

Université M'hamed Bougara Boumerdes
Faculté des Sciences de l'Ingénieur (FSI, exINGM)



Condor Electronics
Entreprise

Filière : Génie électrique
Spécialité : Systèmes informatiques distribués

CONCEPTION ET RÉALISATION D'UN CHATBOT FACEBOOK

AMGHAR ABDENOUR
ELHADDAD ADEL

*Mémoire présenté pour l'obtention du diplôme
Master en Systèmes Informatiques Distribués*

Soutenue le 20/06/2017 en présence de :

Dr.	Mohamed Amine RIAHLA	<i>Président</i>	UMBB
Pr.	Mhamed HAMADOUCHE	<i>Examineur</i>	UMBB
Mme.	Samira MECHID	<i>Promoteur</i>	UMBB

Remerciements :

En guise de reconnaissance, nous tenons à témoigner nos sincères remerciements à toutes les personnes qui ont contribué de près ou de loin au bon déroulement de notre stage de fin d'étude et à l'élaboration de ce modeste travail.

Nos sincères gratitudes à tous nos enseignants pour la qualité de leurs enseignement, et leurs conseils et intérêt incontestable qu'il porte à tous les étudiants.

Nous tenons à remercier l'ensemble du personnel de Condor Electronics pour leur patience, leurs conseils pleins de sens et pour le suivi et l'intérêt qu'ils ont portés à nos travaux.

Dans l'impossibilité de citer tous les noms, nos sincères remerciements vont à tous ceux et celles, qui de près ou de loin, ont permis par leurs conseils et leurs compétences la réalisation de ce mémoire.

Enfin, nous n'oserions oublier de remercier tout le corps professoral de département infotronique, pour le travail énorme qu'il effectue pour nous créer les conditions les plus favorables pour le déroulement de nos études.

Dédicaces

À nos parents

À nos frères

À nos amis

Table des matières

Introduction générale	2
1 Présentation de l'organisme d'accueil	4
1.1 Introduction	4
1.2 Présentation de Condor Electronics	4
1.2.1 Condor Electronics	4
1.2.2 Historique	4
1.3 La direction High City	5
1.3.1 Présentation	5
1.4 Fiche technique de condor	5
1.4.1 Carte d'identité	5
1.4.2 les unités qui composent l'entreprise :	5
1.4.3 Statistiques des effectifs	7
1.5 Conclusion	7
2 Étude de l'existant	8
2.1 Introduction	8
2.2 Présentation du cadre de stage	8
2.3 Système actuel	8
2.3.1 Sources d'informations	9
2.4 Critiques sur le processus actuel	11
2.5 Solution proposée	11
2.6 Conclusion	11
3 Agents conversationnels	13
3.1 Introduction	13
3.2 Un chatBot pour améliorer la relation client-entreprise	13
3.3 Définition chatBot	13
3.4 Types chatBots :	14
3.5 Traitement du langage naturel	14
3.6 L'intelligence artificielle et les chatBots	15
3.6.1 AIML	16
3.6.2 ChatterBot	21
3.6.3 Réseaux de neurones	30
3.7 Avantages et Inconvients	34

3.7.1	AIML	34
3.7.2	ChatterBot	34
3.7.3	Réseau de neurones	34
3.8	Test de Turing	35
3.9	Conclusion	35
4	API Facebook	36
4.1	Introduction	36
4.2	Qu'est ce que Facebook ?	36
4.3	Facebook, pourquoi faire ?	36
4.4	Facebook Messenger Plateform	37
4.5	Principe de fonctionnement	37
4.5.1	Mise en place :[?]	37
4.5.2	Fonctionnement :	38
4.6	Types de contenu	39
4.6.1	Non structuré	39
4.6.2	Structuré	41
4.7	Divers[?]	43
4.7.1	Réponse rapide	43
4.7.2	Menu persistant	44
4.8	Conclusion	45
5	Conception et réalisation	46
5.1	Méthodologie et approche adoptée :	46
5.1.1	Définition d'UML	46
5.2	Cahier des charges	46
5.3	Conception	48
5.3.1	introduction	48
5.3.2	Présentation du projet :	48
5.4	Réalisation	58
5.4.1	Introduction :	58
5.4.2	Environnement du travaux :	58
5.4.3	Choix des langages et outils de développement	58
5.4.4	Langage de programmation	58
5.4.5	Outils de développement	59
5.4.6	Mise en place du chatBot	61
5.5	Conclusion	66

Table des figures

1.1	Organigramme structurel de la société [?]	6
1.2	Statistiques sur Condor Electronics en 2016	7
2.1	Organigramme système actuel	10
3.1	DL Étape 1	25
3.2	DL Étape 2	25
3.3	DL Étape 3	26
3.4	Organigramme fonctionnement ChatterBot[?]	29
3.5	Représentation d'un réseau de neurones [?]	30
3.6	Représentation d'un réseau de neurones	31
3.7	Mode supervisé	32
3.8	Mode non supervisé	32
3.9	Mode surapprentissage	33
4.1	Exemple Texte	39
4.2	Exemple Fichier	40
4.3	Exemple Boutons	41
4.4	Exemple Réponses rapides	43
4.5	Exemple Menu persistant	44
5.1	Diagramme de déploiement	48
5.2	Diagramme de cas d'utilisation	49
5.3	Diagramme de séquence d'authentification	50
5.4	Diagramme de séquence de recherche	51
5.5	Diagramme de séquence contrôle du chatBot	52
5.6	Diagramme de séquence gestion du chatBot	54
5.7	Diagramme de gestion d'utilisateurs	55
5.8	Diagramme de séquence controle globale du bot	56
5.9	Diagramme de séquence envoi de message	57
5.10	Organigramme de fonctionnement du chatBot	63
5.11	capture d'écran de l'interface Principale	65
5.12	capture d'écran d'une discussions a partir de Messenger	66

Introduction générale

[?]

