnes de requêtes provenant de plusieurs visiteurs.

4.3. Conception avant-projet

4.3.1. Proposition de solutions

Nous avons proposé deux solutions afin d'assurer la présence en temps réels et en permanence d'un agent qui pourra répondre instantanément aux requêtes des visiteurs.

- Solution 1 : Engager plusieurs agents humains qui vont être disponibles à tout moment pour pouvoir répondre instantanément à tous les requêtes des visiteurs.
- Solution 2 : Créer un chatbot qui va traiter en temps réels les requêtes simples envoyées par les visiteurs et puis engager l'intervention d'un agent humain pour le traitement des requêtes complexes.

Les avantages et les inconvénients de chaque solution sont présentés dans le tableau 8.

Tableau 8 : Avantages et inconvénients de chaque solution

SOLUTIONS	AVANTAGES	INCONVENIENTS
1	<ul style="list-style-type: none"> ➤ Gère mieux les conversations et requêtes complexes avec le visiteur ➤ Conversation naturelle avec le visiteur 	<ul style="list-style-type: none"> ➤ Ressources humaines mise à disposition coûteux ➤ Temps de réponse peut être longue
2	<ul style="list-style-type: none"> ➤ Disponible à tout moment ➤ Temps de réponse immédiat pour le cas des requêtes simples 	<ul style="list-style-type: none"> ➤ Temps d'attente de réponse peut être longue dans le cas des requêtes complexes

Après avoir comparé ces deux solutions, on a opté pour la deuxième car la première nécessite des ressources humaines assez coûteux.

4.3.2. Méthodes

En effet, la conception d'un système d'information n'est pas évidente parce qu'il faut réfléchir à l'ensemble de l'organisation que l'on doit mettre en place. La phase de conception nécessite des méthodes permettant de mettre en place un modèle sur lequel nous allons appuyer. La modélisation consiste à créer une représentation virtuelle d'une réalité de telle façon à faire ressortir les points auxquels nous nous intéressons. Ce type de méthode est appelé analyse.

Il existe plusieurs méthodes d'analyse, l'une d'elle est la méthode 2TUP dont nous allons utiliser comme méthode de conception. Une brève présentation de cette méthode est évoqué ci-dessous.

Présentation de 2TUP

2TUP signifie « 2 Track Unified Process ». C'est un processus UP.

Le processus 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information de l'entreprise. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes. « 2 Track » signifie littéralement que le processus suit deux chemins. Il s'agit des chemins « fonctionnels » et « d'architecture technique », qui correspondent aux deux axes de changement imposés au système informatique.

4.3.3. Outils proposés

Pour réaliser la solution qui a été adopté précédemment, nous avons besoin des outils suivants :

- Un framework de développement pour la partie interface de l'application
- Un langage de programmation pour mettre en place la partie traitement
- Un SGBD pour la gestion des données
- Un environnement de développement ou un éditeur
- Un outil de modélisation
- Un mode de création de chatbot

a. Choix du Framework

Il existe plusieurs framework pour le développement d'application web. Une brève comparaison de quelques-uns de ces framework est dressée dans le tableau 9.

Tableau 9 : Comparaison de quelques framework Javascript

FRAMEWORK	ANGULAR	REACTJS
Avantages	<ul style="list-style-type: none">• Nouvelles fonctionnalités améliorées telles que RXJS, compilation plus rapide (en moins de 3 secondes)• Documentation détaillée• Liaison de données bidirectionnelle qui minimise les risques d'erreurs possibles• MVVM (Model-View-View Model) qui permet aux développeurs de travailler séparément sur la même section de l'application en utilisant le même ensemble de données• Injection de dépendance des fonctionnalités liées aux composants avec modules et modularité en général• Même si ce framework peut être difficile à apprendre. Il dispose d'une grande communauté qui fournit une quantité de formations et de discussions suffisantes pour	<ul style="list-style-type: none">• Facile à apprendre, en raison de sa simplicité en ce qui concerne la syntaxe• Haut niveau de flexibilité et de réactivité• DOM virtuel (modèle d'objet de document)• Combiné avec ES6 / 7, ReactJS peut travailler avec la charge élevée d'une manière facile.• Librairie JavaScript 100% open source (reçoit beaucoup de mises à jour et d'améliorations quotidiennes)• La migration entre les versions est généralement très facile, avec « codemods », fournit par Google, pour automatiser une grande partie du processus.

	se lancer dans l'utilisation d'AngularJS sans être bloqué.	
Inconvénients	<ul style="list-style-type: none"> • Les problèmes de migration qui peuvent apparaître lors du passage de l'ancienne version aux plus récentes • Les applications dynamiques n'ont pas toujours été aussi performantes. Les single-page application complexes pouvaient être à la traîne et peu pratiques à utiliser en raison de leurs tailles. 	<ul style="list-style-type: none"> • Manque de documentation officielle – le développement ultrarapide de ReactJS ne laisse aucune place à la documentation appropriée • React est <i>unopinionated</i>, ce qui signifie que les développeurs ont parfois trop de choix. • Long à maîtriser, ReactJS nécessite une connaissance approfondie de la façon d'intégrer l'interface utilisateur dans le framework MVC.

Cependant nous avons choisi d'utiliser le framework **ANGULAR** puisqu'il possède une documentation détaillée.

Présentation d'Angular

Angular est un framework applicatif JavaScript développé par Google, qui permet de créer des applications web monopages (c'est-à-dire complètement chargées sur le client avant toute utilisation par l'internaute). Ces applications ne sollicitent le serveur que pour échanger des données (et non pour charger des pages web). Dans le cadre très fréquent d'une architecture MEAN, Angular est couplé avec un serveur Node.js (lui-même enrichi avec le framework Express) et en liaison avec un système de gestion de base de données NoSQL MongoDB.

b. Choix du langage de programmation

Les langages de programmation (côté serveur) à choisir pour développer une application sont nombreuses. Le tableau 10 illustre une comparaison de quelques-uns de ces langages

Tableau 10 : Comparaison entre NodeJS et PHP

LANGAGE	NodeJS	PHP
Avantages	<ul style="list-style-type: none">➤ Prise en charge du multithreading➤ Syntaxe unique pour coté clients et serveurs➤ Prise en charge intégré du serveur➤ Rapide	<ul style="list-style-type: none">➤ Facilité de développement➤ Prend en charge plus de framework et de CMS➤ Pris en charge par tous les services d'hébergement internet
Inconvénients	<ul style="list-style-type: none">➤ Requier une bonne compréhension sur les fonctions de rappel et les blocages➤ Configuration de l'environnement de développement difficile	<ul style="list-style-type: none">➤ Lenteur

Ainsi, nous avons choisi le langage Node.js en raison de sa rapidité.

Présentation de NodeJS

NodeJS est une plateforme de développement Javascript. Ce n'est pas un serveur, ce n'est pas un framework, c'est juste le langage Javascript avec des bibliothèques permettant de réaliser des actions comme écrire sur la sortie standard, ouvrir/fermer des connections réseau ou encore créer un fichier.

Il est souvent confondu avec un serveur car c'est son origine : Node.js a été créé par Ryan Dahl dans le but de pouvoir créer des applications temps réel où le serveur est capable de pousser de l'information au client. C'est dans ce but qu'il utilise la bibliothèque libuv pour réaliser son modèle d'entrée sortie non bloquante.

c. Choix du SGBD

Une brève comparaison de quelques SGBD est dressée dans le tableau 11.

Tableau 11 : Comparaison de quelques SGBD

SGBD	MongoDB	PostgreSQL	MySQL
Avantages	<ul style="list-style-type: none"> ➤ Open Source ➤ Agile et évolutive ➤ Léger ➤ Donnée non structuré 	<ul style="list-style-type: none"> ➤ Meilleure prise en charge des données volumineuses ➤ Open Sources ➤ Interface ODBC et JDBC ➤ Dispose d'un dispositif de contrôle de l'intégrité de la structure et de la cohérence des données ➤ Basé sur un modèle orienté objet 	<ul style="list-style-type: none"> ➤ Open Sources ➤ Interface ODBC ➤ Facile à déployer, rapide et performant ➤ Très populaire
Inconvénients	<ul style="list-style-type: none"> ➤ Données brutes ➤ NoSQL 	<ul style="list-style-type: none"> ➤ Consomme beaucoup de ressources 	<ul style="list-style-type: none"> ➤ Ne supporte pas des bases de données volumineuses ➤ Seulement basé sur un modèle relationnel classique

Ainsi, nous avons choisi **MongoDB** en raison de sa légèreté.

Présentation de MongoDB

MongoDB est un système de gestion de bases de données open source développé par MongoDB Inc depuis 2007. Il fait partie de la mouvance NoSQL. Son nom vient de l'anglais "humongus" qui peut être traduit par "énorme". L'objectif est donc de pouvoir gérer de très grandes quantités de données. MongoDB est réparti sur un nombre quelconque d'ordinateurs, on parle de scaling. MongoDB est orienté document et ne nécessite pas de schémas prédéfinis de données.

d. Choix de l'environnement de développement (IDE)

Le tableau 12 illustre une comparaison de quelques environnements de développement.

Tableau 12 : Comparaison entre Visual Studio Code et Webstorm

ENVIRONNEMENT DE DEVELOPPEMENT	VISUAL STUDIO CODE	WEB STORM
Avantages	<ul style="list-style-type: none">– Rapide– Léger	<ul style="list-style-type: none">– Dispose de nombreux plugins installés– Riche en fonctionnalités
Inconvénients	<ul style="list-style-type: none">– Nécessite l'installation de certains plugins– Fonctionnalités limitées	<ul style="list-style-type: none">– Consomme beaucoup de ressources– Lent

Ainsi, nous avons choisi **Visual Studio Code** en raison de sa rapidité à fonctionner.

Présentation de Visual Studio Code

Visual Studio Code (VSC par la suite) est un éditeur de code open-source, gratuit et multiplateforme (Windows, Mac et Linux), développé par Microsoft, à ne pas confondre avec Visual Studio, l'IDE propriétaire de Microsoft. VSC est développé avec Electron et exploite des fonctionnalités d'édition avancées du projet Monaco Editor. Principalement conçu pour le développement d'application avec Javascript, TypeScript et Node.js, l'éditeur peut s'adapter à d'autres types de langages grâce à un système d'extension bien fourni

e. Choix de l'outils de modélisation

Le tableau 13 illustre une brève comparaison de quelques outils de modélisation.

Tableau 13 : Comparaison entre UML Designer et Visual Paradigm for UML

Nom	Editeur	Plateforme	Intégré ou autonome	Coût/Licence	Ergonomie	Type de graphiques pris en charges	Autre
UML Designer	Obeo	Multiplateforme	Intégrable dans Eclipse	Open Source EPL	Ouverte	<ul style="list-style-type: none"> -Cas d'utilisation -Classe -Collaboration -Déploiement -Séquence -Etat -Activité -Composant 	Possible de créer ses propres diagrammes
Visual Paradigm for UML	Visual Paradigm	Multi Plateforme	Intégrable dans Eclipse, Netbeans, IntelliJ	Libre	Dispose de différentes fonctions facilitant la conception des diagrammes aux débutants	<ul style="list-style-type: none"> -Cas d'utilisation -Classe -Objet -Etat -Activité -Séquence -Communication -Composant -Déploiement -package -Structure composite -chronogramme 	<ul style="list-style-type: none"> -Génération de code source à partir des diagrammes -Relation du mapping objet relationnel automatiquement à partir des diagrammes de classes

Nous avons alors choisi **Visual Paradigm for UML** parce que c'est un outil complet et d'une facilité d'utilisation

Présentation de Visual Paradigm

Le logiciel de modélisation UML Visual Paradigm for UML est un bon outil pour réaliser des spécifications. Possédant de nombreuses fonctionnalités dans ses versions payantes, il possède une interface intuitive et est rapidement pris en main.

f. Choix de du mode de création de chatbot

En effet, il existe deux types de chatbot :

- **Chatbot simple ou scripté** ayant une capacité de compréhension faible, mais qui permettent de gérer des interactions simples. Ces Chatbots fonctionnent sur le principe de mots clés, et de scénarios de conversations très scriptés (le but est de guider les utilisateurs).
- **Chatbot complexe** qui permettent des interactions plus riches, via un meilleur moteur d'analyse sémantique (compréhension des demandes des clients via le contexte et les synonymes, et non pas une analyse par mots clés), et la possibilité de lancer des actions (ex: passer une commande, gérer un bon de retour...). Ce type de chatbot utilise l'intelligence artificiel.

Le tableau 14 illustre les avantages et les inconvénients entre un chatbot simple et un chatbot complexe.

Tableau 14 : Comparaison entre un chatbot simple et un chatbot complexe

	CHATBOT SIMPLE	CHATBOT COMPLEXE
Avantages	<ul style="list-style-type: none">- Simple à mettre en place- Basé sur le principe de mots clés	<ul style="list-style-type: none">- Permet d'effectuer une analyse sémantique- Permet de détecter l'intention de l'interlocuteur- S'entraîne automatiquement- Intelligent
Inconvénients	<ul style="list-style-type: none">- Ne permet pas de détecter l'intention de l'interlocuteur- Utilise sur la règle « Si... alors... »	<ul style="list-style-type: none">- Difficile à mettre en place- Requiert l'utilisation d'un algorithme d'intelligence artificiel

	<ul style="list-style-type: none"> - Pas d'apprentissage automatique - N'est pas intelligent 	
--	--	--

Nous avons choisi d'opter pour un chatbot complexe car ce type de chatbot nous permettra de faire une analyse sémantique du message envoyé par l'interlocuteur

Présentation d'un Chatbot complexe

Les chatbots alimentés par un logiciel d'intelligence artificielle (IA) sont plus complexes que les chatbots basés sur des règles. Ces robots sont dynamiques et ne s'appuient pas sur des boutons d'appel à l'action pour cartographier les conversations avec leurs visiteurs. Il existe en fait deux types de robots IA.

➤ Chatbot utilisant le langage naturel (NLP)

Le premier type de bot d'IA repose sur le traitement du langage naturel (NLP) pour comprendre le contexte et l'intention du visiteur quelque chose avec lequel les machines ont généralement du mal à percevoir

En ce qui concerne les machines, les humains ne parlent pas de manière logique. Nous utilisons des expressions idiomatiques, de l'argot, parfois des mots mal orthographiés, et nous nous exprimons différemment. Les machines, en revanche, nécessitent une structure, des détails et un processus.

Le NLP aide essentiellement les machines à comprendre le langage humain.

➤ Chatbot utilisant l'apprentissage automatique (machine learning)

Les chatbots d'apprentissage automatique sont similaires aux robots NLP, cependant, ils sont optimisés pour en savoir plus sur le visiteur, conserver des informations sur lui et prédire les prochaines étapes d'une conversation.

Ces robots utilisent des réseaux de neurones artificiels, qui agissent comme un cerveau artificiel pour alimenter de grands ensembles de données. En ce qui concerne les chatbots, ces ensembles de données sont généralement des conversations et des questions précédentes qui peuvent aider le bot à apprendre.

Comme les chatbots NLP, les chatbots d'apprentissage automatique créent une expérience personnalisée pour le visiteur. Les deux robots sont communément appelés « robots intelligents ».

Dans ce présent projet nous avons choisi d'utiliser un chatbot utilisant le NLP car utiliser celui-ci est la condition minimale requise pour la mise en place de notre projet.

Présentation de NLP

Le traitement du langage naturel (NLP) est largement défini comme la manipulation électronique du langage naturel, comme la parole et le texte, par un logiciel.

Le NLP est importante car nous voulons ouvrir la communication entre les machines et les humains de manière plus naturelle. Le NLP a divers cas d'utilisation tels que l'exécution d'un moteur de recherche, l'analyse sentimentale, la reconnaissance d'entités, les applications vocales, les chatbots et les assistants personnels.

Cette technologie peut traiter le langage humain en entrée et effectuer une ou plusieurs des opérations suivantes :

- [Analyse des sentiments](#) (Est-ce une déclaration positive ou négative ?)
- Classification des sujets (De quoi s'agit-il ?)
- Décidez des actions à entreprendre sur la base de cette déclaration
- [Extraction de l'intention](#) (Quelle est l'intention derrière cette déclaration ?)

Le traitement du langage naturel (NLP) est un sous-domaine de la [linguistique](#) , de l' [informatique](#) , de l' [ingénierie de l'information](#) et de l'[intelligence artificielle](#) qui s'intéresse aux interactions entre les ordinateurs et les langues humaines (naturelles), en particulier la façon de programmer des ordinateurs pour traiter et analyser de grandes quantités de données de [langage naturel](#) . .

Il existe de nombreuses bibliothèques NLP basé sur le langage javascript, dans ce présent projet j'ai choisi d'utiliser **NLP.js (node-nlp)**.

Présentation de NLP.js

NLP.js est développé par le groupe AXA. Il s'agit d'une bibliothèque NLP pour la création de robots, avec extraction d'entités, analyse des sentiments, identification automatique de la langue, etc., prend en charge 40 langues.

NLP.js est une bibliothèque node.js parfaite pour créer des chatbots. La documentation est très claire et l'utilisation est très facile.

Chapitre 5. ANALYSE CONCEPTUELLE

Dans cette partie, nous allons établir la conception de base de l'application à développer.

5.1. Dictionnaire des données

Un dictionnaire des données recense et décrit l'ensemble des propriétés utilisées pour élaborer le diagramme de classe. Celle-ci a été remplie à partir d'un recensement des informations dans les documents utilisés.

Le tableau 15 présente la liste de toutes les données manipulées par l'application :

Tableau 15: Dictionnaire des données

Nom de la rubrique	Description	Type	Taille	Observation
auteurReponse	Auteur du réponse (généralement c'est l'agent)	AN	20	
auteurRequete	Auteur du requête (généralement c'est l'internaute)	AN	20	
botName	Nom de l'agent conversationnel	AN	35	
botReply	Permet de savoir si la réponse automatique provenant du robot conversationnel est activée ou non pour un internaute donné	Booléen	5	
botTask	Description de la tâche de l'agent	AN	200	
chatbot_id	Identifiant de l'agent conversationnel	AN	20	
contenuReponse	Contenu de la réponse fournit par un acteur précis	AN	255	

contenuRequete	Contenu du requête	AN	255	
dateEnvoie	Date d'envoi du message	Date		jj-mm-aaaa
descIntention	Description de l'intention	AN	100	
email	Email de l'internaute ou du visiteur (et aussi son identifiant)	AN	30	
file	Fichier inclut dans le message envoyé par l'agent conversationnel vers l'internaute	AN	40	
imgIntention	Image illustrant l'intention	AN	30	
intention_id	Identifiant de l'intention	AN	30	
internaute_id	Identifiant de l'internaute	AN	20	
localisation	Localisation de l'internaute	AN	30	
message	Contenu du message échangé	AN	255	
nomIntention	Nom de l'intention	AN	30	
nomInternaute	Prénom de l'internaute ou du visiteur	AN	30	
reponse_id	Identifiant du réponse	AN	20	
requete_id	Identifiant du requête	AN	20	
typeIntention	Type de l'intention (peut être native, standard et spécifique)	AN	20	
userEmail	Email de l'utilisateur de la page d'administration de l'agent conversationnel	AN	30	
userPwd	Mot de passe de l'utilisateur de la page d'administration de l'agent conversationnel	AN	30	

AN : Alphanumérique
N : Numérique
L : Logique
D : Date

5.2. Règles de gestion

Les règles de gestion correspondent souvent aux normes qu'il faut suivre afin que les données s'harmonisent.

Cependant pour notre projet, il y a quelques règles de gestion à prendre en considération dont :

- ❖ **RG1** : Une intention peut avoir un ou plusieurs requêtes et un ou plusieurs réponses
- ❖ **RG2** : Un message ne correspond qu'à un seul et une seule intention
- ❖ **RG3** : Une requête n'appartient qu'à un seul et une seule intention
- ❖ **RG4** : Une réponse n'appartient qu'à un seul et une seule intention
- ❖ **RG5** : Le nom d'une intention est unique

5.3. Représentation et spécification des besoins

La spécification de besoins constitue la phase de départ de toute application à développer dans laquelle nous allons identifier les besoins fonctionnels de notre application.

5.3.1. Diagramme des cas d'utilisation

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

La représentation d'un cas d'utilisation met en jeu trois concepts : l'acteur, le cas d'utilisation et l'interaction entre l'acteur et le cas d'utilisation.

Acteur

Un acteur est un utilisateur type qui a toujours le même comportement vis-à-vis d'un cas d'utilisation. Ainsi les utilisateurs d'un système appartiennent à une ou plusieurs classes d'acteurs selon les rôles qu'ils tiennent par rapport au système.

Une même personne physique peut se comporter en autant d'acteurs différents que le nombre de rôles qu'elle joue vis-à-vis du système.

Ainsi par exemple, l'administrateur d'un système de messagerie peut être aussi utilisateur de cette même messagerie. Il sera considéré, en tant qu'acteur du système, dans le rôle d'administrateur d'une part et dans celui d'utilisateur d'autre part.

Un acteur peut aussi être un système externe avec lequel le cas d'utilisation va interagir.

Formalisme et exemple

Un acteur peut se représenter symboliquement par un « bonhomme » et être identifié par son nom. Il peut aussi être formalisé par une classe stéréotypée « acteur ».

La figure 6 représente ce qu'est un acteur.

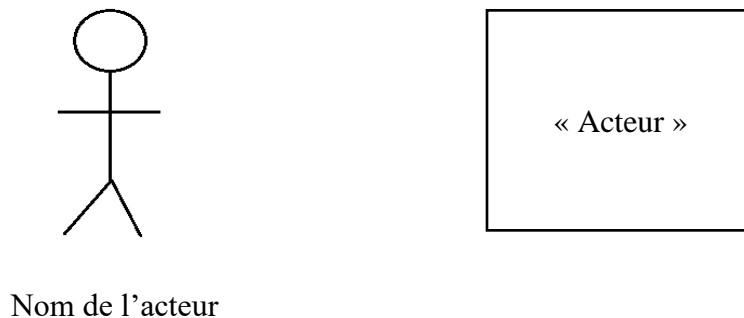


Figure 6 : Représentations d'un acteur

Cas d'utilisation et interaction

Un cas d'utilisation correspond à un certain nombre d'actions que le système devra exécuter en réponse à un besoin d'un acteur. Un cas d'utilisation doit produire un résultat observable pour un ou plusieurs acteurs ou parties prenantes du système.

Une interaction permet de décrire les échanges entre un acteur et un cas d'utilisation.

Formalisme et exemple

Un cas d'utilisation se représente par un ovale dans lequel figure son intitulé.
L'interaction entre un acteur et un cas d'utilisation se représente comme une association.
Elle peut comporter des multiplicités comme toute association entre classes

Le formalisme de base de représentation d'un cas d'utilisation est donné à la figure 7.