De nos jours, 55 % des internautes utilisent au quotidien une messagerie instantanée. le webchat ¹ et le social média constituent les premiers canaux de contacts.

Parmi les 5 applications les plus téléchargées, quatre sont des applications de messagerie instantanée. Un milliard d'utilisateurs de Messenger ont enregistré leurs identifiants Facebook dans leur mobile.

Parler à une personne est une tâche aisée pour les humains. En est-il de même pour une machine ? Ceci définit la problématique.

L'IA ² touche à tout ce qui va se rapprocher d'un comportement humain, les récentes évolutions techniques autour de l'IA permettent enfin aux machines de nous comprendre et de nous répondre de manière instantanée et sans rupture. Comprendre le langage, pouvoir formuler des phrases, dégager du sens et des émotions à travers du langage écrit et oral sont au cœur de l'IA. L'expérience d'interactions avec un logiciel informatique relève de vraies opportunités .

Grâce au développement récent ,Condor Electronics cherche à développer un chat-Bot ³ pour gérer le grand nombre des messages reçus sur sa page Facebook. Afin de réduire les coûts et mieux fidéliser ces clients , un chatBot devrait rapporter des réponses en temps réel avec une disponibilité 24/7. La création d'un chatBot capable d'agir d'une IA assez sophistiquée, capable d'interagir avec des humains de différentes manières, qui se comporte comme un agent humain on répondent aux questions des clients est une nécessité pour Condor Electronics.

L'IA assure cette tâche avec un ensemble de techniques permettant d'imiter le comportement humain d'avoir une intelligence indistinctement de celle de l'homme dans une conversation.

1. Un échange de messages en ligne en temps réel
2. Intelligence Artificiel
3. Agent conversationnel

La réalisation d'un tel chatBot fait l'objet de notre étude et pour le mener à bien , nous allons le diviser en cinq chapitres :

Le premier chapitre « **Présentation de l'organisme d'accueil** » qui représente un tracé et un historique sur de l'organisme d'accueil Condor Electronics .

Le deuxième chapitre « **Étude de l'existant** » qui se penche sur les principaux aspects du fonctionnement actuel et de déduire les insuffisances ainsi que la solution proposée .

Le troisième chapitre « **Agents conversationnels** » décrit des généralités sur les chatBots ainsi que les différentes techniques permettent la création de ce dernier .

Le quatrième chapitre « **API Facebook** » décrit le principe de fonctionnement d'un chatBot facebook.

Le cinquième chapitre « **Conception et réalisation** » représente une description détaillée des exigences de l'entreprise et la conception de l'application de gestion du chatBot , ce chapitre représentera aussi les outils et langages utilisés et les captures des interfaces de l'application et du chatBot.

Enfin ce mémoire sera achevé par une conclusion générale .

Chapitre 1

Présentation de l'organisme d'accueil

1.1 Introduction

Le stage de fin d'études, comme son nom l'indique, s'effectue à la fin du cursus de formation, sert à bien faire le lien entre les connaissances théoriques acquises et l'application dans un contexte de projet de fin d'études, ce dernier est proposé soit par l'organisme soit par l'intéressé afin d'attendre ses objectifs et ses besoins. Pour cela, il est primordial d'avoir une vision globale sur l'organisme dans le but de faire un pas vers la réalisation du projet, Dans ce premier chapitre nous allons faire une bref présentation de la société Algérienne Condor.

1.2 Présentation de Condor Electronics

1.2.1 Condor Electronics

SPA¹ Condor Electronics, est une société spécialisée dans la fabrication de : (équipement électronique et électroménager, informatique, agroalimentaire, emballage, matériaux de construction et commerce international.), ce qui a permis à Condor Electronics de brasser un gros volume d'affaires et de projets.

1.2.2 Historique

À l'origine, Condor était une petite affaire de commerce de denrées alimentaires et de transport, fondée et façonnée par le patriarche de la famille, Monsieur El Hadj Mohamed Taher Benhamadi.

Grâce à ses réflexes du métier, un esprit commercial et un sens de la création très éveillé, El Hadj Mohamed Taher, a tracé la première voie qui a mené à la création du Groupe Benhamadi.

1. SPA : Société par action

Aujourd'hui, le groupe représente un des conglomérats d'entreprises Algériennes les plus puissantes et actives sur la sphère économique du pays. Il opère dans différents domaines d'activité et affiche des résultats dignes d'être cités en exemple.

1.3 La direction High City

1.3.1 Présentation

High City est une direction qui rapporte la valeur ajoutée en matière d'application mobile, desktop ou web pour les produits Condor Electronics c'est à dire développement des outils ou applications préinstallées pour les clients de Condor SPA aussi il satisfait les besoins de la société .

1.4 Fiche technique de condor

1.4.1 Carte d'identité

Slogan de l'entreprise : "Prenez votre envol"

Date de création : 09/02/2002

Date début de l'activité : Octobre 2002

Date début de la production : 23/11/2002

Date lancement de la marque : 30/04/2003

Capital social : 2450 000 000 DA

Superficie totale : 80 104 m²

Nature d'activité : la fabrication, la commercialisation et le service après-vente des appareils électroniques, articles électroménager, des panneaux solaires et multimédia

Directeur général : Ben Hammadi Amar

Président du conseil d'administration : Ben Hammadi Abderrahman

Adresse : zone industriel, rue de M'sila, division 70 branche 161 Bordj Bou Arreridj, 34000, Algérie

Site web : www.condor.dz

Condor Electronics Company a obtenu plusieurs certificats en matière de normes de qualité International(ISO 9001, ISO 14001 et OHSAS 18001)

1.4.2 les unités qui composent l'entreprise :

1. Unité des climatiseurs et des machines à laver.
2. Unité des réfrigérateurs.
3. Unité de conversion en matière plastique.
4. Unité des produits blancs.
5. Unité des produit bruns.
6. Unité de matériel informatique.