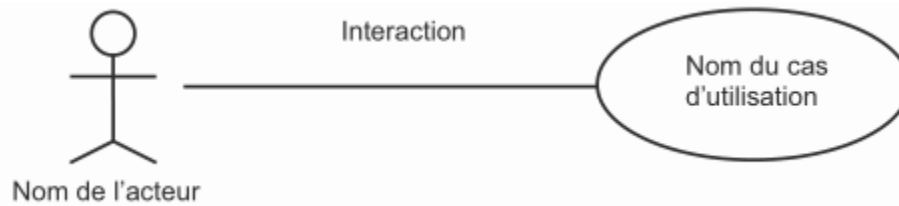


Figure 7 : Formalisme de base de représentation d'un cas d'utilisation

Cependant en revenant dans notre domaine d'étude, les acteurs qui interagissent avec le système sont **l'internaute** et **l'administrateur (agent humain)**.

Le tableau 16 illustre les informations détaillées concernant ces acteurs :

Tableau 16 : Informations détaillées de chaque acteur

Acteur(s)	Internaute (ou client ou visiteur)	Administrateur (ou agent humain)
Rôle	Envoie les questions à l'agent conversationnel	Administre l'agent conversationnel
Critère de satisfaction	Rapidité, fiabilité	Rapidité, performance
Nombre d'instance	20 000	100
Fréquence d'utilisation	Quotidienne	Quotidienne
Connaissance technologique	Moyenne	Moyenne
Connaissance métier	Moyenne	Moyenne
Description	C'est celui qui surfe sur internet et c'est celui qui	C'est celui qui va gérer la base de connaissance de l'agent conversationnel

	visite le site web de l'entreprise	
--	------------------------------------	--

Le tableau 17 illustre la liste des cas d'utilisation répertorié dans le domaine d'étude.

Tableau 17 : Liste des cas d'utilisation

CAS D'UTILISATION	ACTEURS		MESSAGES	
S'authentifier	PRINCIPAL	Administrateur	EMIS	Consulter la page d'administration du chatbot
			REÇU	Demande de saisie de l'adresse email et du mot de passe
Afficher statistique	PRINCIPAL	Administrateur	EMIS	Consulter la page de tableau de bord
			REÇU	Affichage du page tableau de bord
Créer intention	PRINCIPAL	Administrateur	EMIS	Demande de création d'une intention
			REÇU	Affichage de la page de création d'intention
Mettre à jour intention	PRINCIPAL	Administrateur	EMIS	Modifier les informations concernant un intention
			REÇU	Affichage de la page de mise à jour d'une intention
Discuter avec l'internaute	PRINCIPAL	Administrateur	EMIS	Demander la page de discussion en privée
			REÇU	Affichage de la page de discussion en privée

	SECONDAIRE	Internaute	REÇU	Réception du message provenant de l'administrateur
Répondre aux requêtes non comprises par le chatbot	PRINCIPAL	Administrateur	EMIS	Demander la page d'affichage des requêtes non comprises par le chatbot
			REÇU	Affichage de la page illustrant les requêtes non comprises par le chatbot
	SECONDAIRE	Internaute	REÇU	Réception de la réponse provenant de l'agent humain au requête non comprise par le chatbot
Poser question	PRINCIPAL	Internaute	EMIS	Demander la zone de saisie pour l'envoi de questions
			REÇU	Affichage de la zone d'envoi de questions
	SECONDAIRE	Administrateur (dans le cas où la requête n'a pas été compris par le chatbot)	REÇU	Réception du message envoyé par l'internaute
Visualiser la liste des internautes	PRINCIPAL	Administrateur	EMIS	Demander la liste des internautes ayant effectué des sessions avec le chatbot

			REÇU	Réception de la liste de ces internautes
--	--	--	------	--

La figure 8 illustre le diagramme des cas d'utilisation globale du domaine d'étude.

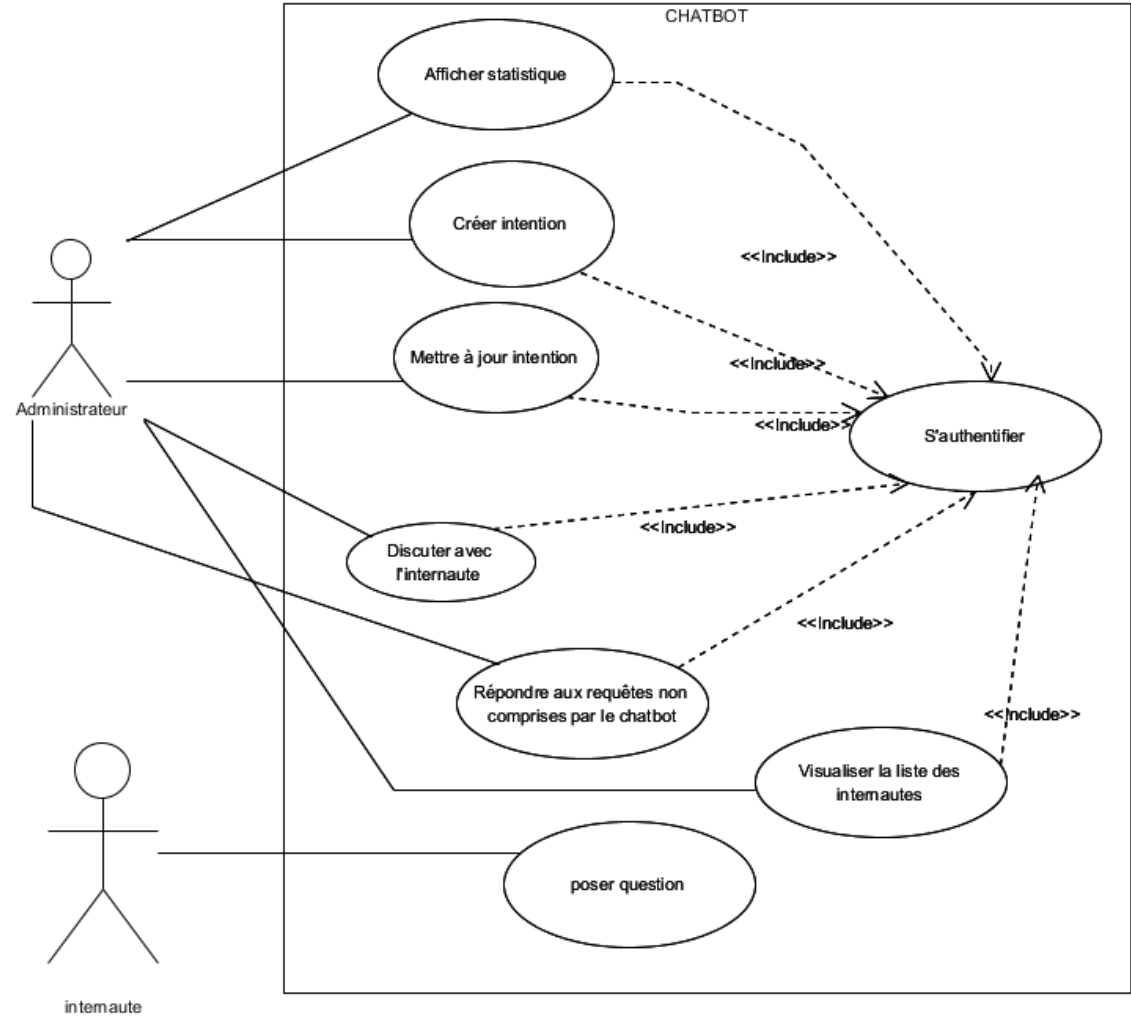


Figure 8 : Diagramme des cas d'utilisation globale

Chaque cas d'utilisation doit être décrit sous forme textuelle afin de bien identifier les traitements à réaliser par le système en vue de la satisfaction du besoin exprimé par l'acteur.

Nous allons cependant décrire textuellement chaque cas d'utilisation.

Description textuelle du cas d'utilisation « S'AUTHENTIFIER »

- + **Acteur principal** : Administrateur
- + **Précondition** : L'administrateur s'authentifie
- + **Début** : L'administrateur consulte la page d'authentification
- + **Post condition** : L'administrateur finalise son authentification
- + **Fin** : L'administrateur est redirigé vers la page d'accueil
- + **Scénario nominal** :

Administrateur	Système
1. L'administrateur saisi les informations pour l'authentification	2. Le système traite et vérifie si les données saisies sont correctes 3. Le système redirige l'utilisateur vers la page d'accueil

+ **Scenario alternative**

- A1** : Email ou mot de passe incorrecte
 - Démarrage à l'étape 1

Description textuelle du cas d'utilisation « Afficher statistique »

- + **Acteur principal** : Administrateur
- + **Précondition** : L'administrateur souhaite voir le statistique des informations traités par le système
- + **Début** : L'administrateur consulte la page statistique ou tableau de bord
- + **Post condition** : L'administrateur est redirigé vers la page de tableau de bord
- + **Fin** : Les informations a affiché dans la page statistique ont été afficher avec succès à l'administrateur
- + **Scénario nominal** :

Administrateur	Système
1. L'administrateur consulte la page statistique	2. Le système redirige l'administrateur vers la page concernée 3. Le système collecte les données à visualiser 4. Le système affiche ses données une fois visualisées

Description textuelle du cas d'utilisation « Créer intention »

- + **Acteur principal** : Administrateur
- + **Précondition** : L'administrateur souhaite créer une nouvelle intention
- + **Début** : L'administrateur consulte la page de création d'intention
- + **Post condition** : L'administrateur saisie les informations nécessaires puis finalise tout.
- + **Fin** : L'administrateur a été notifié de la création de l'intention qu'il a créé
- + **Scénario nominal** :

Administrateur	Système
1. L'administrateur saisie puis valide les informations relatives à la nouvelle intention	3. Le système vérifie s'il n'y a pas d'intention existant qui a le même nom que la nouvelle intention à créer 2. Le système traite et enregistre les données saisies 4. Le système notifie l'administrateur que l'intention a été créé avec succès

+ **Scenario alternative**

- A1 : Le nom de l'intention existe déjà
 - Démarrage à l'étape 1

Description textuelle du cas d'utilisation « Mettre à jour intention »

- + **Acteur principal** : Administrateur
- + **Précondition** : L'administrateur souhaite modifier des informations relatives à une intention
- + **Début** : L'administrateur modifie quelques informations
- + **Post condition** : L'administrateur finalise les modifications relatives à l'intention
- + **Fin** : L'administrateur quitte la page
- + **Scénario nominal** :








Administrateur	Système
1. L'administrateur sélectionne l'intention à mettre à jour 3. L'administrateur met à jour les informations relatives à l'intention et puis valide ces mises à jour	2. Le système redirige l'administrateur vers la page affichant les informations concernant l'intention sélectionné 4. Le système vérifie les informations modifiées. 5. Le système enregistre les informations modifiées relative à l'intention. 6. Le système notifie l'administrateur que l'intention sélectionnée a été mis à jour avec succès.

Scenario alternative

A1 : La nouvelle nom de l'intention (si elle a été modifié) est déjà associé à une intention existante

➤ Démarrage à l'étape 1

Description textuelle du cas d'utilisation « Discuter avec l'internaute »

-  **Acteur principal :** Administrateur
-  **Acteur secondaire :** Internaute
-  **Précondition :** L'administrateur souhaite répondre personnellement aux messages envoyés par l'internaute
-  **Début :** L'administrateur sélectionne l'internaute auquel il va répondre
-  **Post condition :** L'administrateur a fini de discuter avec l'internaute
-  **Fin :** L'administrateur quitte la page après avoir terminé la discussion
-  **Scénario nominal :**

Client	Système
1. L'administrateur consulte la page de discussion 3. L'administrateur sélectionne l'internaute avec qui il va discuter 5. L'administrateur envoie un message privé à l'internaute concerné	2. Le système affiche la page de discussion 4. Le système affiche la zone de discussion privée avec l'internaute 6. Le système enregistre le message à envoyer 7. Le système envoie à l'internaute concerné le nouveau message envoyé par l'administrateur 6. Le système notifie l'administrateur que le message a été envoyé avec succès vers l'internaute.

Description textuelle du cas d'utilisation « Répondre aux requêtes non comprises par le chatbot »

- + **Acteur principal** : Administrateur
- + **Acteur secondaire** : Internaute
- + **Précondition** : L'administrateur souhaite apporter des réponses aux requêtes non comprises par le chatbot
- + **Début** : L'administrateur consulte la page visualisation les requêtes incomprises par le chatbot
- + **Post condition** : L'administrateur répond à un ou plusieurs requêtes
- + **Fin** : L'administrateur quitte la page après avoir répondu à un ou plusieurs requêtes
- + **Scénario nominal** :

Administrateur	Système
<p>1. L'administrateur consulte la page de visualisation des requêtes incomprises par le chatbot</p> <p>4. L'administrateur sélectionne la requête auquel il va apporter une réponse</p> <p>6. L'administrateur associe une réponse à la requête sélectionnée puis finalise son action</p>	<p>2. Le système collecte toutes les requêtes auquel le chatbot n'a pas pu trouver de réponse</p> <p>3. Le système affiche la liste de ses requêtes</p> <p>5. Le système affiche les informations concernant la requête sélectionnée</p> <p>7. Le système enregistre la réponse saisie par l'administrateur pour la requête sélectionné</p> <p>8. Le système notifie à l'administrateur que la réponse a été associé à la requête.</p> <p>9. Le système envoie la réponse associée au requête non comprise par le chatbot à l'internaute qui est l'auteur de la requête</p>

Description textuelle du cas d'utilisation « Poser question »

- + **Acteur principal** : Internaute
- + **Acteur secondaire** : Administrateur
- + **Précondition** : L'internaute souhaite poser des questions sur l'entreprise
- + **Début** : L'internaute consulte le chatbot pour pose des questions
- + **Post condition** : L'internaute attend la réponse à sa question
- + **Fin** : L'internaute envoie sa question et puis attend la réponse provenant du chatbot
- + **Scénario nominal** :

Internaute	Système
1. L'internaute pose une question	2. Le système traite la question posée par l'internaute 3. Le système recherche une réponse appropriée à la question posée par l'internaute. 4. Le système enregistre le message échangé. 5. Le système envoie la réponse sélectionnée vers l'internaute

Scenario alternative

A1 : Le système ne trouve aucune réponse associée au question posée par l'internaute.

- Le système envoie la requête incomprise par le chatbot vers l'administrateur
- Le système notifie l'internaute de son regret pour l'incompréhension de sa requête
- Le système vérifie si l'internaute a déjà fourni son adresse email
 - **Si adresse email NON-FOURNI,**
 - Le système notifie l'internaute que sa requête va être répondu dans le plus bref délai par un agent humain si sa requête a été bien validé.
 - Le système demande un adresse email à l'internaute (si ce dernier ne l'a pas encore fourni) afin de le notifier de la réponse à sa requête une fois répondu.
 - Si l'internaute fourni l'adresse email alors le système enregistre l'email et puis notifie à l'internaute que l'adresse email a été enregistré avec succès

Description textuelle du cas d'utilisation « Visualiser la liste des internautes »

- + **Acteur principal** : Administrateur
- + **Précondition** : L'administrateur souhaite voir la liste des internautes ayant interagit avec le chatbot
- + **Début** : L'administrateur consulte la page d'affichage de la liste des internautes
- + **Post condition** : La liste des internautes a été affiché à l'administrateur
- + **Fin** : L'administrateur a fini de consulter la page affichant liste des internautes
- + **Scénario nominal** :

Administrateur	Système
1. L'administrateur consulte la page affichant la liste des internautes ayant interagit avec le chatbot	3. Le système collecte la liste de ces internautes 2. Le système affiche la liste correspondante

5.3.1. Priorisation des cas d'utilisation

Puisque nous venons d'élaborer le diagramme des cas d'utilisations, nous allons maintenant déterminer la priorité de chacun de ces cas d'utilisation.

Le tableau 18 indique la priorité de chaque cas d'utilisation (orientés selon les risques).

Tableau 18 : Priorité de chaque cas d'utilisation

CAS D'UTILISATION	PRIORITE
Répondre aux requêtes non comprises par le chatbot	1
Discuter avec l'internaute	2
Poser question	1
Créer intention	1
Mettre à jour intention	1
Visualiser la liste des internautes	2
Afficher statistique	2
S'authentifier	2

1 : Indispensable

2 : Moyen

3 : Optionnel

5.3.2. Diagramme de séquences système pour chaque cas d'utilisation

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.

Un diagramme de séquence se représente globalement dans un grand rectangle avec indication du nom du diagramme en haut à gauche.

La figure 9 présente le formalisme d'un diagramme de séquence.

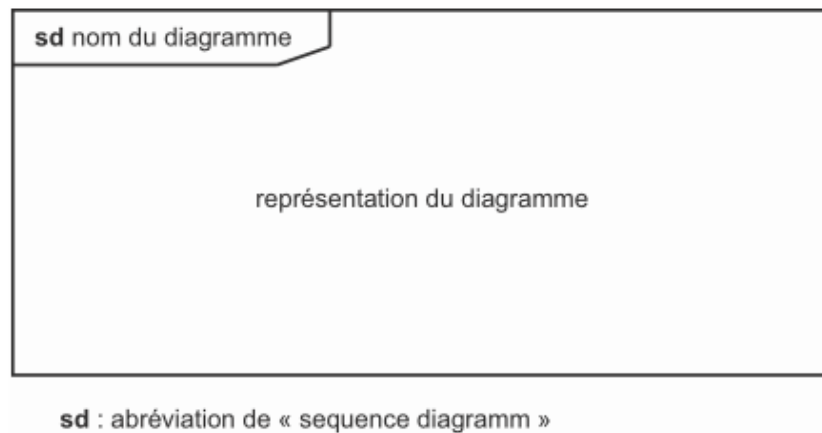


Figure 9 : Formalisme général du cadre d'un diagramme de séquence

Ligne de vie

Une ligne de vie représente l'ensemble des opérations exécutées par un objet. Un message reçu par un objet déclenche l'exécution d'une opération. Le retour d'information peut être implicite (cas général) ou explicite à l'aide d'un message de retour.

Maintenant nous allons représenter le diagramme de séquence système de chaque cas d'utilisation de notre domaine d'étude.

❖ **Diagramme de séquence système du cas d'utilisation « S'authentifier »**

La figure 10 représente le diagramme de séquence système du cas d'utilisation « S'authentifier ».

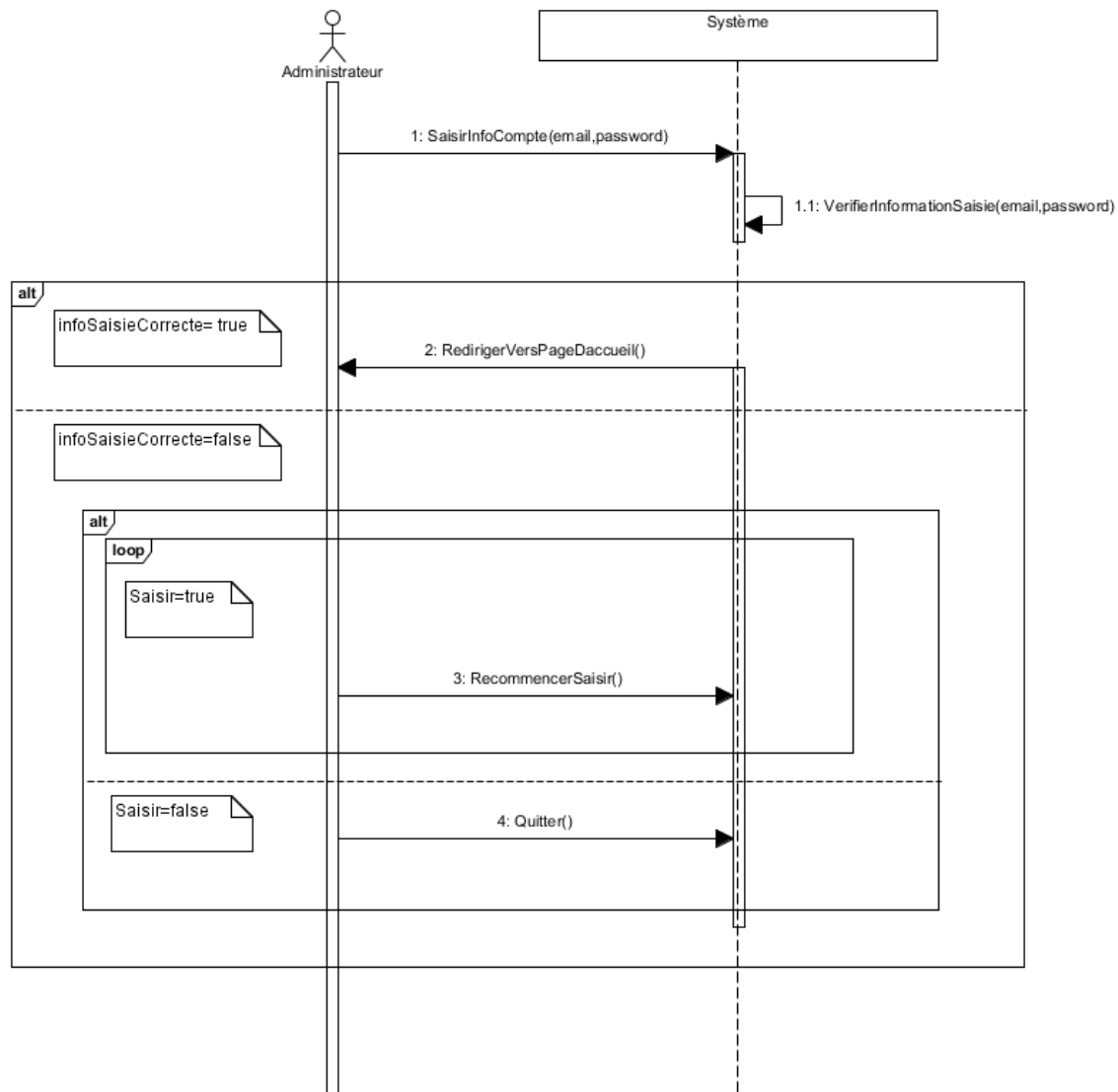


Figure 10 : Diagramme de séquence système du cas d'utilisation « S'authentifier »

❖ **Diagramme de séquence système du cas d'utilisation « Afficher statistique »**

La figure 11 représente le diagramme de séquence système du cas d'utilisation « Afficher statistique ».