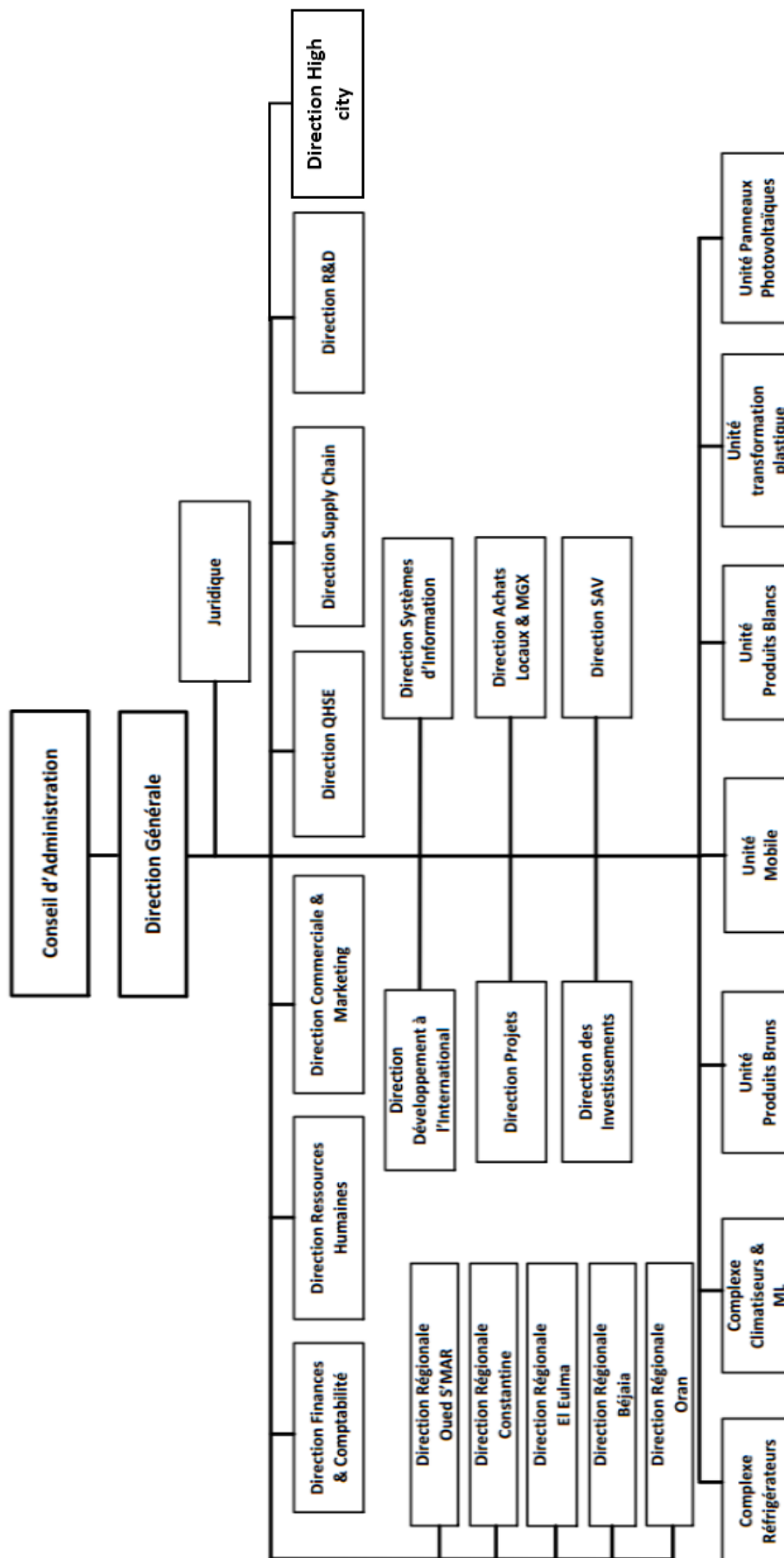


FIGURE 1.1 – Organigramme structurel de la société [?]

7. Unité de l'industrie des panneaux solaires.

1.4.3 Statistiques des effectifs

Effectifs par genre			
Année	Homme	Femme	Total
Fin 2016	5291	889	6180

Effectif par situation familiale					
Année	Célibataire	Marié(e)s	Divorcé(e)s	Veuf (ev)s	Total
Fin 2016	2453	3611	98	17	6180

Effectifs par catégorie socioprofessionnelle				
Année	Cadres	Maitrise	Exécution	Total
Fin 2016	740	1673	3767	6180

Effectifs par type de contrat					
Année	CDD	CDI	CTA	DAIP	Total
Fin 2016	3194	2882	80	24	6180

FIGURE 1.2 – Statistiques sur Condor Electronics en 2016

1.5 Conclusion

Ce chapitre nous a servi à mettre le projet dans son cadre et nous a permis d'avoir une vision globale sur Condor Electronics. Le chapitre suivant nous aborderons le mode de fonctionnement du système actuel et de déduire les insuffisances de ce dernier

Chapitre 2

Étude de l'existant

2.1 Introduction

L'étude de l'existant est une phase très importante dans le projet , elle consiste à mettre à plat de façon aussi claire que possible le fonctionnement du système actuel et poser le contexte dans lequel le stage se déroule . Cette étude permet de faciliter la reconnaissance des besoins. Le but de ce chapitre est de déduire les insuffisances ou problèmes afin d'améliorer la qualité du service clientèle.

2.2 Présentation du cadre de stage

Le stage faisant l'objet de projet de fin d'études Master porte le thème de « Conception et Réalisation d'un ChatBot Facebook » déroulé dans l'entreprise Condor Electronics au niveau du service High city .High City est un service qui a pour but d'apporter la valeur ajoutée software pour les produits de Condor Electronics . Afin de répondre aux besoins du service après-vente (SAV)¹ qui gère les pages Facebook de la société , le thème proposé permet de faciliter la gestion de ces pages Facebook et répondre instantanément et automatiquement aux messages répétitifs en temps-record.