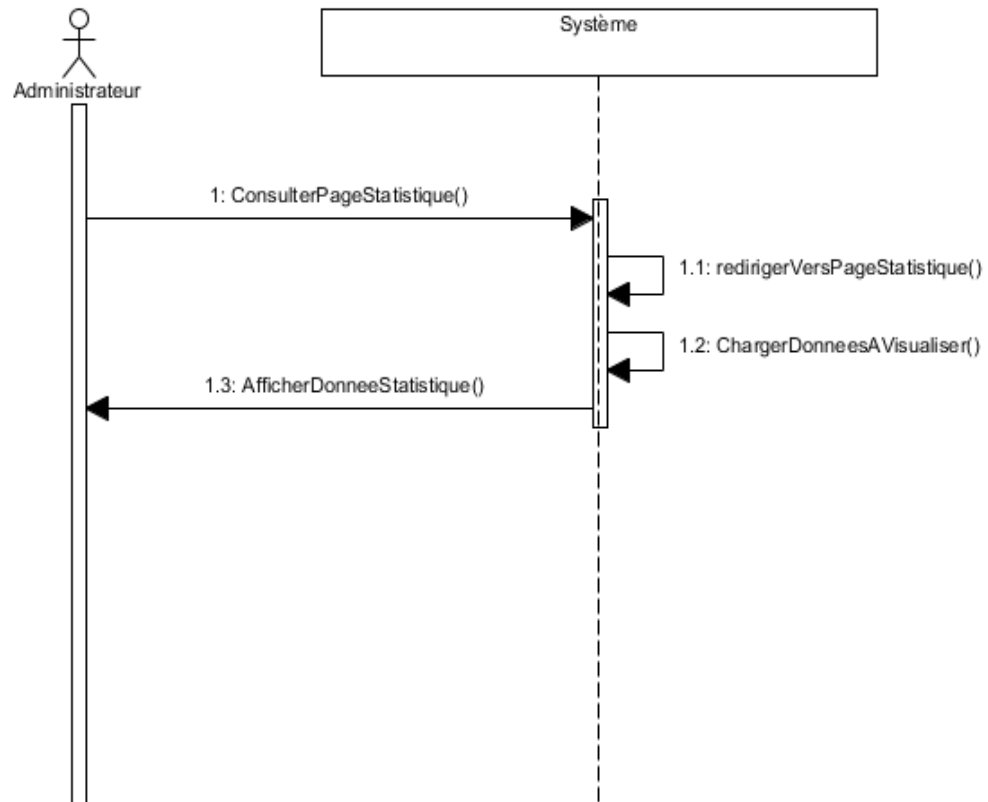


Figure 11 : Diagramme de séquence système du cas d'utilisation « Afficher statistique »

❖ **Diagramme de séquence système du cas d'utilisation « Créer intention »**

La figure 12 représente le diagramme de séquence système du cas d'utilisation « Créer intention ».

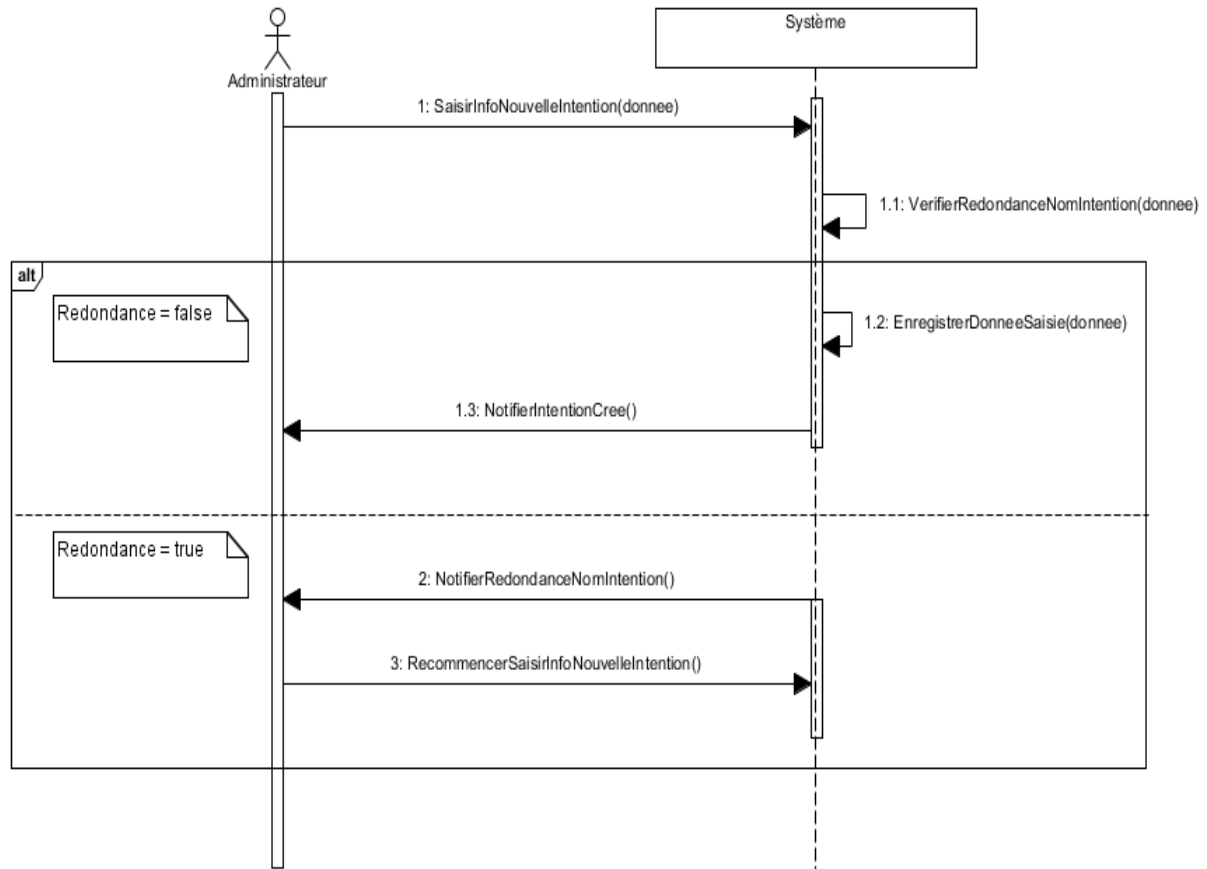


Figure 12 : Diagramme de séquence système du cas d'utilisation « Créer intention »

❖ **Diagramme de séquence système du cas d'utilisation « Mettre à jour intention »**

La figure 13 représente le diagramme de séquence système du cas d'utilisation « Mettre à jour intention ».

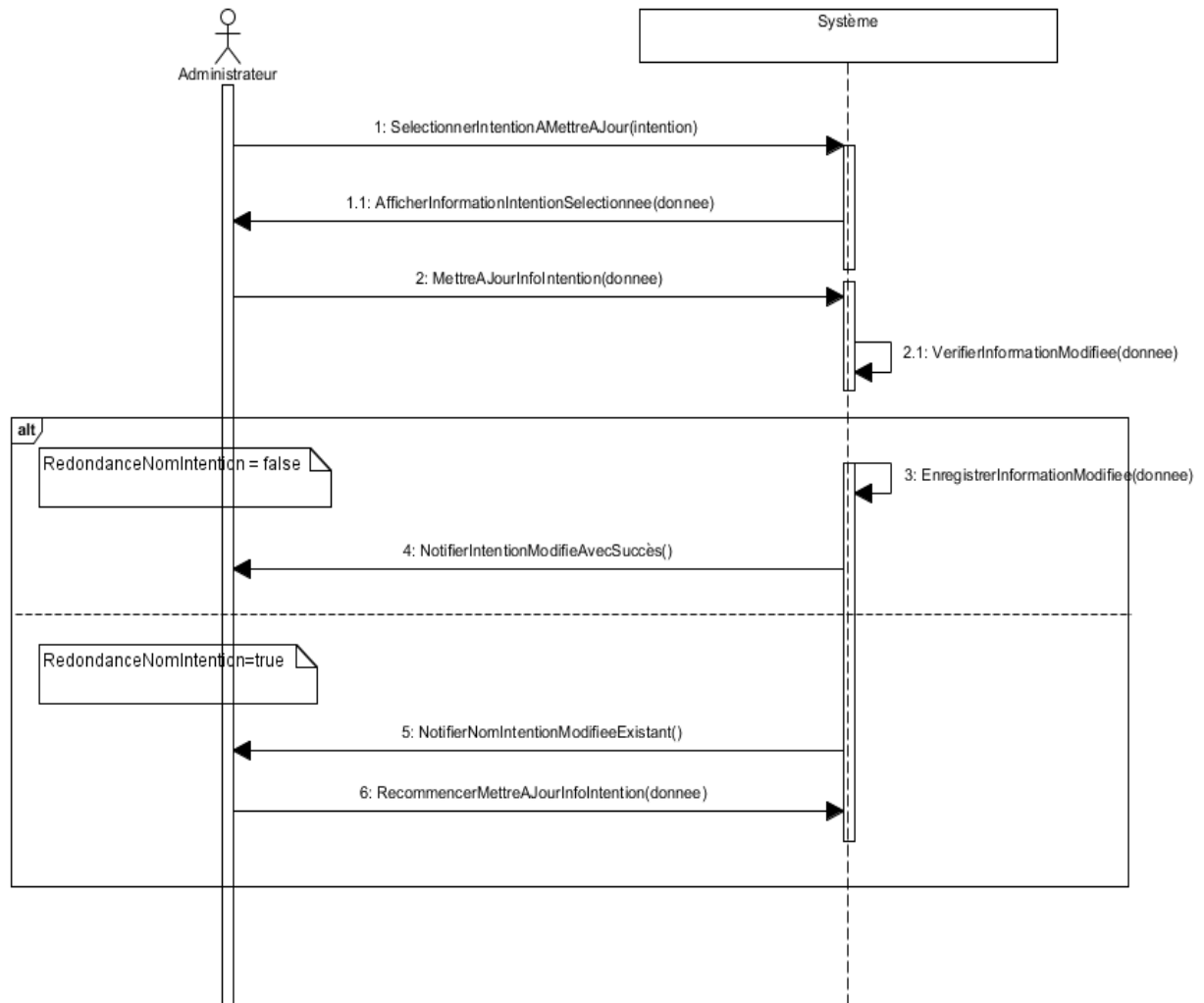


Figure 13 : Diagramme de séquence système du cas d'utilisation "Mettre à jour intention"

❖ **Diagramme de séquence système du cas d'utilisation « Discuter avec l'internaute »**

La figure 14 représente le diagramme de séquence système du cas d'utilisation « Discuter avec l'internaute ».

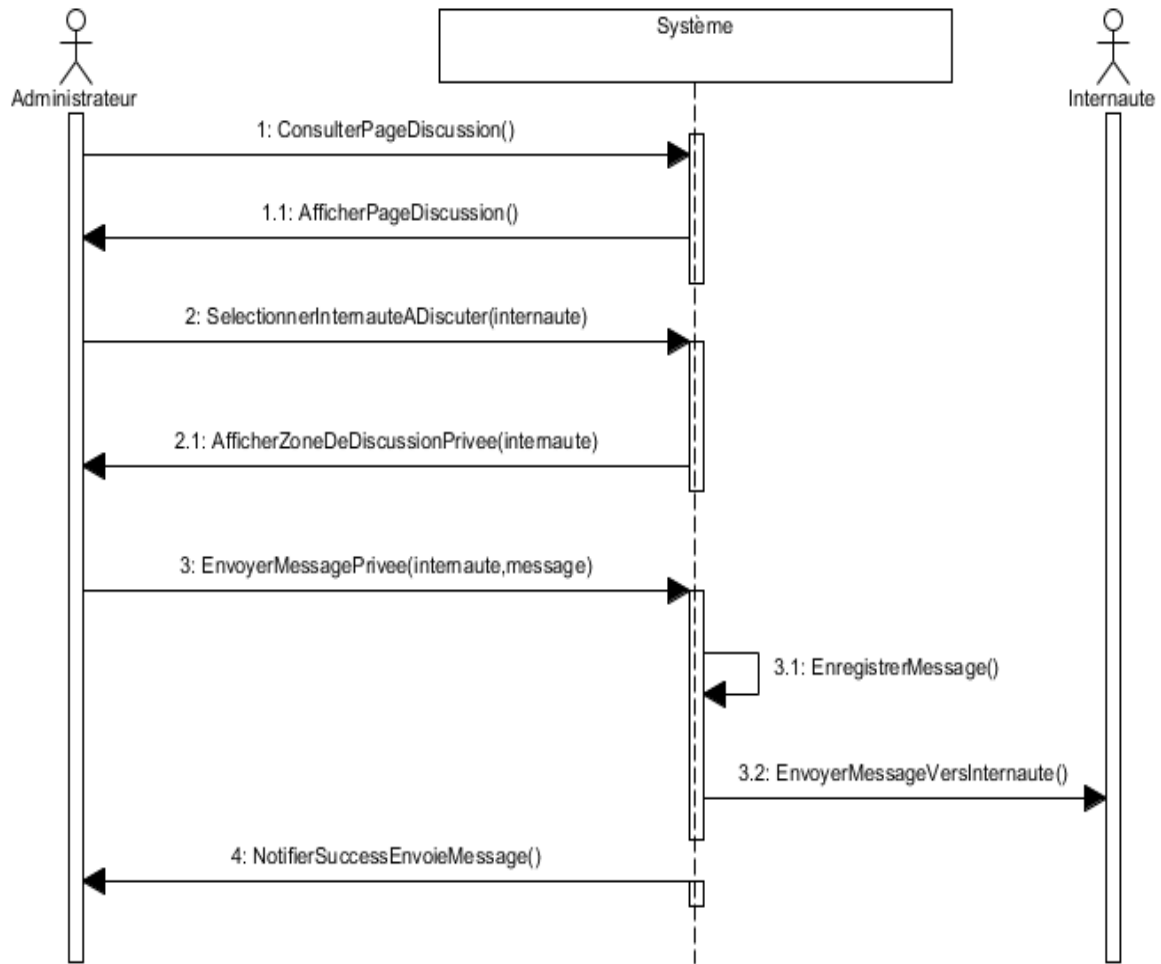


Figure 14 : Diagramme de séquence système du cas d'utilisation "Discuter avec l'internaute"

❖ **Diagramme de séquence système du cas d'utilisation « Répondre aux requêtes non comprises par le chatbot »**

La figure 15 représente le diagramme de séquence système du cas d'utilisation « Répondre aux requêtes non comprises par le chatbot ».

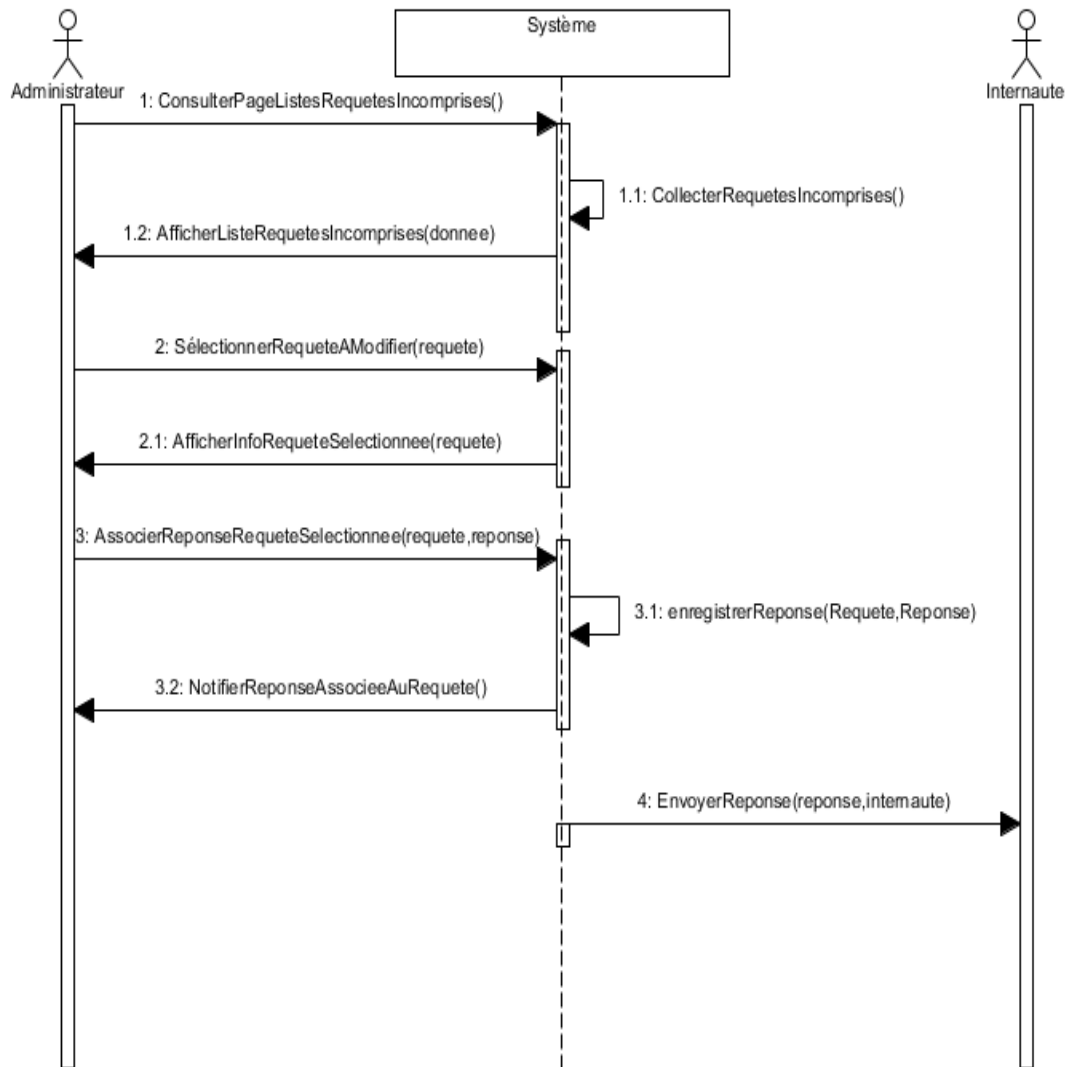


Figure 15 : Diagramme de séquence système du cas d'utilisation "Répondre aux requêtes non comprises par le chatbot"

❖ Diagramme de séquence système du cas d'utilisation « Poser question »

La figure 16 représente le diagramme de séquence système du cas d'utilisation « Poser question ».

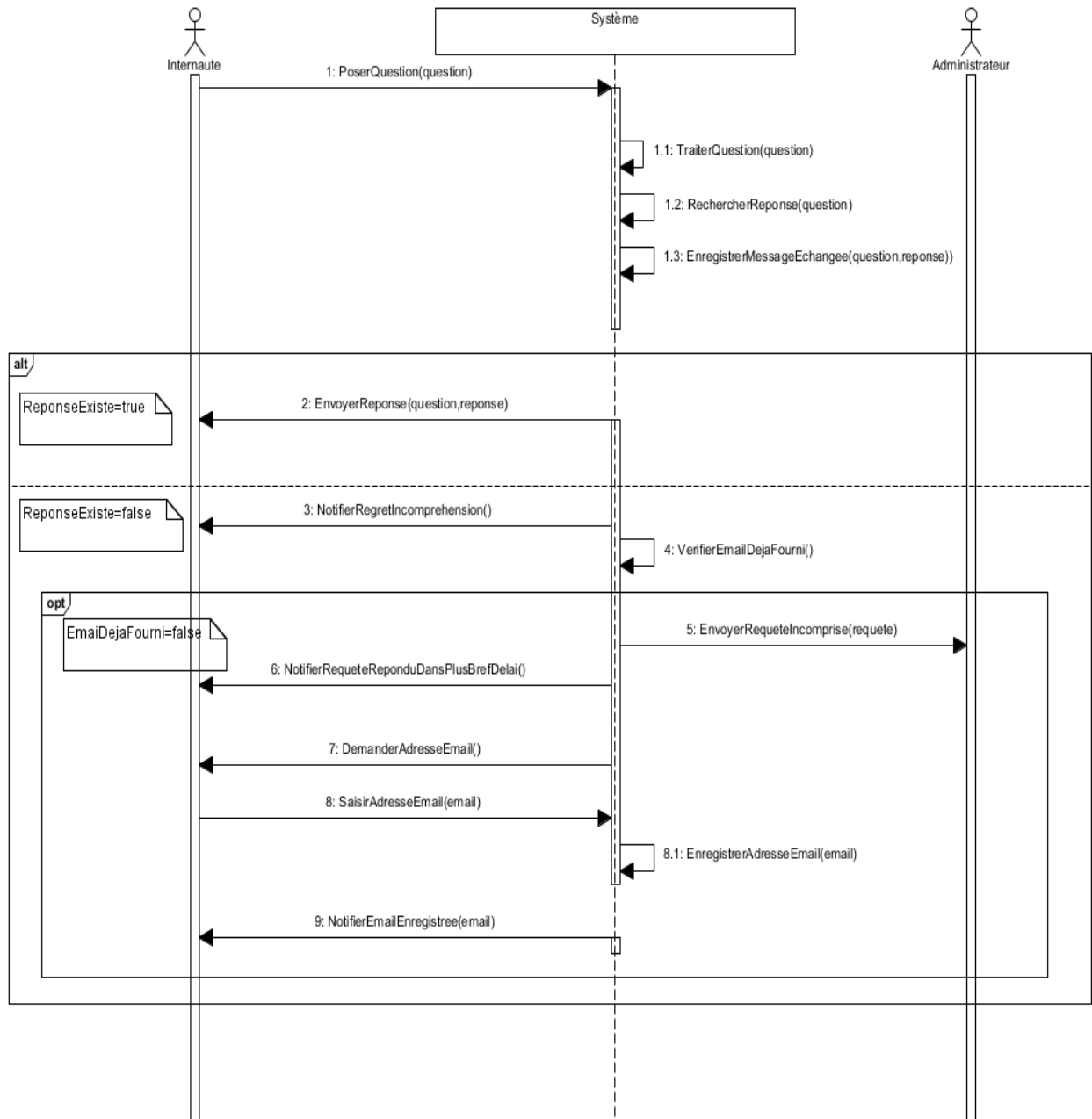


Figure 16 : Diagramme de séquence système du cas d'utilisation "Poser question"

❖ **Diagramme de séquence système du cas d'utilisation « Visualiser la liste des internautes »**

La figure 17 représente le diagramme de séquence système du cas d'utilisation « Visualiser la liste des internautes ».