2.3 Système actuel

Condor Electronics dispose actuellement d'un SAV qui utilise une solution externe(social media) permettant de stocker des questions / réponses , si la question posée par le client via Facebook correspond à une déjà enregistrée le manager de la page fait juste un "drag and drop"² sur la page afin de répondre .

1. SAV : Service après-vente

2. drag and drop :glisser-déposer

2.3.1 Sources d'informations

Les informations mentionnées ci-dessous permettent de décrire le fonctionnement actuel :

- Le chef du service connecte plus de 25 machines à la page Facebook de Condor afin de ne pas distribuer le mot de passe de la page avec tous les employés du service .
- Chaque employé ouvre la solution externe sur sa machine et commence à répondre aux messages reçus sur la page .
- Si des questions sont déjà posées sur la page , l'employé fait une recherche sur la solution externe et s'il trouve une réponse déjà enregistrée il fait un "drag and drop" sur le champs de saisi de la page Facebook .
- Si la question n'existe pas , il saisit sa propre réponse ou demande au chef du service la réponse exacte et ce dernier l'enregistre sur la solution .
- Les clients de Condor Electronics ont le droit d'avoir une réponse que durant les horaires de travail.
- Tous les clients ont la possibilité de faire des appels aux SAV donc beaucoup d'appels téléphoniques sont reçus pour obtenir des informations .

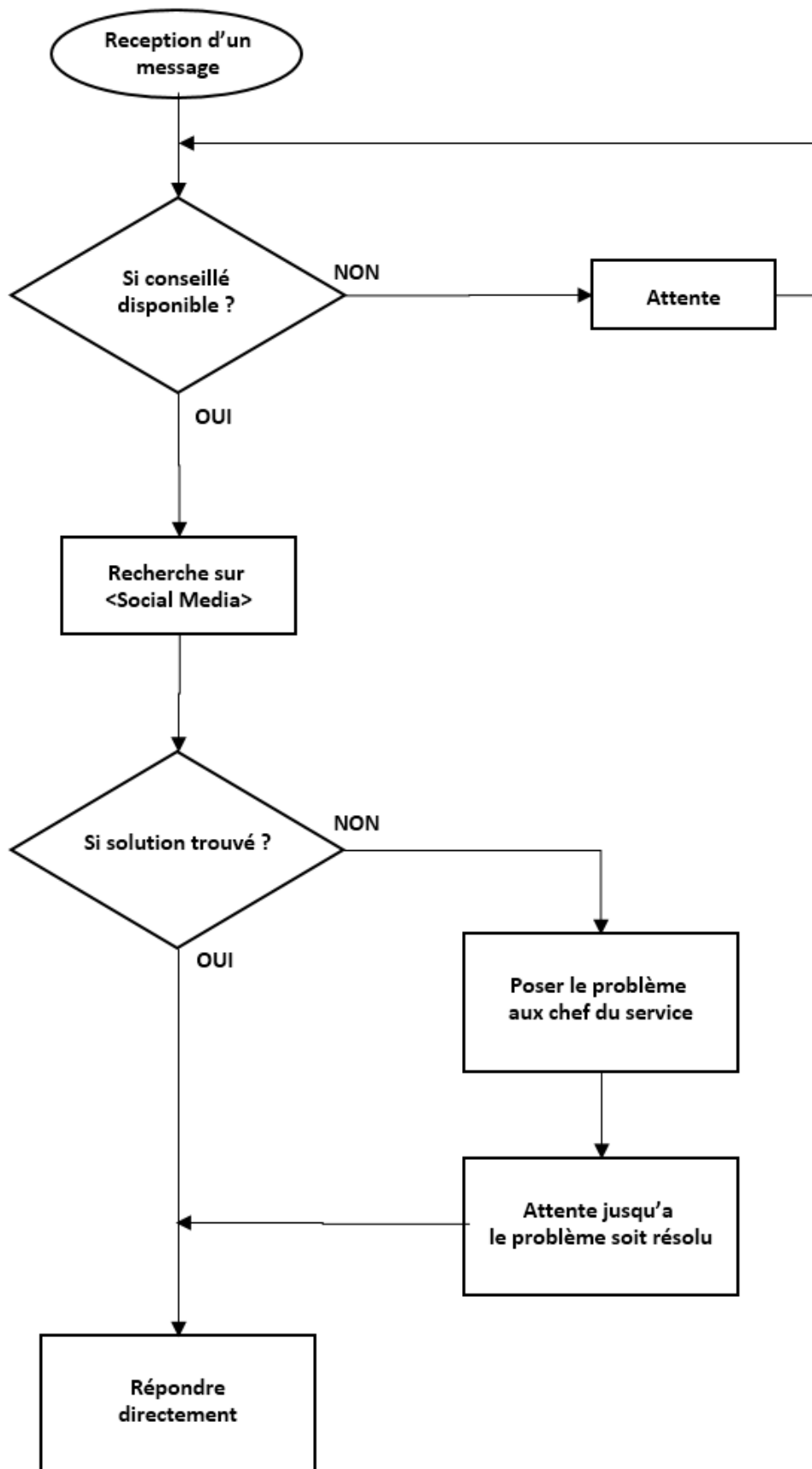


FIGURE 2.1 – Organigramme système actuel

2.4 Critiques sur le processus actuel

le processus actuellement adopté n'assure pas la disponibilité immédiate des réponses aux clients au temps voulu. Ce processus ne répond pas à tous les besoins exprimés par l'entreprise :

- Latence : le temps de réponse dépend du nombre des messages reçus .
- Persistance : les heures de travail sont fixées de 8h du matin à 16h , donc aucun employé ne sera disponible pour répondre en dehors de cette période.
- Qualité inégale : la qualité des réponses entre un employé et un autre est différente même si on assure la formation en continue des employées.
- Temps d'attente : il y a une forte possibilité qu'un client n'aura pas une réponse au temps voulu .
- Financier : plus les produits de Condor Electronics se multiplient ,les clients aussi et les questions de ces derniers aussi , ce qui oblige la société à recruter plus d'employés dans SAV.

2.5 Solution proposée

L'idée est de proposer une aide active sans monopoliser un grand nombre de conseillers . Cette aide devrait être proposée au client au moment voulu et non quelque temps après. D'où vient l'idée de créer un Chabot (un agent conversationnel) dans la langue de Shakespeare,ce dernier peut guider l'utilisateur rapidement afin de lui trouver la solution sur Facebook de ce fait une optimisation de temps, de coût et de qualité des réponses.

2.6 Conclusion

Ce chapitre nous a permis de bien comprendre les insuffisances du système actuel , il nous a ouvert la voie au processus de conception de nouveau système . Dans le chapitre suivant nous allons aborder une étude des concepts théoriques et rassembler le maximum d'informations sur les chatBots .

*« On peut apprendre à un ordinateur à dire :
« Je t'aime »,
mais on ne peut pas lui apprendre à aimer. ».*
Albert Jacquard

Chapitre 3

Agents conversationnels

3.1 Introduction

Ce chapitre a pour but de déterminer de façon détaillée et précise ce qu'est le chatBot et ce qui devrait faire, ainsi que son rôle par rapport au service clientèle. Pour satisfaire de réels besoins qui se font ressentir du côté client et aussi de côté société, la mise en place d'un chatBot est indispensable.

3.2 Un chatBot pour améliorer la relation client-entreprise

Un nouveau canal commercial? , Oui .Les chatBots sont capables d'aider et de guider les consommateurs sur des problématiques simples. Les chatbots ne servent pas uniquement un but commercial. Les chatbots peuvent également être des aides au service client. Là encore, les marques partent d'un constat simple : tout le monde a eu une expérience insatisfaisante avec un service client. Là encore, le chatBot est conçu comme un moyen qui facilite la vie du client en proposant un service après-vente (SAV) simplifié et direct. Plus besoin de passer forcément par un centre téléphonique, étape qui rime souvent avec attente

3.3 Définition chatBot

Afin d'établir une conversation entre homme et machine, une conversation homme-machine en langage naturel est utilisée, c'est cela que l'on appelle communément un chatBot ou agent conversationnel . Un programme informatique écrit en langage quelconque répond instantanément, en se basant sur un corpus (ensemble de question / réponse) .

3.4 Types chatBots :

les agents conversationnels sont classés par la représentation de leur connaissance [?] :

- Reformulation : Aucune connaissance et se limitent à reformuler les questions .
- Base de connaissances fixe "Prédéfini" : Capable de répondre uniquement aux questions dont la réponse est contenue dans sa base de connaissance.
- Base de connaissances évolutive "Autoapprenant" : certains agents ont la capacité d'apprendre au fil des conversations avec leur interlocuteur et de mémoriser les informations.

Suivant l'expérience d'interactions, On distingue principalement quatre « familles » de Chatbots [?] :

- Chatbot serviciel : apporte un service précis à l'utilisateur.
- Chatbot expérientiel : fait vivre une expérience.
- Chatbot commercial : réalise une transaction commerciale.
- Chatbot entertainment : fonctionne comme divertissement.

3.5 Traitement du langage naturel

Le langage naturel ne désigne pas une langue à proprement parler, mais la manière naturelle de s'exprimer des êtres humains, par opposition au langage binaire et aux langages utilisés en programmation. C'est le langage des mails, des descriptions, du chat...

C'est une discipline à la frontière de la linguistique[?], de l'informatique et de l'intelligence artificielle, qui concerne l'application des programmes et des techniques informatiques à tous les aspects du langage humain. Mais afin que le chatBot comprend ce que l'on souhaite, il est nécessaire que l'entrée brute soit processé pour en déduire l'intention et des mots clés.

Prenons l'exemple courant d'un chatBot serviciel météo. Lorsqu'on demande « quel temps fait-il à Alger ? », il faut que la machine comprenne que l'intention de l'utilisateur et d'obtenir la météo. Le contexte de Alger doit être aussi pris en compte afin qu'il puisse répondre de manière pertinente et ne donne pas une réponse ne

concernant pas la capitale Algérienne.

il existe plusieurs méthodes reposent sur l'identification d'un ou plusieurs mots-clé, c'est-à-dire des mots que l'on considère comme essentiels dans la réponse comme "pattern matching".

La procédure de pattern matching consiste en une mise en correspondance, mot par mot, caractère par caractère, du message de l'utilisateur et du 'pattern' spécifié par le concepteur.

Par défaut, le matching n'autorise aucune différence entre les caractères du pattern et du message, y compris des différences mineures telles que la présence d'une majuscule au lieu d'une minuscule. Toute tolérance dans la mise en correspondance doit être spécifiée par le concepteur.

3.6 L'intelligence artificielle et les chatBots

[?]

L'étude suivante présente les différentes techniques permettent la création d'un chatBot. De plus, ces techniques ont été choisis parmi d'autres car leur algorithme principal sont accessibles . Ils permettent ainsi de nous faire une idée sur les solutions que nous pouvons élaborer. Par ailleurs il existe des outils intéressants et relativement simples d'utilisation. Mais, ils restent très limités et leurs algorithmes principaux sont inaccessibles à titre d'exemple :

- **Chatfuel** : Un outil, qui permet quant à lui de créer des bots pour Facebook Messenger . Il s'agit d'un outil destiné aux débutants, à celles et ceux qui n'ont vraiment aucunes connaissances en programmation. Cet outil permet d'interpréter le langage naturel.
- **Rebot.me** :
est l'un des seuls outils de création des bots disponibles en version française et aussi gratuit, ce qui est un de ses avantages. Son principe d'utilisation est simple. Lorsque vous créez votre chatBot sur Rebot.me, vous êtes invité à définir un ensemble de questions / réponses.
- **Botsify** : comme Chatfuel, permet de créer un chatBot pour Facebook Messenger. Là encore, la simplicité d'utilisation est l'argument principal mis en avant par l'éditeur. Pas besoin de savoir coder pour générer votre chatBot. L'outil peut comprendre les phrases comportant des fautes d'orthographe ou de syntaxe .