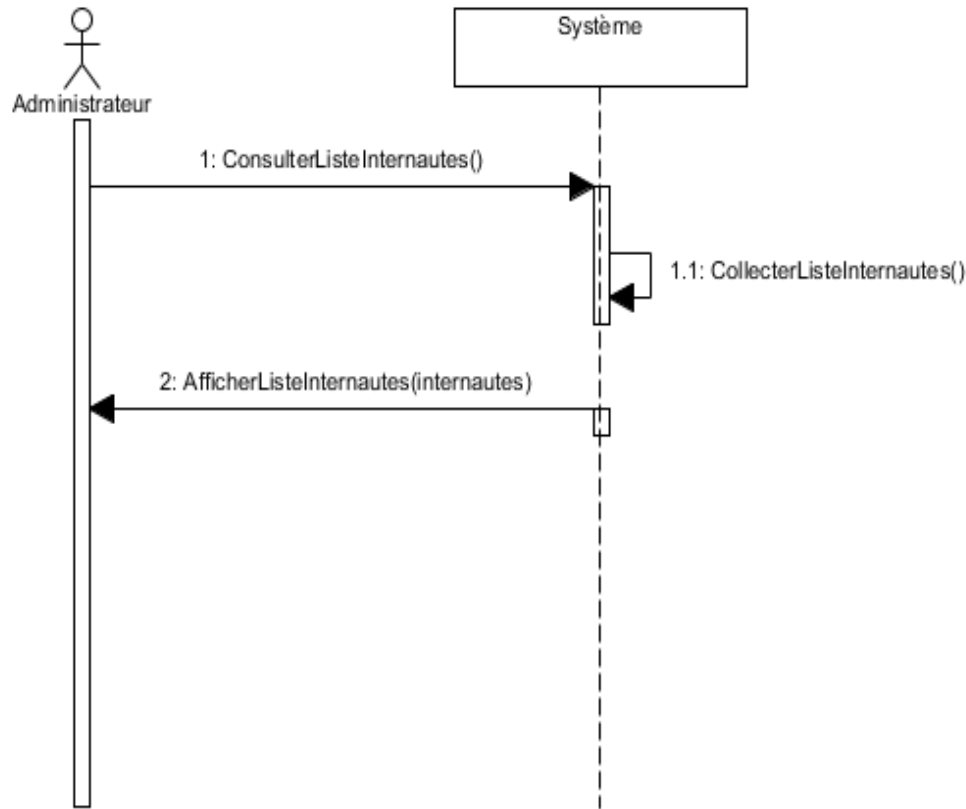


Figure 17 : Diagramme de séquence système du cas d'utilisation "Visualiser la liste des internautes"

5.4. Spécification des besoins techniques

Les besoins techniques décrivent toutes les contraintes techniques, ergonomiques et esthétiques auxquelles est soumis le système pour sa réalisation et pour son bon fonctionnement. Et ce qui concerne notre application, nous avons dégagé les besoins suivants :

- **La disponibilité** : Le chatbot doit toujours être disponible 24h/24 et 7j/7 pour pouvoir répondre aux requêtes des internautes.
- **Temps de réponse** : Fournir des réponses le plus rapidement que possibles
- **Précision** : La réponse fournis par le chatbot doit être la plus précise et compréhensive que possible.
- **Conversation** : La conversation avec le chatbot doit être une conversation le plus fluide que possible (semblable à une conversation avec un humain).
- **La convivialité de l'interface graphique** : l'application doit fournir une interface conviviale et simple pour tout type d'utilisateur car elle présente le premier contact de l'utilisateur avec l'application et par le biais de celle-ci on découvrira ses fonctionnalités.

- **Une solution ouverte et évoluée** : l'application peut être améliorée par l'ajout d'autres modules pour garantir la souplesse, l'évolutivité et l'ouverture de la solution.
- **La possibilité de retourner au menu principal de l'application** à partir de n'importe quelle fenêtre de celle-ci.
- **L'application doit se connecter en permanence à internet** afin de se communiquer aux serveurs.

5.5. Modélisation du domaine

La modélisation du domaine a pour objet de faire une visualisation du monde réel plus ou moins abstraite pour parvenir à la modélisation complète des classes.

La phase d'analyse du domaine permet d'élaborer la première version du diagramme de classe appelée modèle du domaine. Ce modèle doit définir les classes qui modélisent les entités ou concepts présents dans le domaine de l'application. Il s'agit donc de produire un modèle des objets du monde réel dans un domaine donné. Ces entités ou concepts sont identifiés directement à partir de la connaissance du domaine ou par des entretiens avec des experts du domaine.

Pour établir le diagramme, on suit les étapes suivantes :

- ❖ Identifier les entités ou concepts du domaine ;
- ❖ Identifier et ajouter les associations et les attributs ;
- ❖ Organiser et simplifier le modèle en éliminant les classes redondantes et en utilisant l'héritage.

La figure 18 décrit la modélisation du domaine concerné.

Chapitre 6. CONCEPTION DETAILLEE

Nous allons maintenant établir la conception détaillée du système dont nous avons mis en place.

6.1. Architecture du système

L'architecture précédente s'appuie sur MVC (Modèle-Vue-Contrôleur) [3] qui est une architecture destinée à répondre aux besoins des applications interactives en séparant les problématiques liées aux différents composants en les regroupant par couches. Comme son nom l'indique, MVC regroupe les fonctions nécessaires en trois catégories : Modèle, Vue et Contrôleur.

La figure 19 présente le schéma d'une architecture MVC.

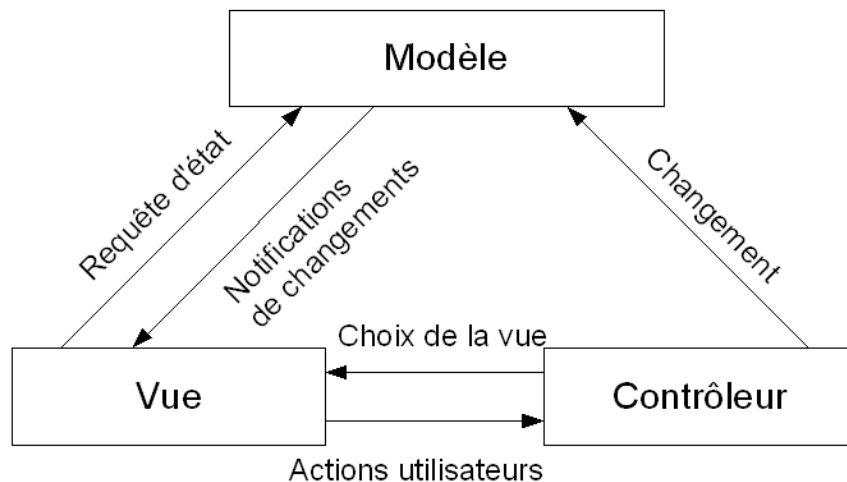


Figure 19 : Schéma de l'architecture MVC

- **Le Modèle (modèle de données)** : cette partie gère les données de l'application. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc les requêtes SQL ou NoSQL.

- **La Vue (présentation, interface utilisateur)** : cette partie se concentre sur l'affichage. Elle ne fait presque aucun traitement et se contente juste de récupérer des variables pour savoir ce qui doit être affiché. On y trouve essentiellement du code HTML.
- **Le Contrôleur** (logique de contrôle, gestion des événements, synchronisation) : cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le Modèle et la Vue : le Contrôleur va demander au Modèle les données pour les analyser, prendre des décisions et renvoyer le texte à afficher à la Vue.

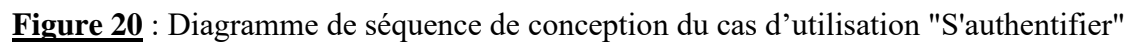
L'architecture MVC a été utilisée ici à cause de la clarté de l'architecture qu'elle impose. En effet, la modification des traitements ne change en rien la Vue. Par exemple on peut passer d'une base de données de type SQL à XML en changeant simplement les traitements d'interaction avec la base, et les vues ne s'en trouvent pas affectées.

La section est consacrée à l'implémentation des différentes fonctionnalités requises à l'authentification.

6.2. Diagramme de séquence de conception pour chaque cas d'utilisation

Le diagramme de séquence de conception permet de présenter les interactions entre les différents modules à l'intérieur du système lors d'une intervention précise. Ce type de diagramme considère le système comme une boîte blanche.

La figure 20 représente le diagramme de séquence de conception du cas d'utilisation « S'authentifier ».



```
sequenceDiagram
    actor Admin as Administrateur
    participant HP as HomePage
    participant CHP as ControleurHomePage
    participant CST as ControleurStatistique
    participant BD as Base de donnée
    participant PS as PageStatistique

    Admin->>HP: 1: ConsulterPageStatistique()
    activate HP
    HP->>CHP: 1.1: VoirPageStatistique()
    deactivate HP
    activate CHP
    CHP->>CST: 1.1.1: VisualiserPageStatistique()
    deactivate CHP
    activate CST
    CST->>BD: 1.1.1.1: ActiverPageStatistique()
    activate BD
    BD->>CST: 1.1.1.2: CollecterInfoStatistique()
    deactivate BD
    CST->>PS: 1.1.1.3: EnvoieAffichageDonneeStatistique()
    activate PS
    PS->>CST: 1.1.1.3.1: AfficherDonneeStatistique()
    deactivate PS
    CST->>Admin: 
    deactivate CST
    deactivate HP
```

{ 63 }

La figure 22 représente le diagramme de séquence de conception du cas d'utilisation « Créer intention ».

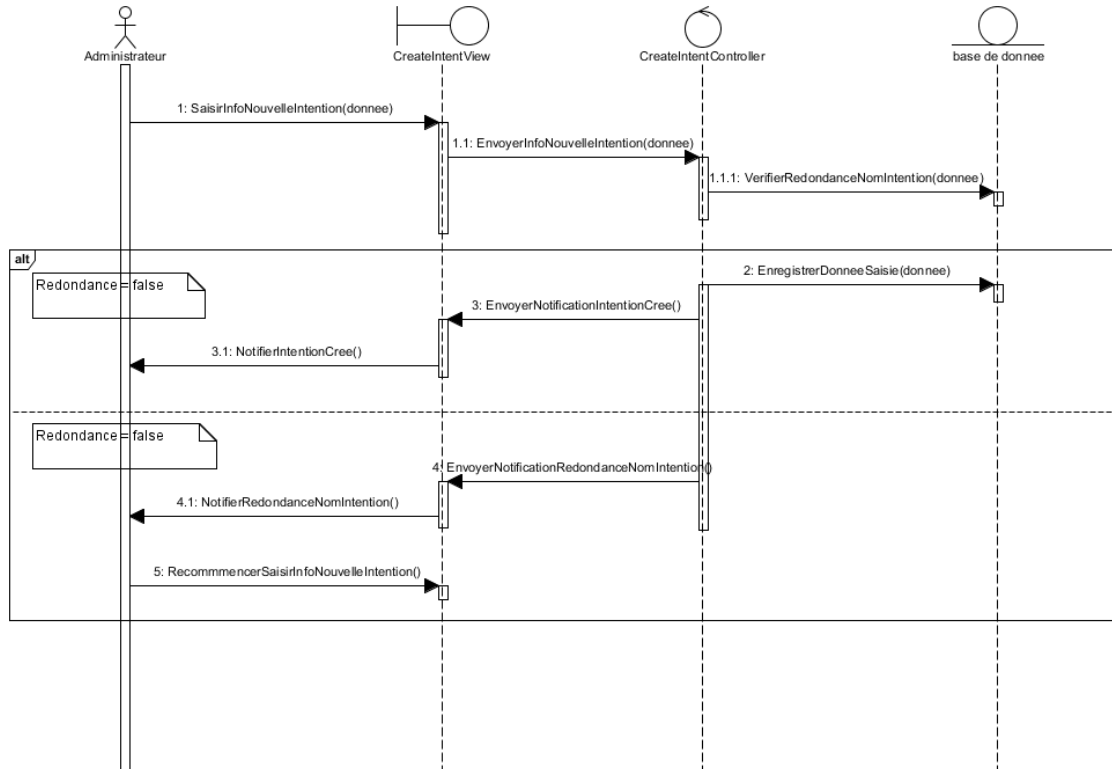


Figure 22 : Diagramme de séquence de conception du cas d'utilisation du cas d'utilisation "Créer intention"

La figure 23 représente le diagramme de séquence de conception du cas d'utilisation « Mettre à jour intention ».

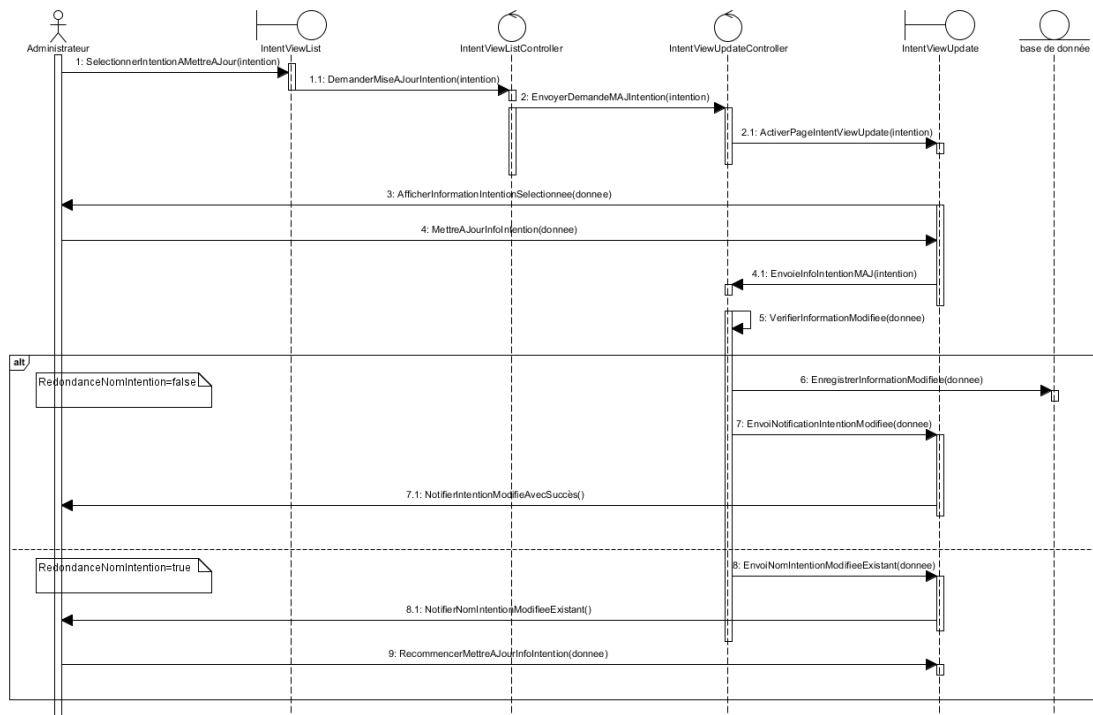


Figure 23 : Diagramme de séquence de conception du cas d'utilisation "Mettre à jour intention"

La figure 24 représente le diagramme de séquence de conception du cas d'utilisation « Discuter avec l'internaute ».

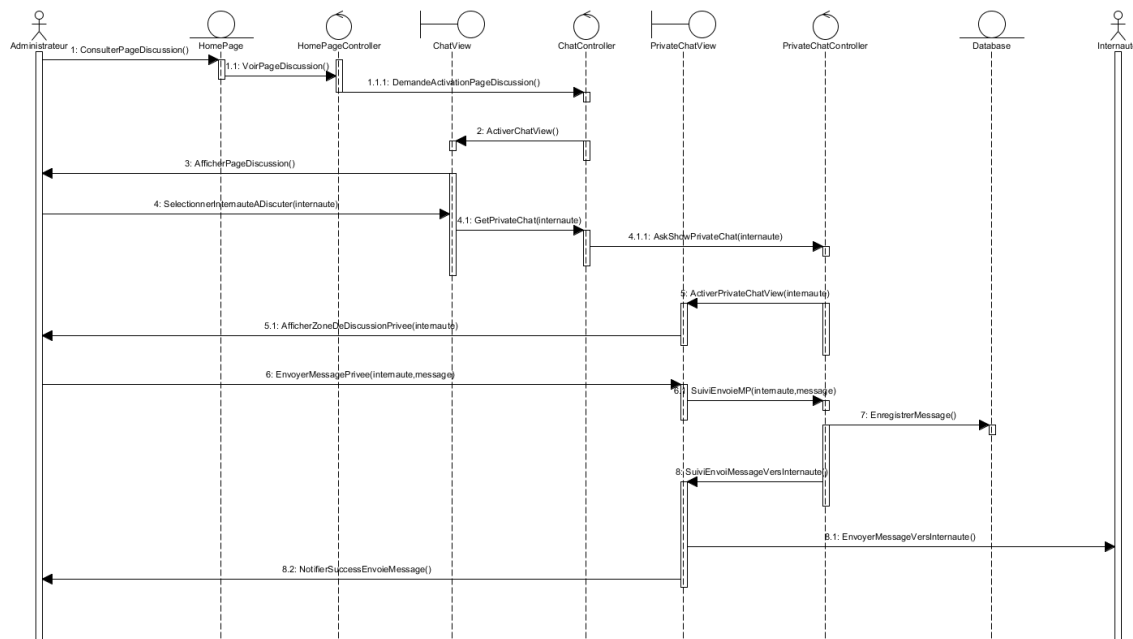


Figure 24 : Diagramme de séquence de conception du cas d'utilisation "Discuter avec l'internaute"

La figure 25 représente le diagramme de séquence de conception du cas d'utilisation « Répondre aux requêtes non comprises par le chatbot ».

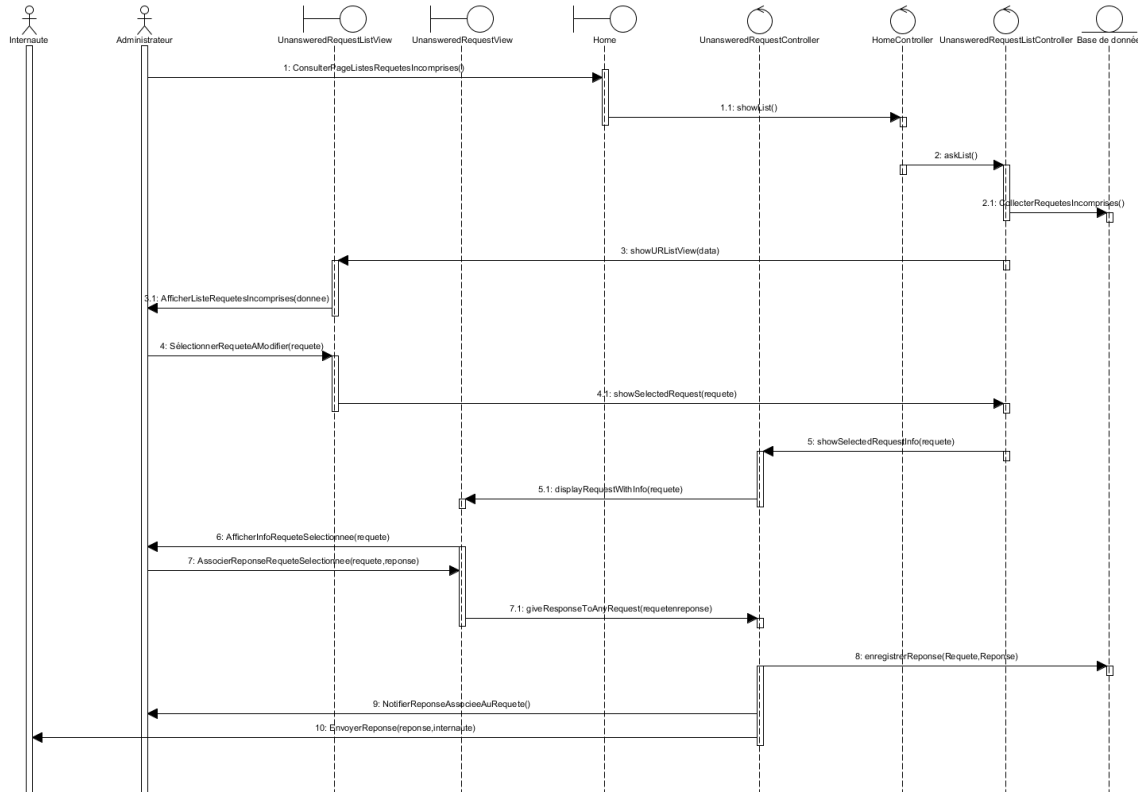


Figure 25 : Diagramme de séquence de conception du cas d'utilisation "Répondre aux requêtes non comprises par le chatbot"

La figure 26 représente le diagramme de séquence de conception du cas d'utilisation « Poser question ».

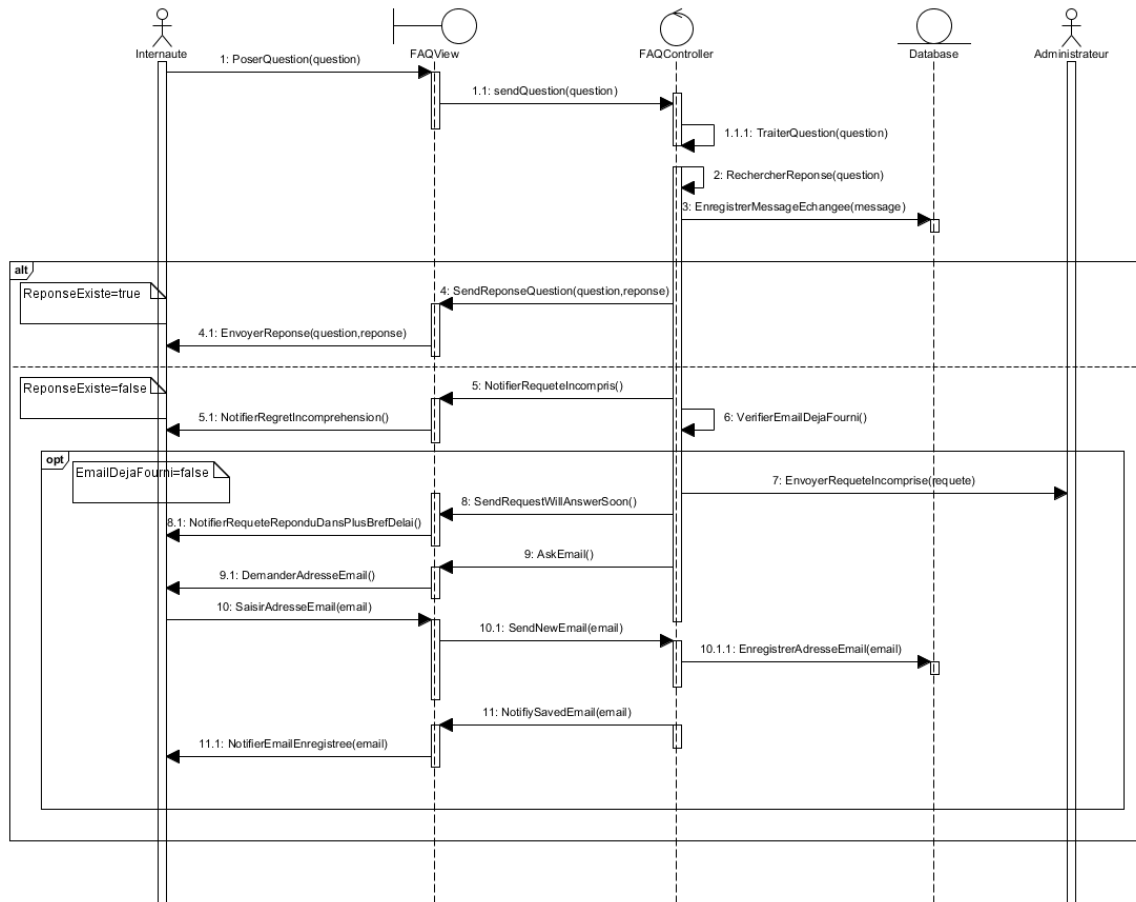


Figure 26 : Diagramme de séquence de conception du cas d'utilisation "Poser question"

La figure 27 représente le diagramme de séquence de conception du cas d'utilisation « Visualiser la liste des internautes ».