- **Wit.ai** : moteur d'IA gratuit et l'unique outil proposé par facebook afin de créer un chatBot pour Facebook Messenger. Le concept est simple , chaque chatBot wit se base sur des scénarios ou dialogues .Wit.ai propose des librairies dans la plupart des langages qui sont disponibles.

3.6.1 AIML

Définition :

L'AIML[?]¹ est un langage dérivé de XML² utilisé pour gérer la connaissance des bots ,développé par Richard Wallace entre 1995 et 2002 repris par la suite par des communautés Open Source. Ce langage utilise une vingtaine de balises de base. Chaque balise a un rôle bien défini comme les conditions ou enregistrement ... etc. Implémenté avec plusieurs langages de programmation comme Java , Php et Python . Le chatBot le plus célèbre basé sur AIML est A.L.I.C.E³ , il permet de dialoguer avec les visiteurs via une interface web ou par audio .

Principe de fonctionnement :

Chaque chatBot AIML doit absolument avoir des fichiers ".AIML" qui constituent ça connaissance , chaque fichier doit avoir à l'entête deux balises principales qui sont :

```
<?xml version="1.0" encoding="UTF-8"?>
<aiml version="1.0">
```

par la suite les balises suivantes :

category :

la balise `<category>` encadre et définit une catégorie, c'est à dire un ensemble de stimulus-réponse (une question / réponse). Une catégorie comprend une et une seule balise `<pattern>` , au plus une balise`<that>` optionnelle, et une et une seule balise`<template>` . Ce qui donne le code `<aiml>` type suivant :

```
<category>
    <pattern >...</pattern>
    <that>...</that> ( balise facultative )
    <template></template>
</category>
```

1. AIML : Artificial Intelligence Markup Language
 2. XML :Extensible Markup Language
 3. A.L.I.C.E :Artificiel Linguistique Internet Computer Entity

Exemple :

```

<category>
  <pattern> A bientôt </pattern>
  <template> AU REVOIR </template>
</category>

```

pattern :

la balise `<pattern>` sert à indiquer le gabarit d'entrée . L'entrée qui est la question qu'est normalisée puis comparée aux différents gabarits d'entrée si l'un de ces derniers correspond , le modèle de réponse de la catégorie correspondante sera traité et utilisé comme une réponse que le bot utilisera pour répondre.

les gabarits d'entrée sont constitués d'une chaîne alphanumérique complétée par une ou plusieurs caractères joker * ou - .

Exemple :

```

<pattern> BONJOUR * </pattern>
<that >... <that>
<template >... </template>

```

srai :

La balise `<srai>` commande à l'interpréteur AIML de passer le résultat du traitement du contenu de l'élément `srai` à la boucle de correspondance AIML, comme si l'entrée avait été produite sous cette forme par l'utilisateur (en incluant le processus complet de normalisation de l'entrée). La balise `srai` n'a pas d'attributs, et peut contenir n'importe quelle balise AIML utilisable dans un template.

Exemple :

```

<category>
  <pattern>CIAO</pattern>
  <template><srai>AU REVOIR</srai></template>
</category>
<category>
  <pattern>AU REVOIR</pattern>
  <template>A bientôt.</template>
</category>

```

think :

La balise `<think>` sert à masquer, dans un template, l'affichage du texte en sortie produit par une autre balise.

Exemple :

```
<category>

    <pattern>Je suis un homme</pattern>
    <template>
        Merci de m avoir indique quel etait votre sexe.
        <think><set_gender>homme</set_gender></think>
    </template>

</category>
```

La balise `<set_gender>homme</set_gender>` renvoie normalement la chaîne de caractères Homme. Mais comme la balise `<think>` cache le texte produit par `<set_gender>homme</set_gender>` dans la réponse, celle ci devient : Merci de m'avoir indiqué quel était votre sexe.

condition :

La balise condition sert à sélectionner, dans une série de réponses possibles contenues dans un template, celle qui est la plus appropriée compte tenu du contexte. Les conditions permettent d'effectuer des tests sur les valeurs actuelles d'un ou plusieurs prédicats. Il existe trois types de condition :

1. la condition de type bloc `<condition name="X" value="Y">` : cette syntaxe permet de tester directement si le prédicat dont le nom est X a bien pour valeur Y, si oui, le code AIML contenu entre la balise fermante et la balise ouvrante sera évalué.

Exemple d'utilisation de plusieurs conditions de type "bloc" :

```
<category>
    <pattern>Je suis developpeur</pattern>
    <template>j ' aime
        <condition
            name="langage "
            value="java"> programmer en java !
        </condition>
        <condition
            name="langage "
            value="python"> pas trop le python
        </condition>
```

```
</template>
</category>
```

2. la condition de type prédicat unique `<condition name="X">` : cette syntaxe permet de tester les valeurs du prédicat dont le nom est X. Les valeurs à tester sont indiquées par une suite de balise ``, la première balise `` dont la valeur correspond est évaluée et renvoyée. La syntaxe complète de la balise condition donne donc ceci :

```
<condition name="X">
  <li value="1">...[AIML]... </li>
  <li value="2">...[AIML]... </li>
  <li value="3">...[AIML]... </li>
  ...
  <li value="99">...[AIML]... </li>
  <li>..... </li>
</condition>
```

La dernière balise `` contient la réponse par défaut, utilisée si aucune autre valeur ne correspond.

Exemple d'utilisation d'une balise condition à prédicat unique :

```
<category>
  <pattern>Je suis developpeur</pattern>
  <template>j 'aime
    <condition name="langage">
      <li value="java">programmer en java !.</li>
      <li value="python"> pas trop le python .</li>
      <li programmer en POO</li>
    </condition>
  </template>
</category>
```

Remarque : on peut donc utiliser l'une ou l'autre des syntaxes en fonction des besoins. Le deuxième exemple montre que cette syntaxe permet d'avoir une réponse par défaut, ce qui n'est pas le cas de la balise de type "bloc".

3. la condition de type prédicat multiple `<condition>` : cette syntaxe permet de tester la valeur de plusieurs prédicats. Les prédicats et les valeurs à tester sont indiqués par une suite de balises ``, la première balise `li` dont la valeur correspond est évaluée et renvoyée.

```
<condition name="X">
  <li name="X" value="1">...[AIML]... </li>
  <li name="Y" value="1">...[AIML]... </li>
```



```

<li name="Z" value="oui " >...[AIML]... </li>
<li name="Z" value="oui " >...[AIML]... </li>
<li name="Z" value="je ne sais pas " >...[AIML]... </li>
<li >..... </li>
<condition>

```

La dernière balise contient la réponse par défaut, utilisée si aucune autre valeur ne correspond.

Attention à la syntaxe des balises en fonction du type de condition utilisé. On peut placer du code avant et après une balise condition, et placer plusieurs conditions dans le même modèle de réponse. Par contre, on ne peut pas imbriquer les conditions.

Exemple d'utilisation d'une balise condition à prédicats multiples :

```

<category>
  <pattern>Je suis developpeur</pattern>
  <template> J'aime
    <condition>
      <li name="langage" value="java"> programmer en java !.</li>
      <li name="person" value="developpeur"> les developpeurs</li>
      <li name="langage" value="python"> pas le python.</li>
      <li > programmer en POO</li>
    </condition>
  </template>
</category>

```

Remarque : avec le langage AIML, il est impossible de tester si un prédicat est égal à NULL. Ce qui signifie des surprises quand on teste des variables non définies. La solution préconisée est de prévoir une valeur par défaut pour chaque prédicat utilisé, mais tout les interpréteurs ne le permettent pas. Ce dernier exemple montre que cette syntaxe permet de tester plusieurs valeurs de prédicats différents pour choisir la bonne réponse. L'ordre des est important. Le premier rencontré dont la valeur correspond est évaluée et retournée, et pas les suivants. Dans ce cas, le test "categorie=enfant" ne serait pas effectué systématiquement s'il n'était pas placé en première position. Pour les mêmes raisons, la balise contenant la réponse par défaut doit se situer à la fin de la liste.

sentence :

Cette balise sert à passer en majuscule (en utilisant la table de correspondance majuscule/minuscule Unicode) la première lettre du premier mot de chaque phrase.

Exemple :

```

<category>
  <pattern>Bonjour Monsieur</pattern>
  <template>
    <sentence>bonjour , ..... </sentence>
  </template>
</category>

```

Produit cette réponse :Bonjour ,

set :Permet de stocker une variable.

get :Permet de récupérer une variable définie dans une balise <set>.

3.6.2 ChatterBot

Définition :

ChatterBot est une librairie python⁴ open source, qui facilite la génération des réponses automatiques aux entrées de l'utilisateur.

ChatterBot utilise plusieurs algorithmes IA⁵ pour la sélection des réponses. Cette librairie permet l'apprentissage afin d'enrichir la base de connaissance du chatBot[?] . Le point fort de cette librairie est le support des sessions . L'utilisation de cette librairie impose la création d'une instance de l'objet ChatterBot.

L'objet ChatterBot :**les paramètres d'objet [?] :**

name (str)⁶ - Le nom de chatbot ,(obligatoire).

storage_adapter (str) - Le chemin d'accès à l'adaptateur de stockage.

logic_adapters (list) - Liste des chemins d'accès aux adaptateurs logiques.

input_adapter (str) - Le chemin d'accès à l'input adapter.

output_adapter (str) - d'accès à l'output adapter.

trainer (str) - chemin d'accès au trainer.

filters (list) - Une liste des le filtres .

4. Python : langage de programmation

5. IA : intelligence artificiel

6. str : String - chaine de caractères

Les méthodes d'instance :

`classmethod_from_config(config_file_path)`

Crée une nouvelle instance ChatBot à partir d'un fichier de configuration JSON⁷.

`generate_response(input_statement, session_id)`

Renvoie une réponse en fonction d'une déclaration d'entrée donnée.

`get_response(input_item, session_id=None)`

Renvoie la réponse du chatBot en fonction de l'entrée.

`initialize()`

Cette méthode est lancée automatiquement afin d'initialiser l'objet.

`learn_response(statement, previous_statement)`

Ajoute une réponse à une question déjà fournie

`set_trainer(training_class, **kwargs)`

Met le trainer en paramètre d'entrée par défaut .

`train()`

Lance l'apprentissage .

Adapters :

Types d'adapteurs :

— Storage adapters : génère une interface pour utiliser l'un des systèmes de stockage (MongoDB⁸, fichier local)

1. Json File Storage Adapter :

Cette adapteur permet a chatterBot de sauvegarder les conversations dans un fichier JSON.

2. Mongo Database Adapter :

Une interface qui permet à ChatterBot de sauvegarder les déclarations dans MongoDB.

— Input adapters : Permet de sélectionner la source des données (terminal, url, ...)

1. Terminal input adapter :

Permet à un utilisateur de taper dans son terminal pour communiquer avec le chatBot.