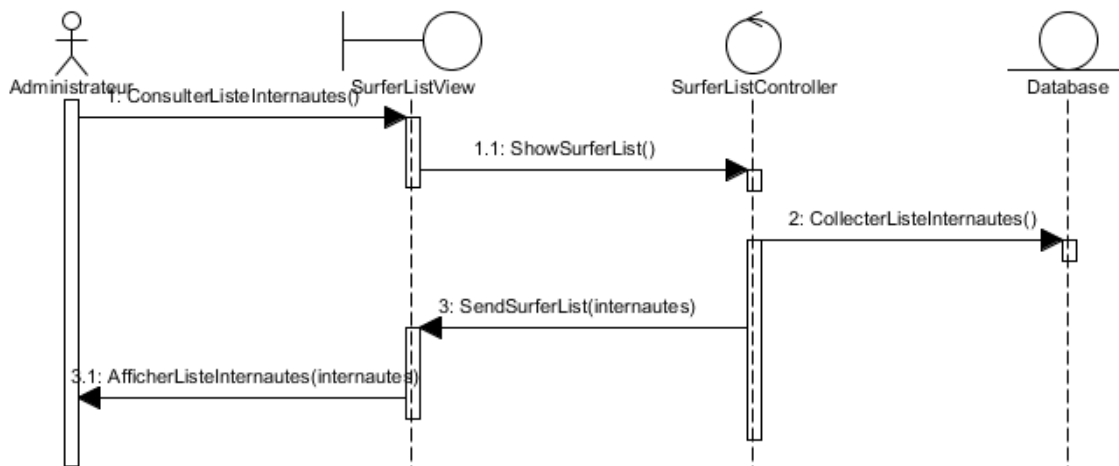


Figure 27: Diagramme de séquence de conception du cas d'utilisation "Visualiser la liste des internautes"

6.3. Diagramme de classe de conception pour chaque cas d'utilisation

A chaque cas d'utilisation, nous allons identifier les classes participantes.

➤ Cas d'utilisation « S'authentifier »

La figure 28 illustre les classes participantes pour le cas d'utilisation « s'authentifier ».

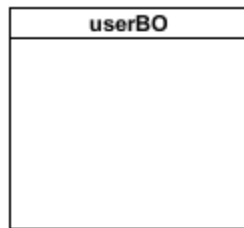


Figure 28 : Diagramme de classe de conception du cas d'utilisation "S'authentifier"

➤ Cas d'utilisation « Afficher statistique »

La figure 29 illustre les classes participantes pour le cas d'utilisation « Afficher statistique ».

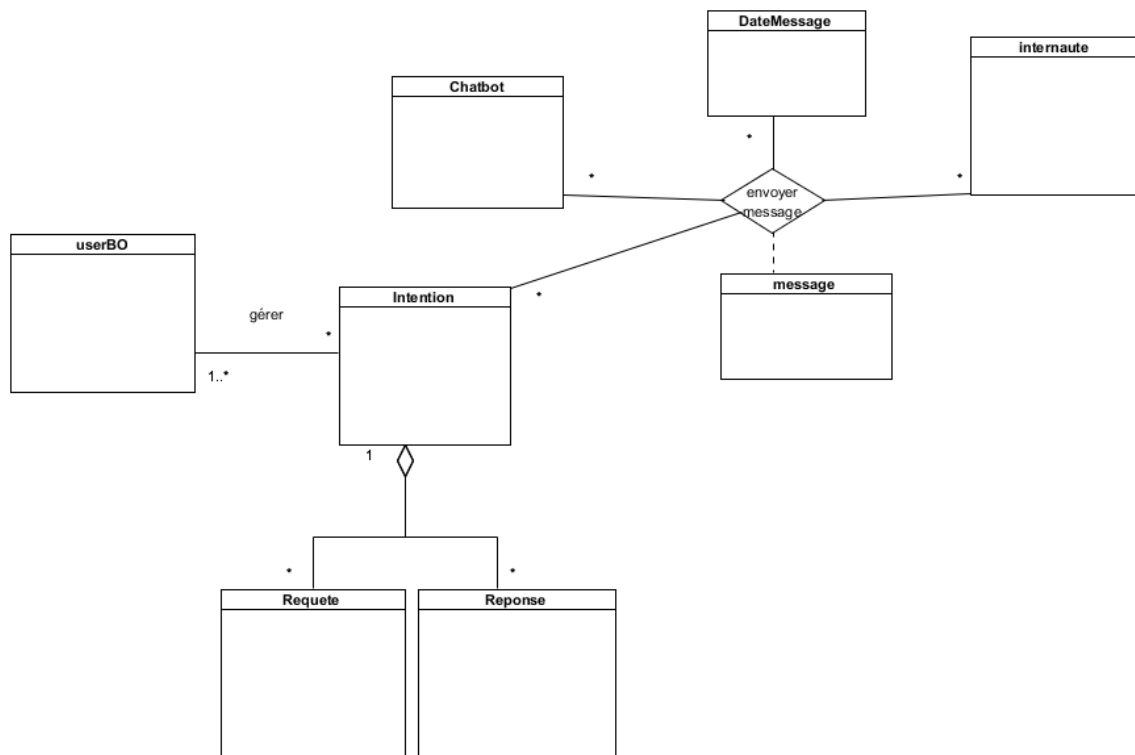


Figure 29 : Diagramme de classe de conception du cas d'utilisation "Afficher statistique"

➤ **Cas d'utilisation « Créer intention »**

La figure 30 illustre les classes participantes pour le cas d'utilisation « Créer intention ».

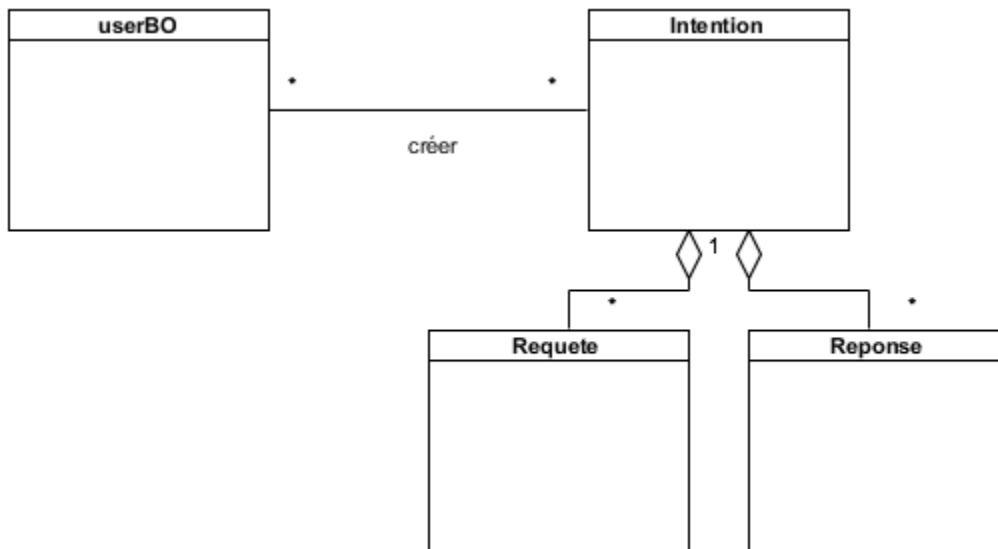


Figure 30: Diagramme de classe de conception du cas d'utilisation "Créer intention"

➤ **Cas d'utilisation « Mettre à jour intention »**

La figure 31 illustre les classes participantes pour le cas d'utilisation « Mettre à jour intention »

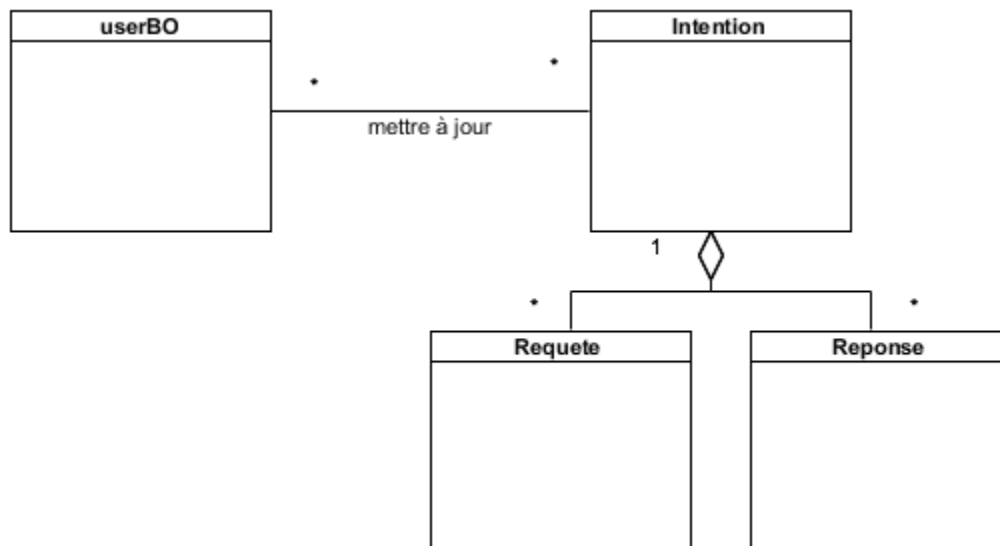


Figure 31 : Diagramme de classe de conception du cas d'utilisation « Mettre à jour intention »

➤ **Cas d'utilisation « Discuter avec l'internaute »**

La figure 32 illustre les classes participantes pour le cas d'utilisation « Discuter avec l'internaute ».

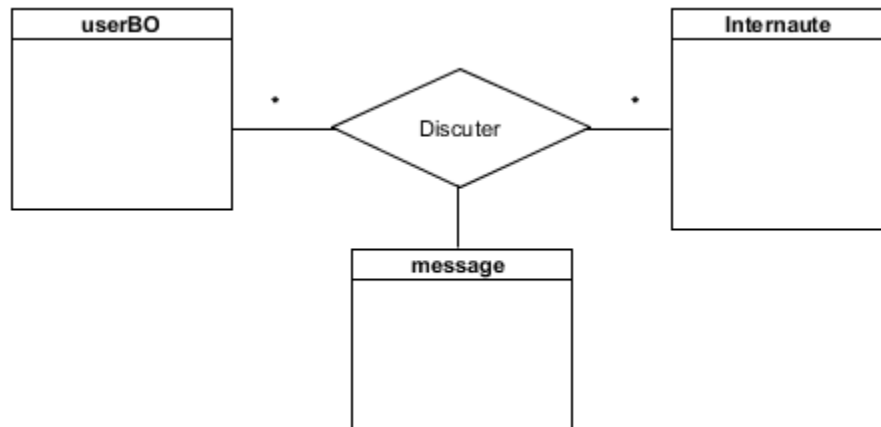


Figure 32 : Diagramme de classe de conception du cas d'utilisation "Discuter avec l'internaute"

➤ **Cas d'utilisation « Répondre aux requêtes non comprises par le chatbot »**

La figure 33 illustre les classes participantes pour le cas d'utilisation « Répondre aux requêtes non comprises par le chatbot ».

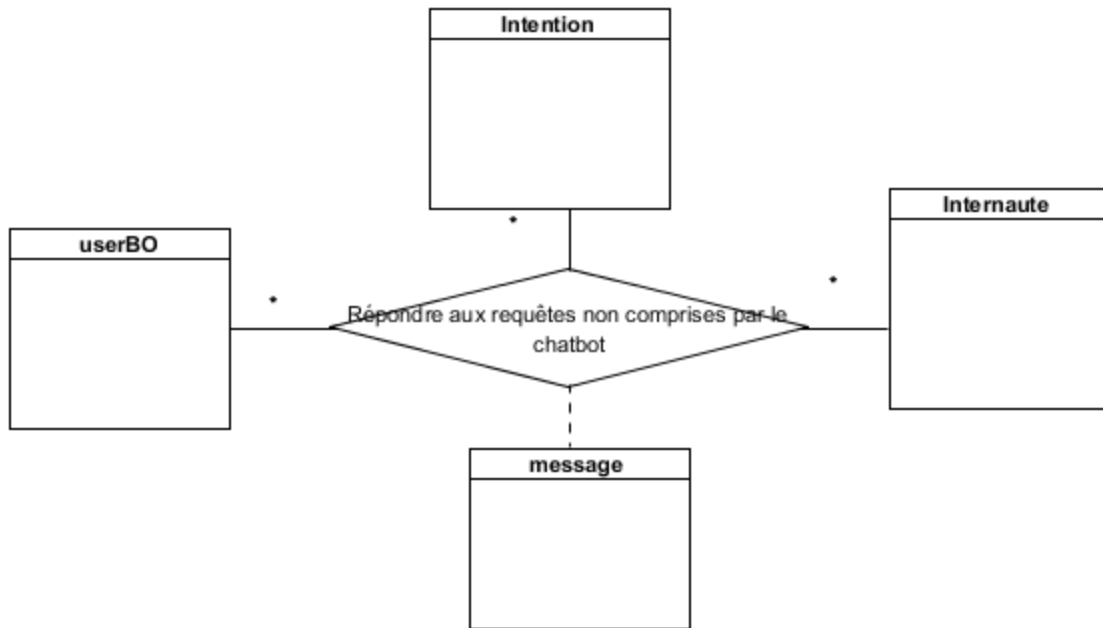


Figure 33 : Diagramme de classe de conception du cas d'utilisation "Répondre aux requêtes non comprises par le chatbot"

➤ **Cas d'utilisation « Poser question »**

La figure 34 illustre les classes participantes pour le cas d'utilisation « Poser question ».

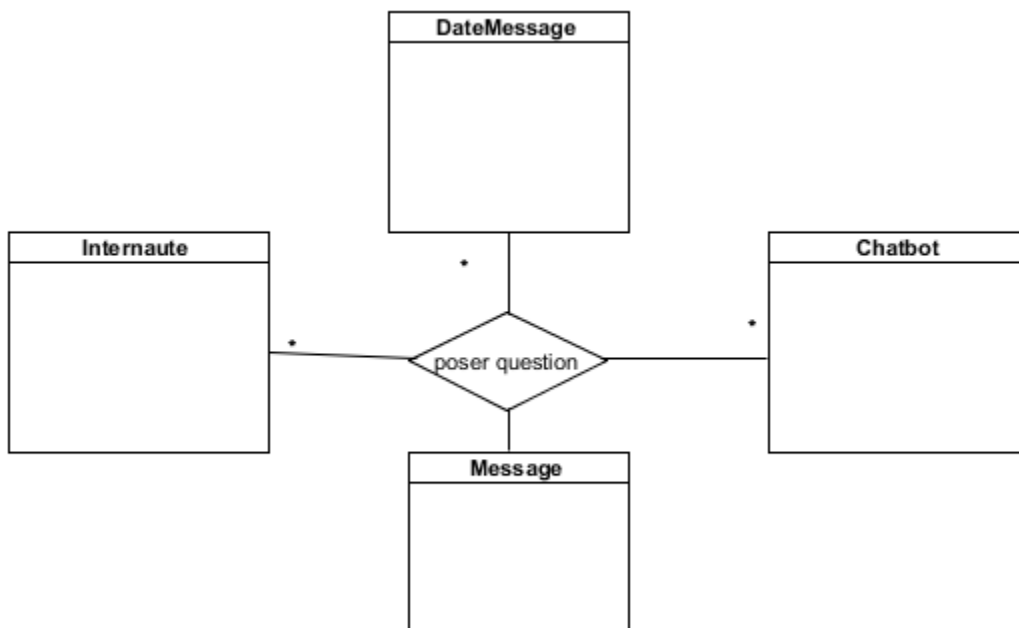


Figure 34 : Diagramme de classe de conception du cas d'utilisation "Poser question"

➤ **Cas d'utilisation « Visualiser la liste des internautes »**

La figure 35 illustre les classes participantes pour le cas d'utilisation « Visualiser la liste des internautes ».

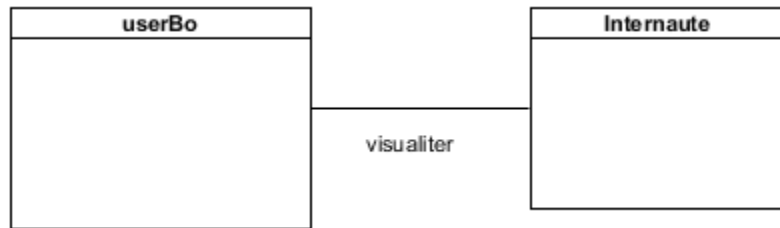


Figure 35 : Diagramme de classe de conception du cas d'utilisation "Visualiser la liste des internautes"

6.4. Diagramme de classe de conception global

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.

Classe

Une classe décrit un groupe d'objets ayant les mêmes propriétés (attributs), un même comportement (opérations), et une sémantique commune (domaine de définition).

Un objet est une instance d'une classe. La classe représente l'abstraction de ses objets. Au niveau de l'implémentation, c'est-à-dire au cours de l'exécution d'un programme, l'identificateur d'un objet correspond une adresse mémoire.

Une classe se représente à l'aide d'un rectangle comportant plusieurs compartiments.

Les trois compartiments de base sont :

- La désignation de la classe,
- La description des attributs,
- La description des opérations.

Deux autres compartiments peuvent être aussi indiqués :

- La description des responsabilités de la classe,
- La description des exceptions traitées par la classe.

Il est possible de manipuler les classes en limitant le niveau de description à un nombre réduit de compartiments selon les objectifs poursuivis par le modélisateur.

La figure 36 montre le formalisme général des compartiments d'une classe et des premiers exemples.

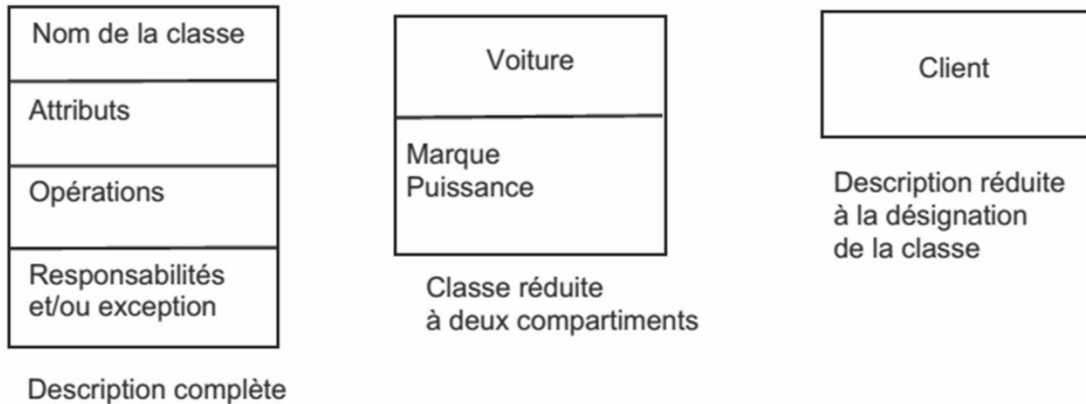


Figure 36 : Formalisme général d'une classe et exemples

Visibilité des attributs et opérations

Chaque attribut ou opération d'une classe peut être de type public, protégé, privé ou paquetage. Les symboles + (public), # (protégé), - (privé) et ~ (paquetage) sont indiqués devant chaque attribut ou opération pour signifier le type de visibilité autorisé pour les autres classes.

Les droits associés à chaque niveau de confidentialité sont :

- Public (+) Attribut ou opération visible par tous.
- Protégé (#) Attribut ou opération visible seulement à l'intérieur de la classe et pour toutes les sous-classes de la classe.
- Privé (-) Attribut ou opération seulement visible à l'intérieur de la classe.
- Paquetage (~) Attribut ou opération ou classe seulement visible à l'intérieur du paquetage où se trouve la classe.

Association

Un lien est une connexion physique ou conceptuelle entre instances de classes donc entre objets. Une association décrit un groupe de liens ayant une même structure et une même

sémantique. Un lien est une instance d'une association. Chaque association peut être identifiée par son nom.

Une association entre classes représente les liens qui existent entre les instances de ces classes.

Le rôle tenu par une classe vis-à-vis d'une association peut être précisé sur l'association.

Multiplicité

La multiplicité indique un domaine de valeurs pour préciser le nombre d'instance d'une classe vis-à-vis d'une autre classe pour une association donnée. La multiplicité peut aussi être utilisée pour d'autres usages comme par exemple un attribut multivalué.

Le domaine de valeurs est décrit selon plusieurs formes :

- Intervalle fermé– Exemple : 2, 3 ..15.
- Valeurs exactes – Exemple : 3, 5, 8.
- Valeur indéterminée notée * – Exemple : 1..*.
 - Dans le cas où l'on utilise seulement *, cela traduit une multiplicité 0..*.
 - Dans le cas de multiplicité d'associations, il faut indiquer les valeurs minimale et maximale d'instances d'une classe vis-à-vis d'une instance d'une autre classe. [5]

La figure 37 montre quelques exemples des principales multiplicités définies dans UML.

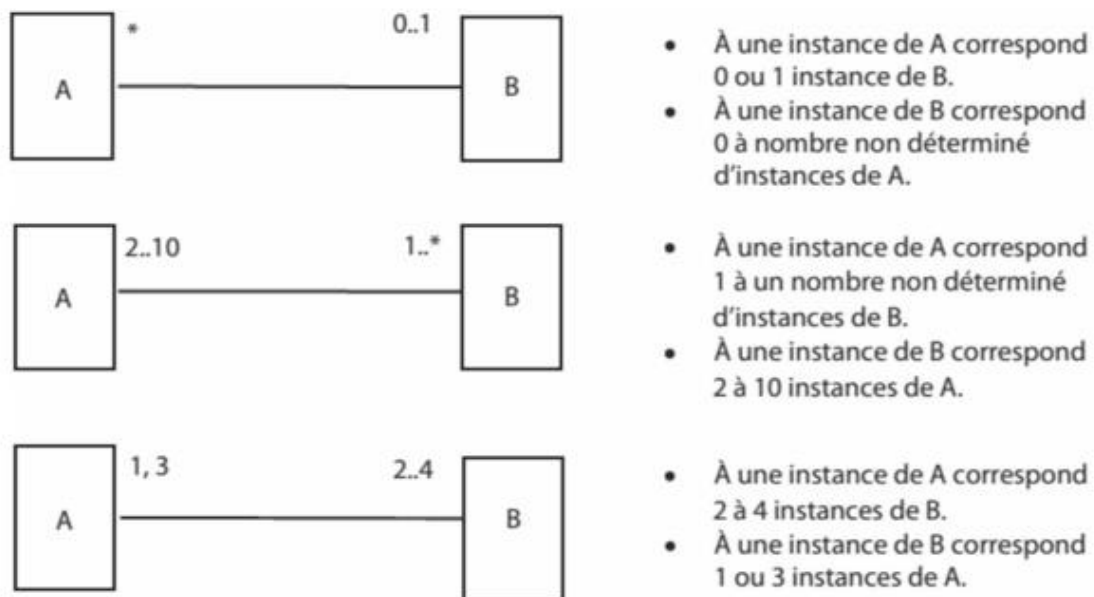


Figure 37 : Exemple de multiplicités

```

classDiagram
    class Chatbot {
        -chatbot_id : string
        -botName : string
        -botTask : string
    }
    class DataMessage {
        -dateEnvoie : date
    }
    class Intention {
        -intention_id : string
        -imgIntention : string
        -nomIntention : string
        -descIntention : string
        -typeIntention : string
        +ajouterIntention()
        +mettreAJourIntention()
    }
    class Message {
        -message : string
        -file : string
    }
    class Requete {
        -requete_id : string
        -contenuRequete : string
        -auteurRequete : string
        +ajouterRequete()
        +supprimerRequete()
    }
    class Reponse {
        -reponse_id : string
        -contenuReponse : string
        -auteurReponse : string
        +ajouterReponse()
        +supprimerReponse()
    }
    class UserBO {
        -userEmail : string
        -userPwd : string
    }
    class Internaute {
        -internaute_id : string
        -nomInternaute : string
        -email : string
        -localisation : string
        -botReply : boolean
        -attribute : string
        +voirListeInternaute()
        +ajouterInternaute()
    }

    Chatbot "1" -- "*" DataMessage
    Chatbot "1" -- "*" Intention
    DataMessage "1" -- "*" Internaute
    Intention "1" -- "*" Message
    Intention "1" -- "*" Reponse
    UserBO "1" -- "1..*" Intention : gérer
    Internaute "1" -- "*" Reponse
    
```

6.5. Diagramme de paquetages

Les éléments d'un paquetage peuvent avoir une visibilité déclarée soit de type public (+) soit privé (-).

[75]

Formalisme et exemple

Il existe trois manières de présenter un paquetage.

- ✓ **Représentation globale** – Le nom du paquetage se trouve à l'intérieur du grand rectangle.
- ✓ **Représentation détaillée** – Les membres du paquetage sont représentés et le nom du paquetage d'ensemble s'inscrit dans le petit rectangle.
- ✓ **Représentation éclatée** – Les membres du paquetage sont reliés par un lien connecté au paquetage par le symbole \oplus .

La figure 39 montre le formalisme général d'un paquetage et les trois manières de présenter un paquetage.

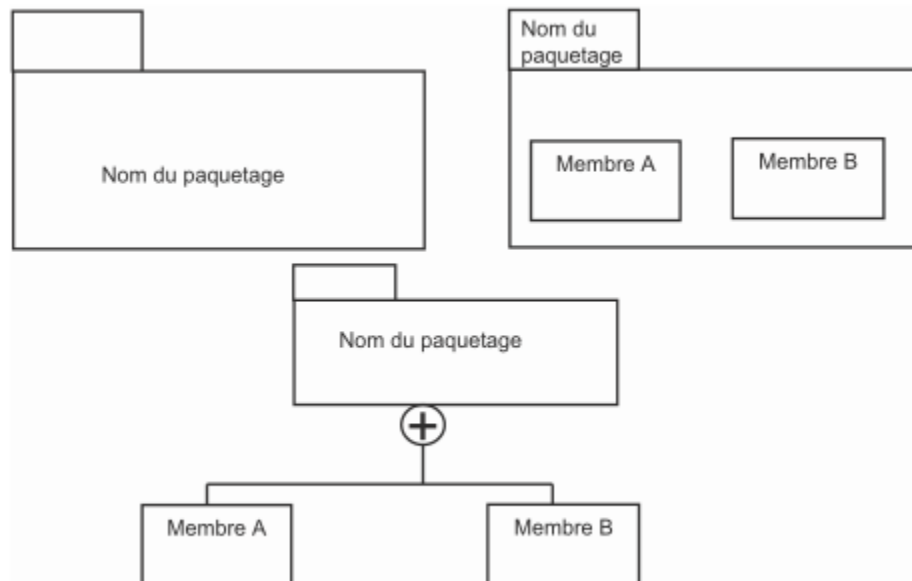


Figure 39: Formalisme de représentation de paquets

Dépendance entre paquetages

La dépendance entre paquetages peut être qualifiée par un niveau de visibilité qui est soit public soit privé. Par défaut le type de visibilité est public.

À chaque type de visibilité est associé un lien de dépendance. Les deux types de dépendances entre paquetages sont :

- « **import** » – Ce type de dépendance permet, pour un paquetage donné, d'importer l'espace de nommage d'un autre paquetage. Ainsi tous les membres du paquetage donné ont accès à tous les noms des membres du paquetage importé sans avoir à utiliser explicitement le nom du paquetage concerné. Ce type de dépendance correspond à un lien ayant une visibilité « public ».
- « **access** » – Ce type de dépendance permet, pour un paquetage donné, d'avoir accès à l'espace de nommage d'un paquetage cible. L'espace de nommage n'est donc pas importé et ne peut être transmis à d'autres paquetages par transitivité. Ce type de dépendance correspond à un lien ayant une visibilité « privé ».

La figure 40 présente un exemple de liens de dépendance entre paquetages.

Dans cet exemple, les éléments de Clients externes sont importés dans Domaine client et ensuite dans Domaine tiers. Cependant, les éléments de Clients internes sont seulement accessibles par le paquetage Domaine client et donc pas à partir du paquetage Domaine tiers.

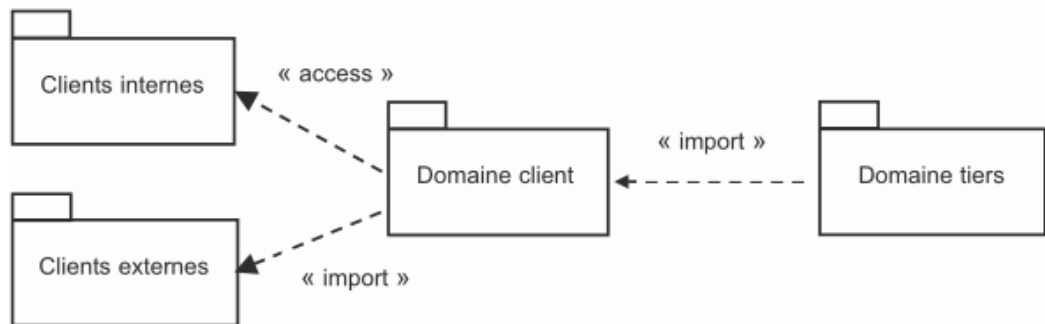


Figure 40 : Exemple de liens de dépendance entre paquetages

Le diagramme de paquets correspondant au domaine d'étude est présenté dans la figure 41.

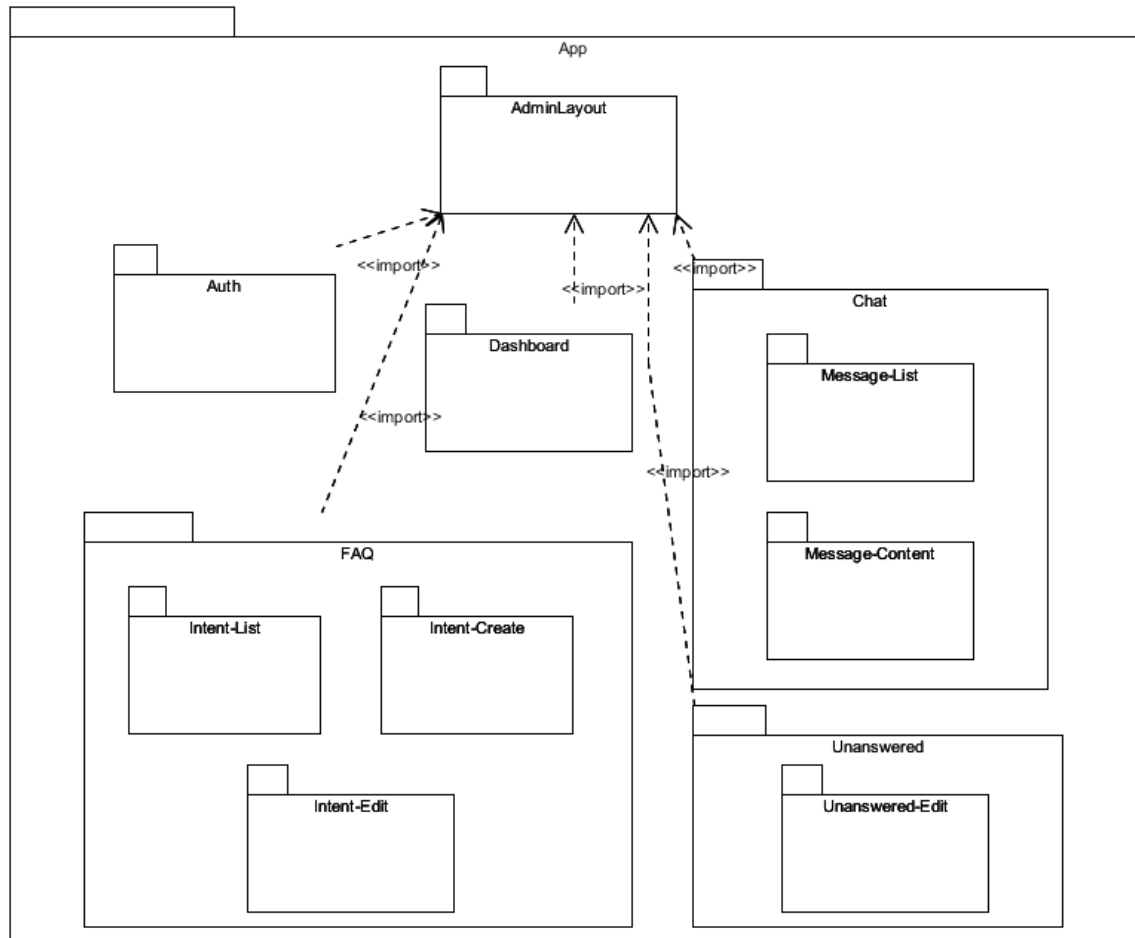


Figure 41 : Diagramme de paquetage du domaine d'étude

6.6. Diagramme de déploiement

Le diagramme de déploiement permet de représenter l'architecture physique supportant l'exploitation du système. Cette architecture comprend des nœuds correspondant aux supports physiques (serveurs, routeurs...) ainsi que la répartition des artefacts logiciels (bibliothèques, exécutables...) sur ces nœuds. C'est un véritable réseau constitué de nœuds et de connexions entre ces nœuds qui modélise cette architecture.

Nœud

Un nœud correspond à une ressource matérielle de traitement sur laquelle des artefacts seront mis en œuvre pour l'exploitation du système. Les nœuds peuvent être interconnectés pour former un réseau d'éléments physiques.

Formalisme et exemple

Un nœud ou une instance de nœud se représente par un cube ou parallélépipède. La figure 42 présente ce qu'est un nœud.



Figure 42 : Représentation de nœuds

Le diagramme de déploiement correspondant au domaine d'étude est présenté sur la figure 43.

Le protocole utilisé dans ce diagramme est le protocole HTTP. Les données manipulées sont de type JSON.

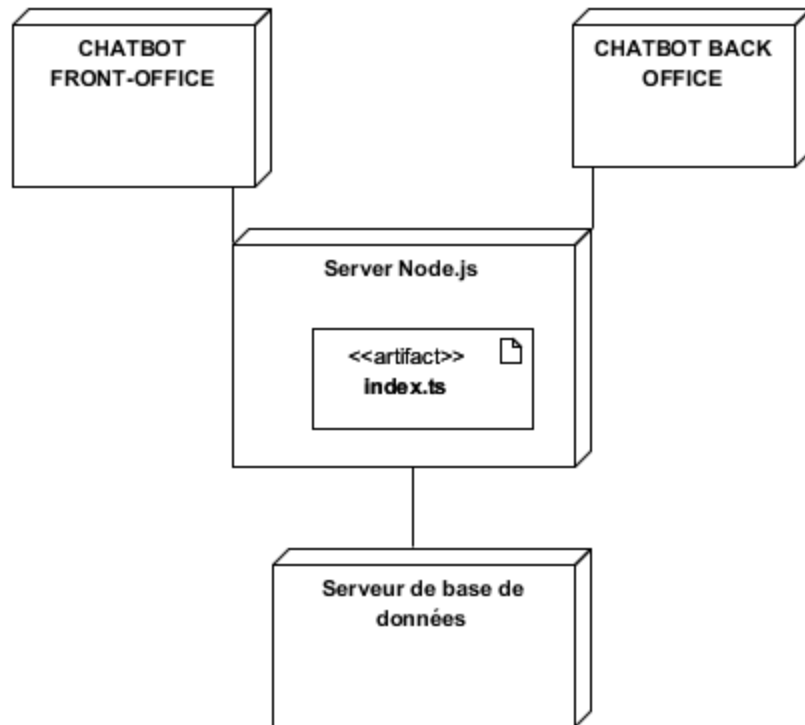


Figure 43 : Diagramme de déploiement du domaine d'étude

PARTIE III : REALISATION

Chapitre 7. Mise en place de l'environnement de développement

Dans ce chapitre nous allons aborder l'installation des outils nécessaires au projet ainsi que l'architecture de l'application.

7.1. Installation et configuration des outils

Comme vue auparavant, nous avons déjà défini l'ensemble des outils dont nous allons utiliser pour l'élaboration du projet, cependant nous allons maintenant les installer.

7.1.1 Installation du framework Angular

Pour installer le framework Angular, il faut avant tout installé plusieurs dépendances dont ce framework a besoin pour fonctionner.

7.1.2 Installation de Node.js

Pour installer NodeJS, il suffit de le télécharger sur le site officiel <https://www.node.org/en/download> .

La figure 44 présente l'installation de Node.js.

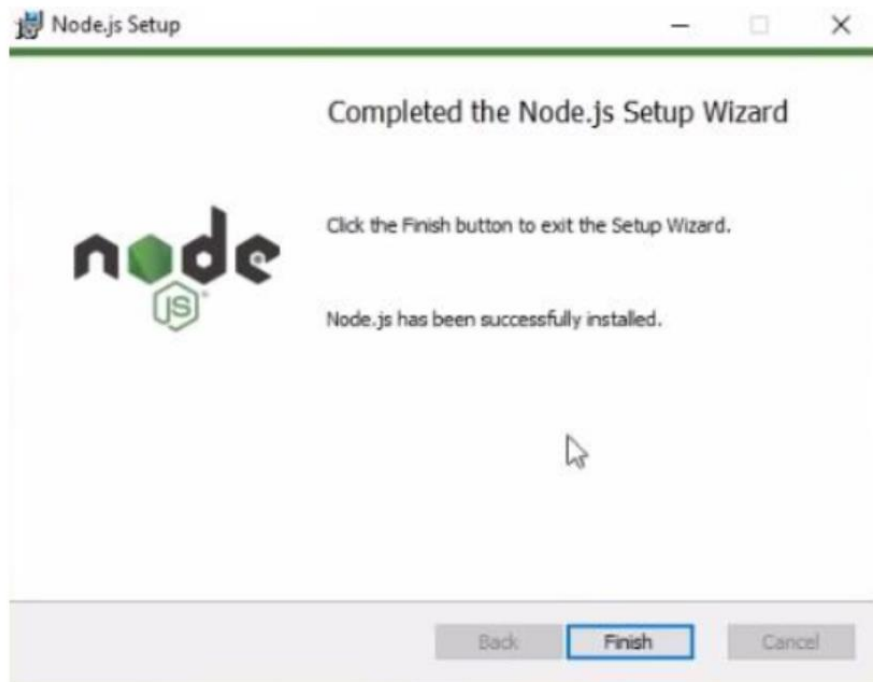


Figure 44 : Installation de Node.js

7.1.3 Installation d'Angular

Finalement pour installer Angular, il suffit d'ouvrir un terminal puis on lance la commande **npm install -g @angular/cli@latest**.

La figure 45 présente l'installation d'Angular

The image shows a terminal window with a black background and white text. The title bar says "C:\> npm". The first line of output is "Your environment has been set up for using Node.js 11.8.0 (x64) and npm." The second line shows the command being executed: "C:\Users\user>npm install -g @angular/cli@latest". The third line shows the progress of the installation: "[.....] / fetchMetadata: sill resolveWithNewModule readable-stream@2.3.6 che".

Figure 45 : Installation d'Angular

7.1.4 Installation de MongoDB

Pour installer MongoDB, il suffit de le télécharger sur le site officiel <https://www.mongodb.com/download-center?lang=fr-fr>.

La figure 46 présente l'installation de MongoDB.

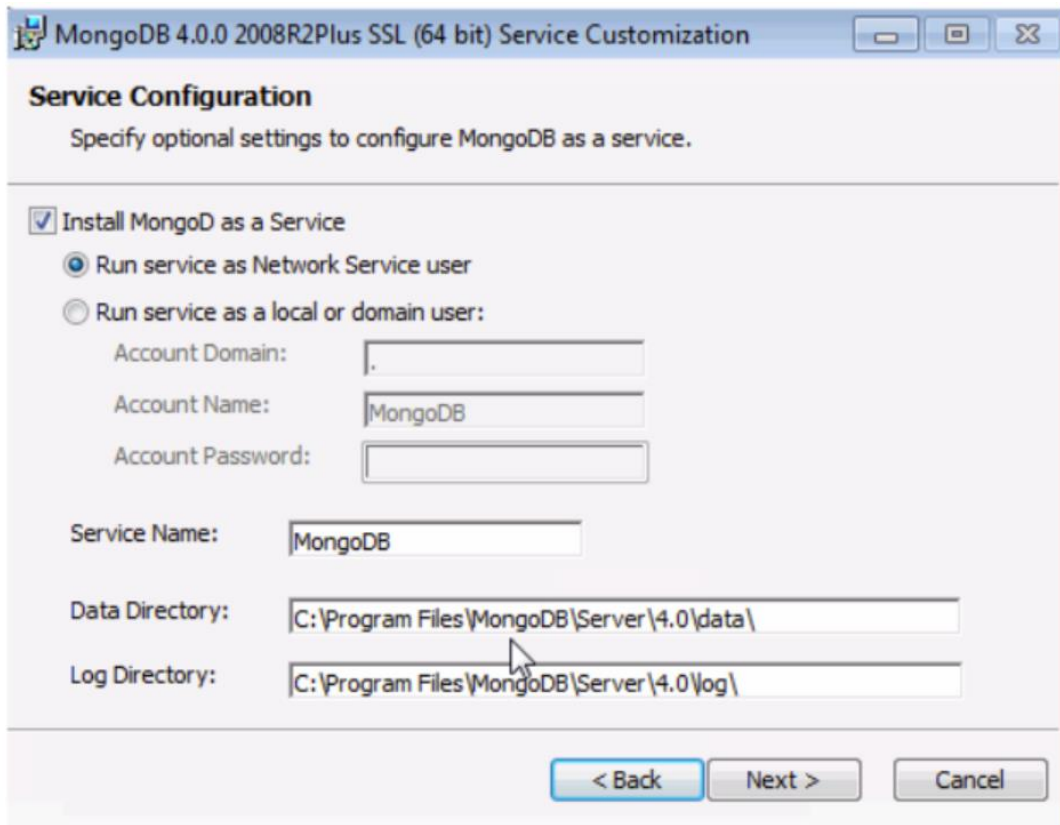


Figure 46 : Installation de MongoDB

7.1.5 Installation de Visual Studio Code

Pour installer Visual Studio Code, il suffit de le télécharger sur le site officiel <https://code.visualstudio.com/download>.

La figure 47 présente l'installation de Visual Studio Code.

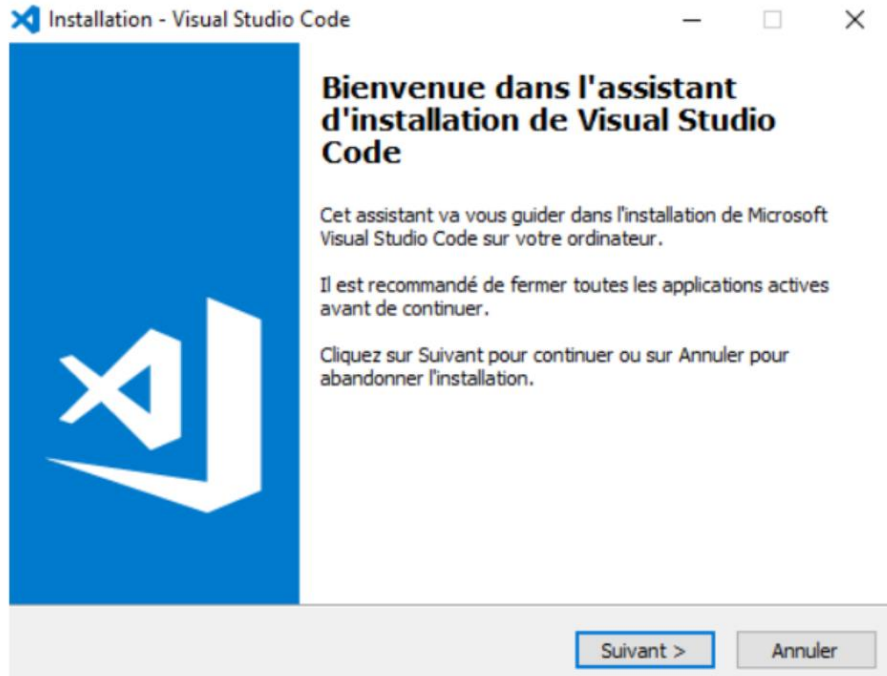


Figure 47 : Installation de Visual Studio Code

7.1.6 Installation de Visual Paradigm

Le Visual Paradigm utilisé pour la modélisation de ce projet est : « Visual Paradigm for UML Enterprise Edition ». Pour l'installer, il suffit de télécharger le programme d'installation sur le lien www.visual-paradigm.com/download/ , puis on l'exécute et normalement la fenêtre d'installation qui s'ouvre doit être comme ce qui est représentée sur la figure 48.

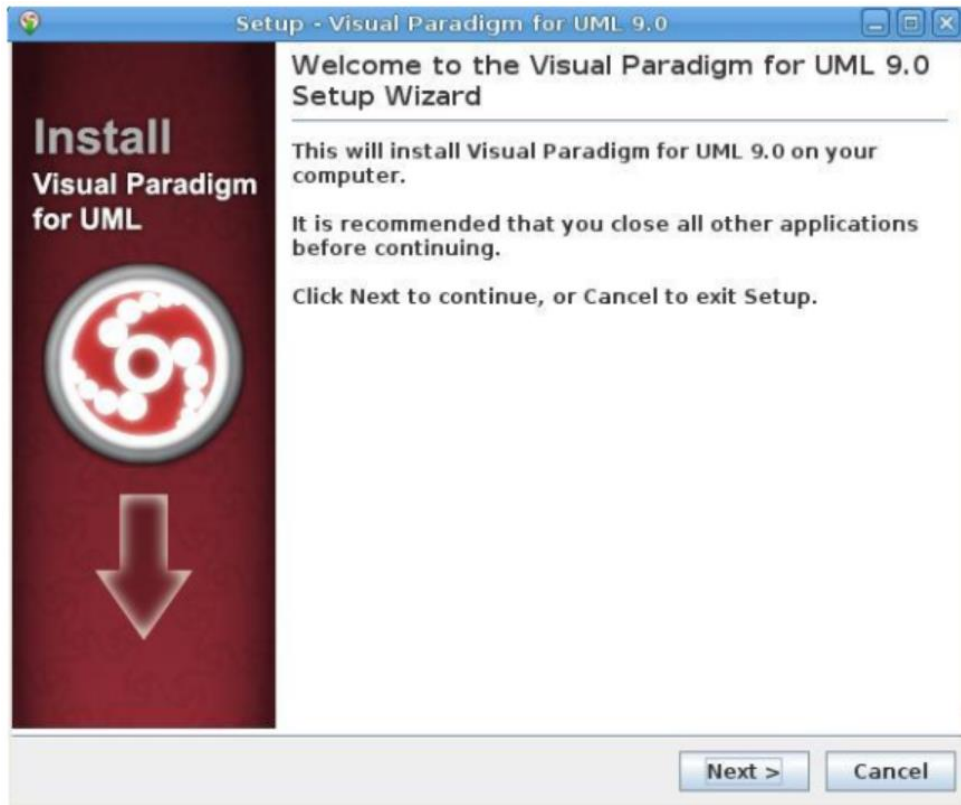


Figure 48 : Installation de Visual Paradigm for UM

7.1.7 Installation de ngx-socket, NLP.js et NgRx

Pour installer *NLP.js*, on lance la commande :

npm install node-nlp --save

Pour installer *ngx-socket* , on lance la commande

npm install ngx-socket-io --save

Pour pouvoir installer *NgRx* ainsi que ses dépendances, on lance la commande

npm install @ngrx/{store,effects,entity,store-devtools,schematics} --save

7.2. Architecture de l'application

Dans ce présent projet nous avons adopté l'architecture à trois tiers qui s'appuie sur trois niveaux dont :

- La vue (ou la présentation)
- La logique applicative (ou les traitements)
- La base de données

La figure 49 illustre l'architecture de notre application.

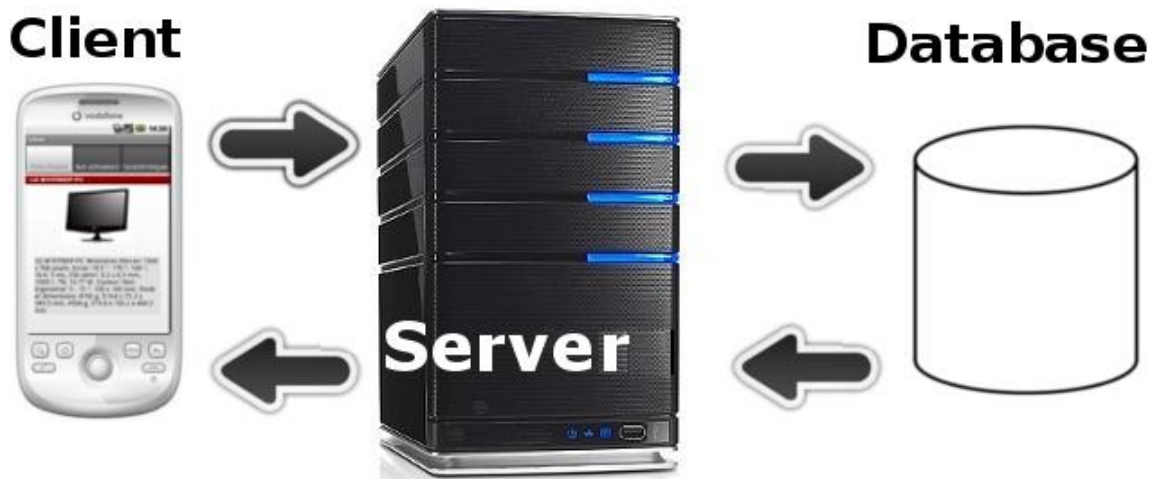


Figure 49 : Architecture de l'application

Chapitre 8. DÉVELOPPEMENT DE L'APPLICATION

Dans ce dernier chapitre nous allons présenter quelques morceaux de codes nécessaires pour la création de la base de données et pour le codage de l'application et enfin nous allons présenter quelques captures d'écran de l'application développée.

8.1. Création de la base de données

Puisque MongoDB est un SGBD NoSQL, l'utilisation de requête SQL pour la création de la base de données ne sera pas possible dans notre cas.

Cependant, pour créer notre base de données nous avons utilisée des scripts via le langage NodeJS (plus précisément Express.js). Ainsi, il suffira d'exécuter ce morceau de code pour créer la base de données (s'il n'existe pas encore)

```
import * as mongoose from "mongoose";

let Schema = mongoose.Schema ;

let UserBoSchema = new Schema({

  userEmail : { type: String , required : true} ,

  userPwd : { type: String , required : true}

}, { _id : true , timestamps : true });

let BotSchema = new Schema({

  botName : { type: String , required : true} ,

  botTask : { type : String }

},{ _id : true , timestamps : true })

export default mongoose.model("bots", BotSchema);

let IntentionSchema = new Schema({

  imgIntention: { type: String },

  nomIntention: { type: String, required: true },
```

```

    descIntention: { type: String },

    // native, standard or specifique

    typeIntention : { type : String }

    },{ _id: true , timestamps : true });

export default mongoose.model("intentions", IntentionSchema);

let InternauteSchema = new Schema({

    nomInternaute : { type: String , required : true } ,

    email : { type : String },

    localisation : { type : Object },

    botReply : { type : Boolean }

    },{ _id : true , timestamps : true })

export default mongoose.model("internautes", InternauteSchema);

let MessageSchema = new Schema({

    sender_id: { type: mongoose.Schema.Types.ObjectId , required: true },

    recipient_id : { type: mongoose.Schema.Types.ObjectId, required : true },

    intention_id : { type: mongoose.Schema.Types.ObjectId, ref: 'intentions' },

    message: { type: String, required: false } ,

    file : { type: String },

    },{ _id: true , timestamps : true } );

export default mongoose.model("messages", MessageSchema);

let ReponseSchema = new Schema({

    contenuReponse : { type: String, required : true },

    auteurReponse : { type: String, required : true },

    intention_id : { type: mongoose.Schema.Types.ObjectId, ref: 'intentions' ,

    required : true }

```

```

    }, { _id: true, timestamps: true });

    export default mongoose.model('reponses', ReponseSchema);

    let RequeteSchema = new Schema({

    contenuRequete : { type: String, required: true },

    auteurRequete : { type: String, required: true },

    intention_id : { type: mongoose.Schema.Types.ObjectId, ref: 'intentions' }

    }, { _id: true, timestamps: true });

    export default mongoose.model('requetes', RequeteSchema);

```

8.2. Codage de l'application

Nous allons présenter quelques morceaux de codes illustrant la collecte de toutes les réponses et requêtes associé à tous les intentions et puis sauvegarder ces informations dans la base de connaissances du chatbot afin de l'entraîner pour bien détecter l'intention de l'internaute lorsque ce dernier envoie sa demande. Certes, cette information sauvegardée permettra aussi au chatbot de bien répondre aux requêtes de l'internaute.

```

import { NlpManager } from 'node-nlp';

import Intention from '../models/intention.model';
import Requete from '../models/requete.model';
import ReponseController from '../controllers/ReponseController';
import RequeteController from '../controllers/RequeteController';

export class QnA {

    constructor(private manager: NlpManager, private lang: string) {}

    async getDataForBotTraining() {

        const promise = new Promise(async (resolve, reject) => {
            const data = await this.getAllIntentsWithData()

```

```

        resolve(data)
    })
    const promise2 = await promise;
    const promise3 = await this.trainBot(promise2);
    return;
}
async getAllIntentsWithData() {
    let tmp: any;
    let tmp2: any;
    try {
        let intents = await Intention.find().lean();
        const promise = new Promise((resolve, reject) => {
            tmp = RequeteController.getAllAssociatedRequestsForEveryIntent(intents);
            resolve(tmp);
        })
        const promise2 = await promise;
        const data = ReponseController.getAllAssociatedResponsesForEveryIntent(promise2);
        return data;
    } catch (err) {
        console.log(err);
    }
}

async trainBot(dataTraining: any) {
    // Reset Nlp Manager
    this.manager = new NlpManager({ languages: ['fr'], forceNER: true,
    useNoneFeature: false });
    this.manager = await this.addDataToBot(this.manager, dataTraining);
    // *** TRAIN BOT *** //
    await this.manager.train();
    this.manager.save();
    console.log('trainBot()');
    // console.log(this.manager)
    return true;
}

addDataToBot(manager: NlpManager, dataTraining: any) {
    let data: any;
    // ***** ADD DATA TO BOT *****//
    for (var i = 0; i < dataTraining.length; i++) {
        // Requete
        for (var j = 0; j < dataTraining[i]['requete'].length; j++) {
            manager.addDocument(this.lang,
            dataTraining[i]['requete'][j]['contenuRequete'], dataTraining[i]['nomIntention']);
        }
        // Reponse
    }
}

```

```

        for (var j = 0; j < dataTraining[i]['reponse'].length; j++) {
            manager.addAnswer(this.lang, dataTraining[i]['nomIntention'],
dataTraining[i]['reponse'][j]['contenuReponse']);
        }
    }
    data = manager;
    return data;
}
async getAnswer(msg: any) {
    // let result: any;
    let data: any;
    await this.getDataForBotTraining();
    console.log('if(promise2)');
    const response = await this.manager.process(this.lang, msg);
    console.log(response)
    data = {
        // None or different
        intent: response['intent'],
        // undefined or other value
        answer: response['answer']
    }
    return data;
}
}
}

```

8.3. Présentation de l'application

Nous allons maintenant visualiser la partie back-office de l'application que nous avons développée.

La figure 50 illustre la page d'authentification de l'application.

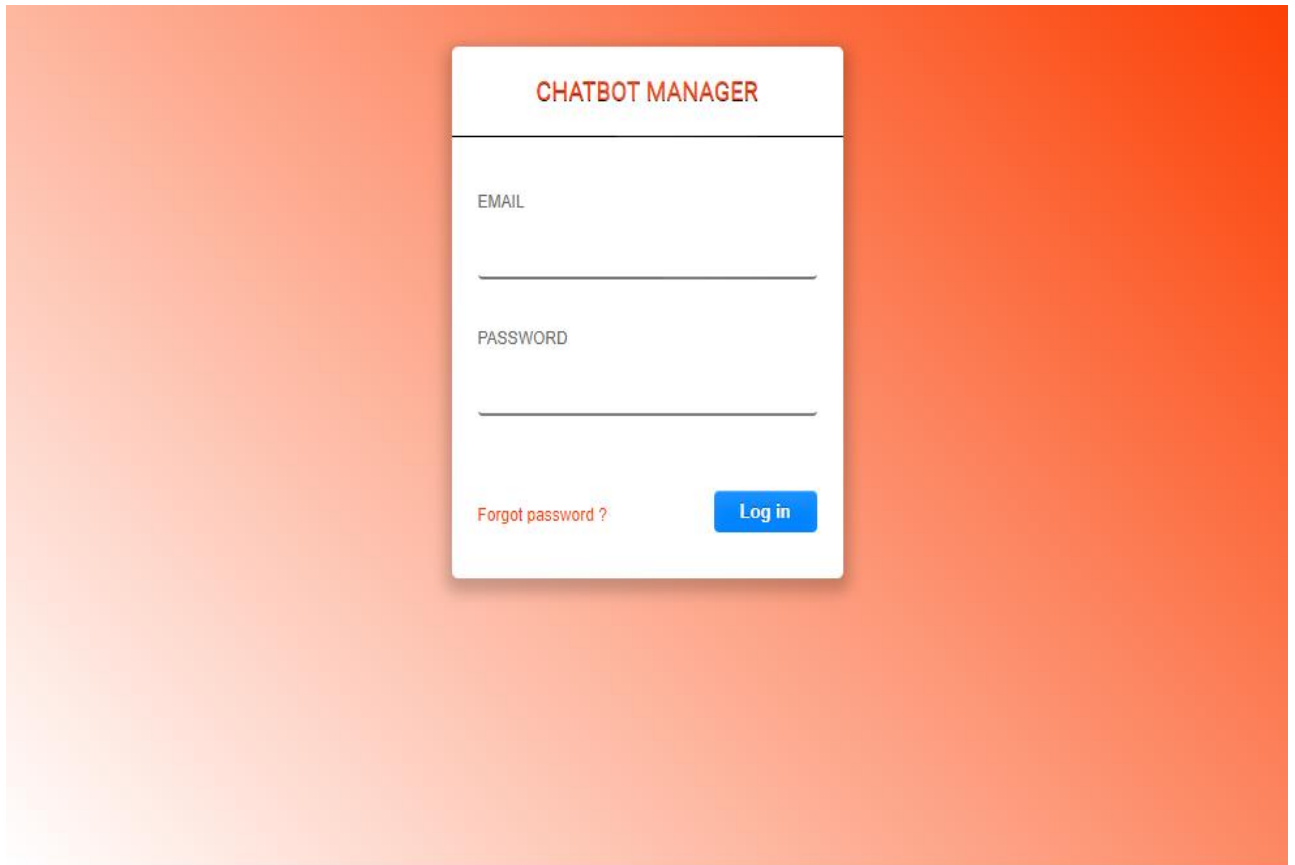


Figure 50 : Page d'authentification

La figure 51 illustre la page tableau de bord (ou page statistique) de l'application.

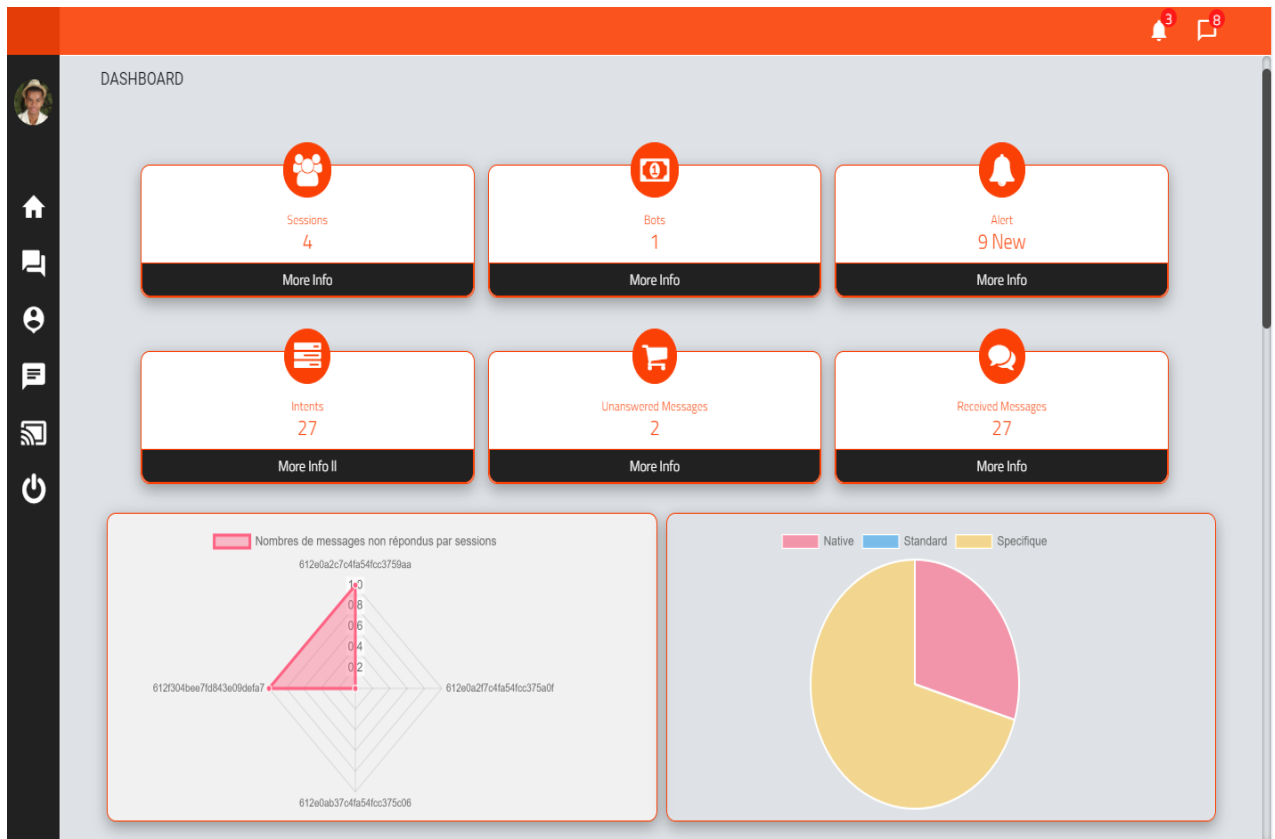


Figure 51 : Page tableau de bord

La figure 52 illustre la liste des intentions.

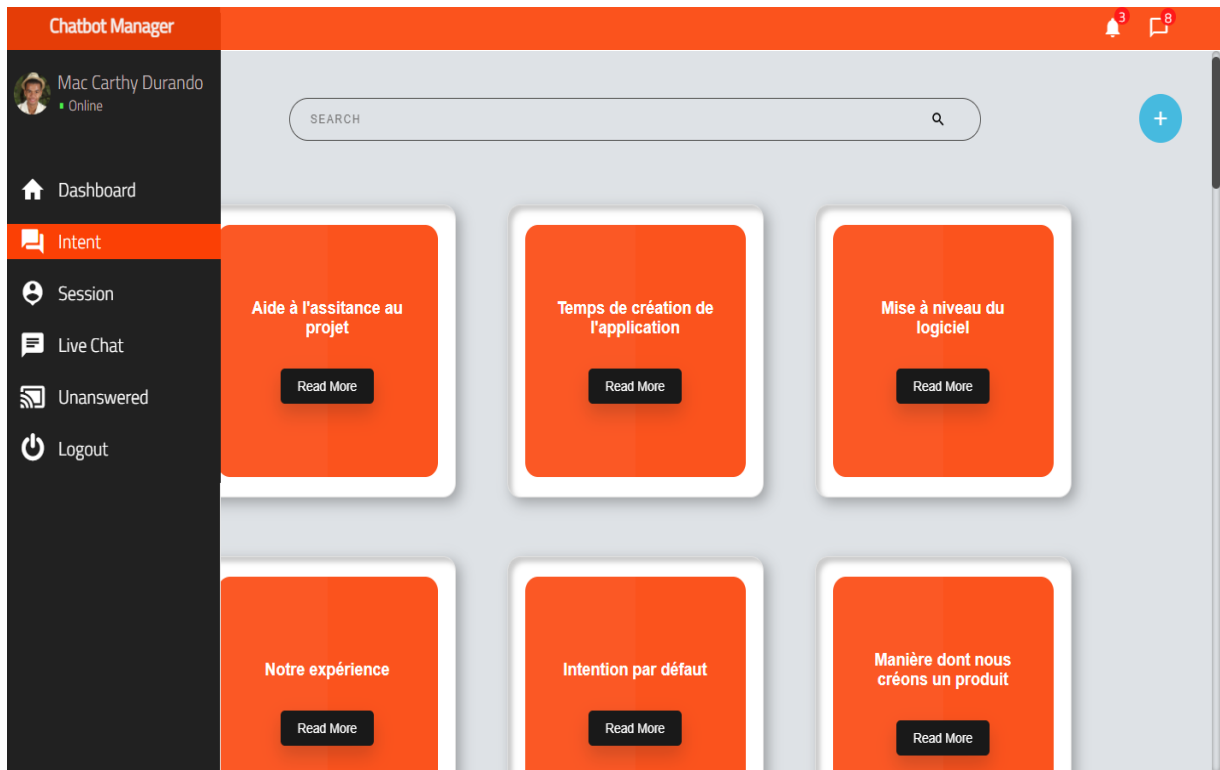


Figure 52 : Liste des intentions

La figure 53 illustre la page de création d'intention.

CREATE AN INTENT

Nom de l'intention

Description de votre intention

Type de l'intention
Native

If user says something similar to
Training phrases

Bot replies with
Bot answer

SAUVEGARDER RETOUR

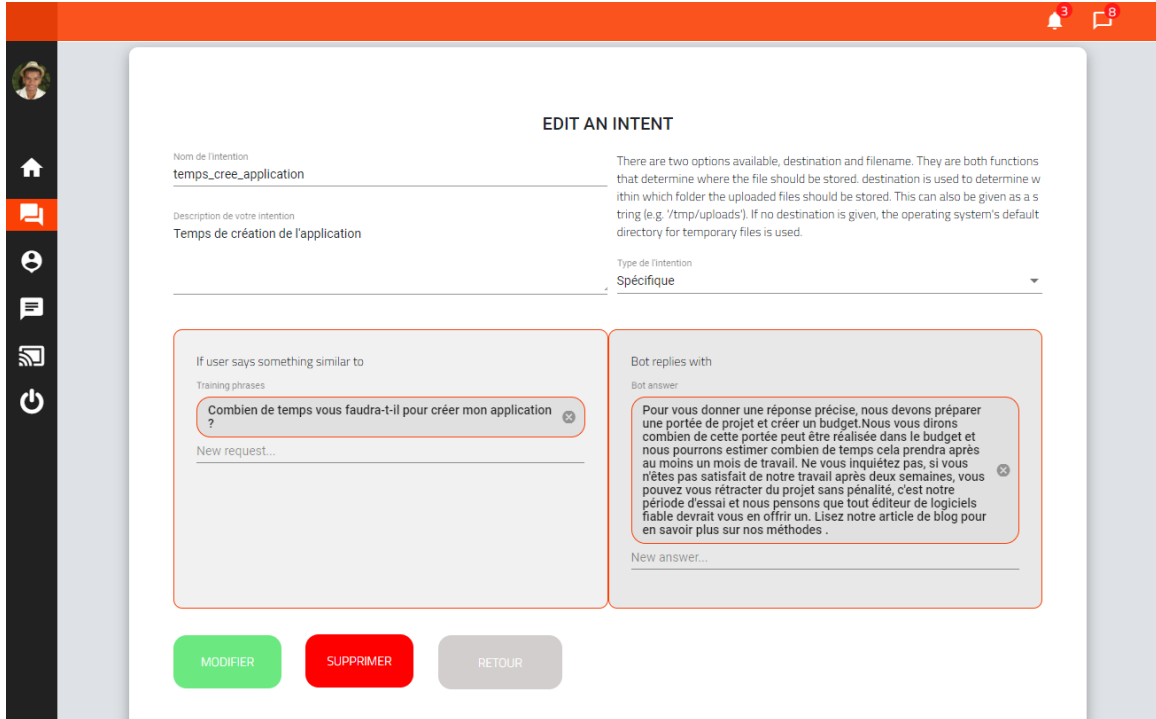
Dans un chatbot, l'intention fait référence à l'objectif que le client a en tête lorsqu'il saisit une question ou un commentaire. Alors que l'entité fait référence au modificateur que le client utilise pour décrire son problème, l'intention est ce qu'il signifie vraiment. L'intention est un facteur critique dans la fonctionnalité du chatbot, car la capacité du chatbot à analyser l'intention est ce qui détermine en fin de compte le succès de l'interaction. Pour qu'un chatbot soit bon dans ce domaine, il doit :

- Être bien programmé et entraîné avec un modèle utile impliquant beaucoup de données d'entraînement
- Tirer parti du machine learning pour progresser et vous améliorer constamment

Copyright © 2021 All rights reserved | By Young.dev

Figure 53 : Page de création d'intention

La figure 54 affiche la page d'édition d'une intention.



EDIT AN INTENT

Nom de l'intention
temps_cree_application

Description de votre intention
Temps de création de l'application

Type de l'intention
Spécifique

If user says something similar to

Training phrases
Combien de temps vous faudra-t-il pour créer mon application ?

New request...

Bot replies with

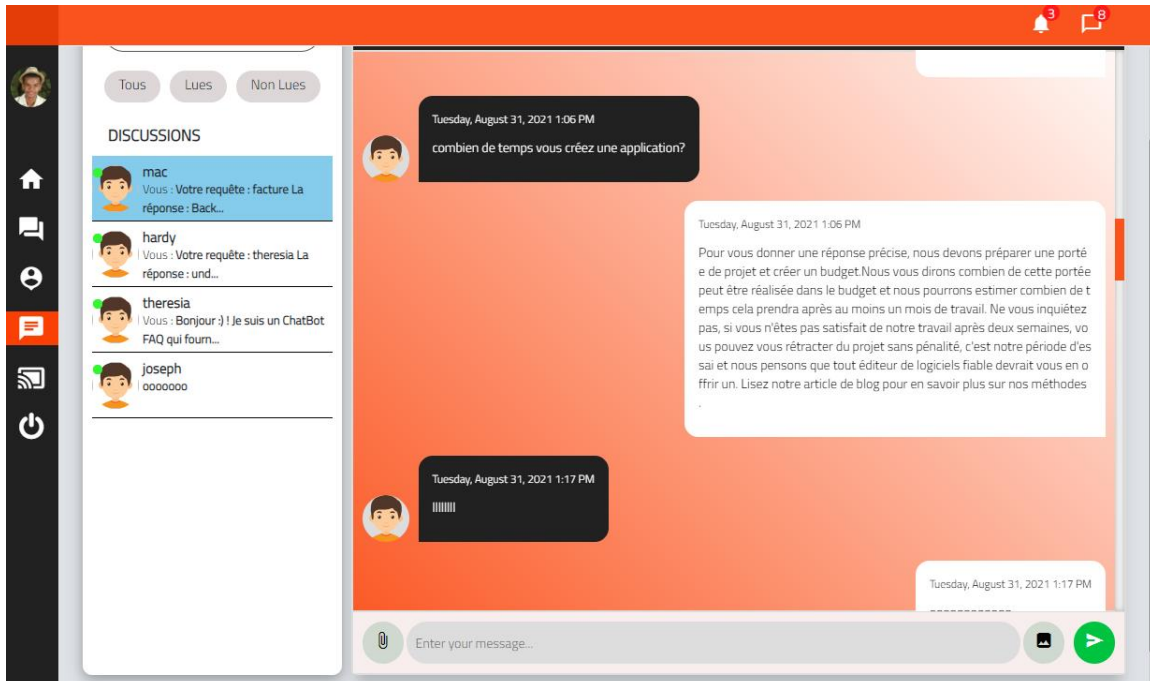
Bot answer
Pour vous donner une réponse précise, nous devons préparer une portée de projet et créer un budget. Nous vous dirons combien de cette portée peut être réalisée dans le budget et nous pourrions estimer combien de temps cela prendra après au moins un mois de travail. Ne vous inquiétez pas, si vous n'êtes pas satisfait de notre travail après deux semaines, vous pouvez vous rétracter du projet sans pénalité, c'est notre période d'essai et nous pensons que tout éditeur de logiciels fiable devrait vous en offrir un. Lisez notre article de blog pour en savoir plus sur nos méthodes.

New answer...

MODIFIER SUPPRIMER RETOUR

Figure 54 : Page d'édition d'une intention

La figure 55 affiche la page de discussion instantanée avec l'internaute.



Tous Lues Non Lues

DISCUSSIONS

mac
Vous : Votre requête : facture La
réponse : Back...

hardy
Vous : Votre requête : thesia La
réponse : und...

theresia
Vous : Bonjour :) ! Je suis un ChatBot
FAQ qui four...

joseph
oooooooo

Tuesday, August 31, 2021 1:06 PM

combien de temps vous créez une application?

Tuesday, August 31, 2021 1:06 PM

Pour vous donner une réponse précise, nous devons préparer une portée de projet et créer un budget. Nous vous dirons combien de cette portée peut être réalisée dans le budget et nous pourrions estimer combien de temps cela prendra après au moins un mois de travail. Ne vous inquiétez pas, si vous n'êtes pas satisfait de notre travail après deux semaines, vous pouvez vous rétracter du projet sans pénalité, c'est notre période d'essai et nous pensons que tout éditeur de logiciels fiable devrait vous en offrir un. Lisez notre article de blog pour en savoir plus sur nos méthodes.

Tuesday, August 31, 2021 1:17 PM

Enter your message...

Figure 55 : Page de discussion instantanée

La figure 56 affiche la liste des sessions (internauts) ayant interagit avec le chatbot.

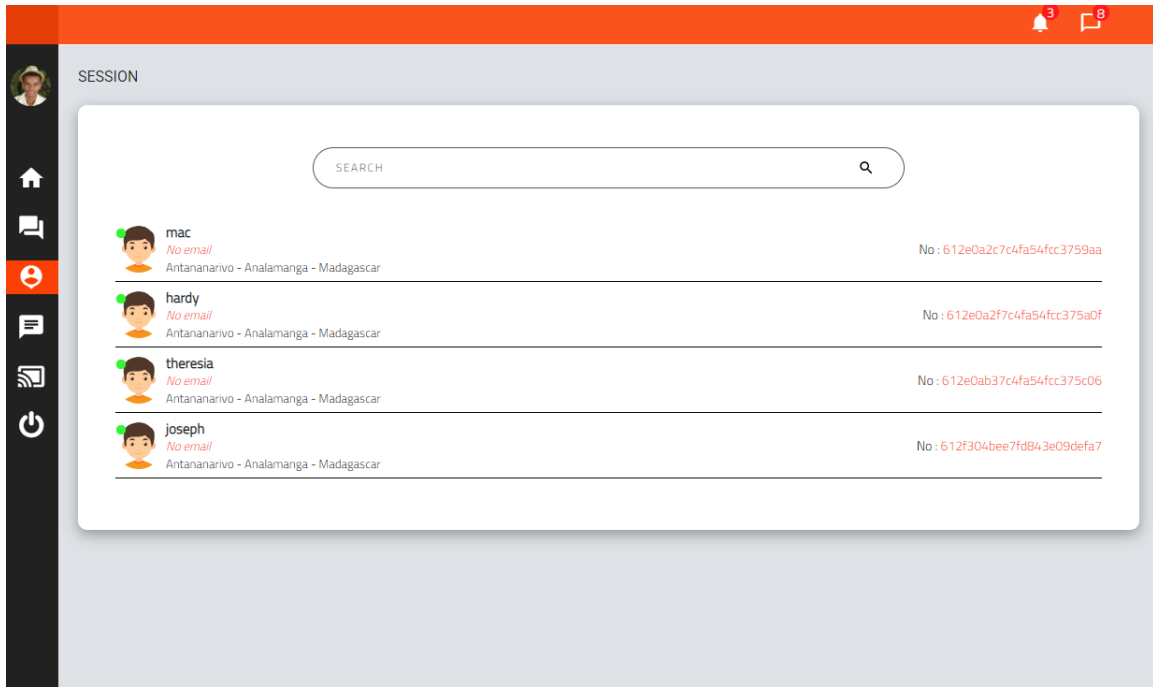


Figure 56 : Liste des sessions internautes

La figure 57 illustre la liste des messages non comprises par le chatbot.

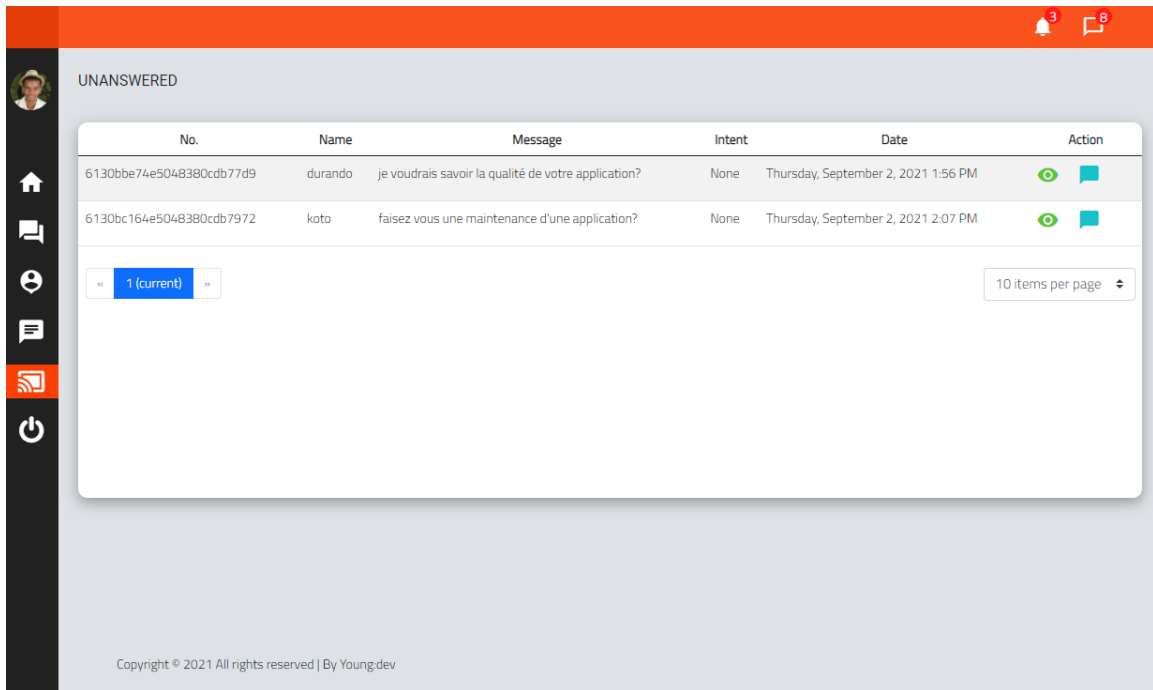


Figure 57 : Liste des messages non comprises par le chatbot

La figure 58 affiche la zone d'édition de l'intention d'une message non comprises par le chatbot.

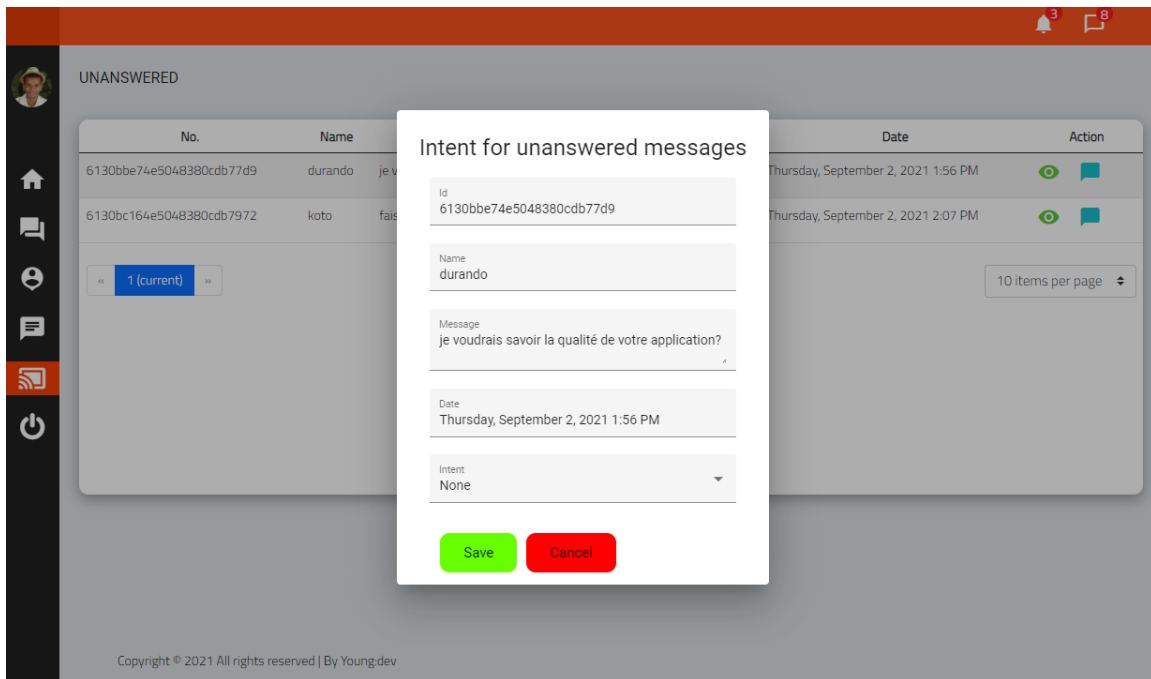


Figure 58 : Zone d'édition de l'intention d'un message non comprise

Enfin la figure 59 affiche la page de présentation de l'entreprise YoungDev ayant une boîte de discussion avec l'internaute en bas à droite (front-office).



Figure 59 : Boite de discussion de l'internaute avec le chatbot

CONCLUSION

Ce stage s'est déroulé au sein de l'entreprise Youngdev basé à Antananarivo où le désir d'améliorer le système de messagerie client est née. Nous avons conçu un agent conversationnel permettant de gérer les requêtes des visiteurs du site web de l'entreprise Youngdev.

Pour réaliser ce projet, on a utilisé la méthode de conception 2TUP, l'outil de modélisation Visual Paradigm, le langage de script Node.js, le SGBD MongoDB et enfin NLP (Natural Language Processing) comme algorithme d'intelligence artificiel permettant de comprendre le langage naturel du visiteur.

L'application permet de poser des questions à l'agent conversationnel afin que ce dernier puisse y répondre. Elle permet aussi à un agent humain d'apporter des réponses face aux requêtes non comprises par l'agent conversationnel. De plus, l'application fournit un système de gestion de la base de connaissance de l'agent conversationnel via la gestion des intentions de ce dernier. Enfin l'application fournit un statistique des données manipulés par cette dernière.

Ce logiciel est opérationnel et répond aux besoins des visiteurs et aussi de l'entreprise. Néanmoins, dans la perspective, on envisage de l'améliorer en y intégrant d'autres fonctionnalités comme un indicateur de performance de l'agent conversationnel basé sur le retour de la satisfaction du visiteur et aussi un système de suggestion de réponses trié par ordre d'importance relatif à un requête précis et enfin on souhaite aussi optimiser l'apprentissage automatique du chatbot afin que celui-ci soit plus performant que possible.

Ce stage nous a permis de nous familiariser avec la vie de l'entreprise et d'approfondir nos connaissances pratiques en matière de création des agents conversationnels.

BIBLIOGRAPHIE

- 1) Joseph et David GABAY, 2008, Analyse et Conception, Paris, Dunod, 242p.
- 2) Pascal Roques et Franck Vallée, 2007, UML 2 en action, Paris, Eyrolles
- 3) Cédric Exbrayat, Become a ninja with Angular

WEBOGRAPHIE

- 4) <https://www.conseilsmarketing.com/chatbot-et-callbot/projet-de-chatbot-le-guide-de-a-a-z-pour-creer-son-premier-chatbot/>, Projet de Chatbot le guide de A à Z pour créer son premier chatbot , Consulté le 15 avril 2021
- 5) <https://research.aimultiple.com/faq-chatbot/>, FAQ Chatbot: Types, Goals and Use Cases in 2021, Consulté le 30 avril 2021
- 6) <https://www.revechat.com/blog/chatbot-vs-live-chat/>, Chatbot vs Live Chat: The Pros & Cons for Your Business, Consulté le 12 mai 2021
- 7) www.angular.io , site officiel pour la documentation d'Angular, consulté le 15 mai 2021
- 8) www.w3school.com, HTML5, CSS3 & Javascript, Consulté le 20 mai 2021
- 9) <https://material.angular.io>, site officiel d'Angular Material, Consulté le 21 mai 2021
- 10) <https://mongoosejs.com/docs/>, site officiel de Mongoose, Consulté le 25 mai 2021
- 11) <https://morioh.com/p/7873ece1c94d> , How to Build CRUD Application with Angular and NgRx, Consulté le 27 Mai 2021
- 12) <https://morioh.com/p/62756071e405>, How to create REST APIs in Node.js using TypeScript, Consulté le 10 juin 2021
- 13) <https://github.com/axa-group/nlp.js/>, Documentation de Node-nlp, Consulté le 15 juin 2021
- 14) <https://www.userlike.com/fr/blog/flux-de-conversation-chatbot>, 6 étapes pour un flux de conversation de chatbot fluide, Consulté 09 juillet 2021
- 15) <https://www.futuremind.com/blog/22-frequently-asked-questions-about-software-dev>, 26 Frequently Asked Questions About Software Development, Consulté le 27 juillet 2021

GLOSSAIRE

Agent conversationnel : c'est un robot conversationnel, mais son domaine d'application est très riche. Ainsi, il est capable d'accomplir d'autres tâches plus intelligentes comme l'analyse sémantique ou thématique d'un texte, la recherche et la synthèse d'informations, l'identification des émotions et des sentiments de son interlocuteur, etc.

Agent humain : (ou l'administrateur) C'est celui qui répond aux requêtes non comprises par le chatbot. C'est aussi celui qui administre le chatbot

Base de connaissances : Il regroupe des connaissances spécifiques à un domaine spécialisé donné, sous une forme exploitable par un ordinateur.

Chatbot : C'est un programme informatique qui communique par l'écrit en langage naturel. Bot est une abréviation du mot robot. Un **chatbot** peut prendre la forme d'un agent virtuel sur un site en ligne ou bien désigner un assistant pour applications mobiles, applications de **messagerie** ou téléphone.

Foire aux questions : c'est une liste faisant la synthèse des **questions** posées de manière récurrente sur un sujet donné, accompagnées des réponses correspondantes, que l'on rédige

Intention : Ils permettent au chatbot de comprendre la question posée par l'utilisateur. Il y a 2 manières de détecter une intention : détection par modèle **NLP**, détection par **mots clés**

Internaute : Appelé aussi visiteur et aussi client est celui qui surfe sur internet. Il peut visiter à tout moment le site web de l'entreprise.

NLP : Le NLP est la branche du machine learning qui utilise des modèles capables de comprendre les **textes**. Les modèles NLP sont entraînés pour apprendre à faire le lien entre le langage humain et les ordinateurs. Leur but principal est d'apporter la meilleure réponse possible à l'utilisateur en garantissant la compréhension des **saisies libres** (via la détection d'intentions ou de mots-clefs).

Requête : Question ou demande envoyé par l'internaute.

Réponse : C'est le résultat que le chatbot envoie suite à un requête utilisateur.

TABLE DES MATIERES

CURRICULUM VITAE	I
SOMMAIRE	IV
REMERCIEMENTS	VI
LISTE DES FIGURES	VII
LISTE DES TABLEAUX	IX
NOMENCLATURE	X
INTRODUCTION	1
PARTIE I : PRESENTATIONS	2
Chapitre 1. PRESENTATION DE L'ENI	3
1.1. Informations d'ordre général	3
1.2. Missions et historiques.....	3
1.3. Organigramme institutionnel de l'ENI	5
1.4. Domaines de spécialisation.....	6
1.5. Architecture des formations pédagogiques	7
1.6. Relations de l'ENI avec les entreprises et les organismes	10
1.7. Partenariat au niveau international.....	11
1.8. Débouchés Professionnels des Diplômés.....	13
1.9. Ressources Humaines	15
Chapitre 2. PRÉSENTATION DE YOUNGDEV	16
2.1. Historique et brève présentation de la société YoungDev	16
2.2. Fiche d'identification	16
2.2.1 Statut juridique	16
2.2.2 Direction.....	16
2.2.3 Localisation et contacts	16
2.3. Services	17
2.4. Organigramme	18
2.5. Valeurs	18
Chapitre 3. DESCRIPTION DU PROJET.....	19
3.1. Formulation.....	19
3.2. Objectif et besoins de l'utilisateur	19
3.3. Moyens nécessaires à la réalisation du projet	20
3.3.1 Moyens matériels	20
3.3.2 Moyens humains	20

3.3.3 Moyens logiciels	20
3.4. Résultats attendus.....	21
3.5. Chronogramme des travaux	21
PARTIE II : ANALYSE ET CONCEPTION	22
Chapitre 4. ANALYSE PREALABLE	23
4.1. Analyse de l'existant.....	23
4.1.1 Organisation actuelle.....	23
4.1.2 Inventaire des moyens matériels	23
4.2. Critique de l'existant.....	23
4.3. Conception avant-projet.....	24
4.3.1. Proposition de solutions	24
4.3.2. Méthodes.....	25
4.3.3. Outils proposés.....	25
Chapitre 5. ANALYSE CONCEPTUELLE	35
5.1. Dictionnaire des données	35
5.2. Règles de gestion	37
5.3. Représentation et spécification des besoins	37
5.3.1. Diagramme des cas d'utilisation.....	37
5.3.1. Priorisation des cas d'utilisation	49
5.3.2. Diagramme de séquences système pour chaque cas d'utilisation.....	50
5.4. Spécification des besoins techniques	58
5.5. Modélisation du domaine.....	59
Chapitre 6. CONCEPTION DETAILLEE.....	61
6.1. Architecture du système.....	61
6.2. Diagramme de séquence de conception pour chaque cas d'utilisation.....	62
6.3. Diagramme de classe de conception pour chaque cas d'utilisation	68
6.4. Diagramme de classe de conception global	72
6.5. Diagramme de paquetages	75
6.6. Diagramme de déploiement	78
PARTIE III : REALISATION	81
Chapitre 7. Mise en place de l'environnement de développement	82
7.1. Installation et configuration des outils.....	82
7.2. Architecture de l'application.....	87
Chapitre 8. DÉVELOPPEMENT DE L'APPLICATION	88

8.1. Création de la base de données	88
8.2. Codage de l'application	90
8.3. Présentation de l'application.....	93
CONCLUSION	100
BIBLIOGRAPHIE	XI
WEBOGRAPHIE	XII
GLOSSAIRE	XIII
TABLE DES MATIERES	XIV
RESUME	XVII
ABSTRACT	XVII

RESUME

Dans ce présent projet nous avons conçu et élaboré un chatbot permettant de gérer une foire aux questions concernant les produits et services fournis par l'entreprise Youngdev (FAQ).

Le chatbot crée a pour but de répondre automatiquement dans le plus bref délai aux requêtes envoyés par les visiteurs.

Pour la réalisation de ce projet, nous avons utilisé la méthode de conception 2TUP, le langage de programmation Node.js puis MongoDB comme SGBD, et enfin NLP.js comme algorithme pour la compréhension du langage naturel de l'homme.

Mots clés : chatbot , FAQ, langage, NLP.js, produits, répondre , requêtes, services, visiteurs

ABSTRACT

In this project, we have designed and built a chatbot to manage frequently asked questions about products and services provided by the company Youngdev (FAQ).

The purpose of the chatbot is to automatically respond as quickly as possible to requests sent by visitors.

For the realization of this project, we used the 2TUP design method, the programming language Node.js then MongoDB as DBMS, and finally NLP.js as an algorithm for understanding human natural language.

Keywords: chatbot, FAQ, language, NLP.js, products, respond, requests, services, visitors