7. JSON : JavaScript Object Notation

8. MongoDB : est une base de données de type No-SQL

2. Gitter input adapter :
Permet à une instance ChatterBot d'obtenir des relevés d'entrée depuis une Gitter.
 3. HipChat input adapter :
Il s'agit d'un adaptateur de saisie qui permet à une instance ChatterBot de communiquer via HipChat.
 4. Mailgun input adapter :
permet à chatterBot de recevoir des courriels à l'aide de l'API Mailgun⁹
 5. Microsot Bot Framework input adapter :
Il s'agit d'un adaptateur de saisie qui permet à une instance de ChatterBot de communiquer via Microsoft en utilisant un protocole de client de ligne directe.
- Output adapters : retourne une réponse pour une source de données définie
1. Terminal output adapter :
Un adaptateur simple de sortie qui permet à une instance ChatterBot de communiquer via le terminal.
 2. Gitter output adapter :
Un adaptateur de sortie qui permet à une instance ChatterBot d'envoyer des réponses à une salle Gitter.
 3. HipChat output adapter :
Un adaptateur de sortie qui permet à une instance ChatterBot d'envoyer des réponses à une salle HipChat.
 4. Mailgun output adapter :
L'adaptateur de carte de correspondance permet à une instance d'envoyer des courriels à l'aide de l'API Mailgun.
 5. Microsot Bot Framework output adapter :
Un adaptateur de sortie qui permet à une instance de ChatterBot d'envoyer des réponses à un Microsoft en utilisant le protocole Direct Line.
1. Logic adapters :Le cœur de la librairie se situe dans le choix et/ou l'implémentation de ses Logic Adapters. Ils définissent la logique que suit le chatBot pour donner une réponse à l'entrée donnée par l'utilisateur et défini le logique à utiliser pour répondre , chaque Logic Adapter renvoie un couple (réponse, taux de confiance). Plusieurs Logic Adapters sont pré-implémentés par la librairie. Par exemple :
 - (a) Le TimeLogicAdapter :
Identifie les instructions dans lesquelles une question sur l'heure actuelle est demandée .

9. API Mailgun : API de mailing pour développeurs

- (b) Le `MathematicalEvaluationAdapter` :
- Détermine si l'utilisateur pose une question nécessitant des calculs mathématiques si c'est le cas l'évaluation mathématique passe par un ensemble des étapes pour extraire l'équation qui doit être résolue :
- Normalisation c'est-à-dire suppression des accents ainsi que certains caractères spéciaux .
 - Conversion des mots en nombres.
 - Extraction de l'équation.
 - Simplification de l'équation.
 - Résolution de l'équation.
- (c) Le `LowConfidenceResponseAdapter` :
- Renvoie une réponse par défaut spécifique si aucune réponse ne peut être déterminée avec grande valeur d'indice de confiance (calculé avec avec une méthode de sélection)??.
- (d) Le `MultiLogicAdapter` :
- Il est possible d'utiliser plusieurs `LogiqueAdapters`, le chatBot va retourner la réponse qui a le plus grand indice de confiance(Pourcentage de similitude).
- (e) Le `SpecificResponseAdapter` :
- Retourne une réponse spécifiée d'une question spécifiée
- (f) Le `BestMatchAdapter` :
- Un adapter logique qui renvoie une réponse en fonction des réponses connues aux correspondances les plus proches de l'instruction de saisie .

Méthode de sélection :

La librairie `ChatterBot` implémente par défaut des algorithmes afin de détecter la similarité entre deux textes afin de bien matcher l'entrée de l'utilisateur avec sa base de connaissance. Tout les algorithmes implémentés retournent un indice de confiance qui varie de 0.0 à 1.0 qui représente la probabilité de la similarité entre deux textes(l'entrée de l'utilisateur, une des questions de la base de connaissance du chatBot).

Distance Levenshtein :[?]

La distance de Levenshtein (LD) est un algorithme conçu par le scientifique russe Vladimir Levenshtein en 1965.

LD est le nombre minimal de remplacements, ajouts et suppressions de lettres pour passer du mot A au mot B.

LD est une mesure de la similitude entre deux chaînes, . La distance est le nombre de suppressions, d'insertions ou de substitutions nécessaires pour transformer pour passer du mot A au mot B.

- soit deux chaînes A : "test" et B : "test", alors $LD(A,B) = 0$, car pas besoin de d'une transformation

- soit deux chaînes A : "test" et B : "tent", alors $LD(A,B) = 1$, car un seul changement est suffisant pour transformer A en B (A devient B)

Exemple :

- Les deux mots étudiés sont écrits (en noir foncé) dans la colonne de gauche et sur la première ligne du tableau Figure ??.
- Ensuite les entiers 0, 1, 2 ... sont écrits (en vert) dans la seconde colonne et sur la seconde ligne Figure ??.

Chaque élément (écrit en noir ou en bleu) de la matrice est le plus petit des trois nombres suivants :

- 1 + le nombre à sa gauche
- 1 + le nombre au-dessus de lui
- le nombre en haut à gauche (en diagonale) +

- 0 si les lettres à gauche et en haut sont identiques
- 1 si ces lettres sont différentes
- La distance entre les deux mots est calculée par l'algorithme, se trouvera dans la case en bas, à droite (en rouge) Figure ??.

		C	O	N	D	O	R
	0	1	2	3	4	5	6
C	1						
O	2						
M	3						
D	4						
O	5						
R	6						

FIGURE 3.1 – DL Étape 1

		C	O	N	D	O	R
	0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
O	2	1	0	1	2	3	4
M	3	2	1	1	2	3	4
D	4	3	2	2	1	2	3
O	5	4	3	3	1	1	2
R	6	5	4	4	3	2	1

FIGURE 3.2 – DL Étape 2

Après avoir rempli la matrice et que la distance est connu on pourrai savoir quelle sont les opérations élémentaires (suppression ,ajout , modification) en parcourant la

matrice depuis la case de l'indice $M(i,j)$ en suivant la plus petite valeur des trois nombres $M(i-1,j)$, $M(i-1,j-1)$, $M(i,j-1)$ si la plus petite valeur se trouve a gauche c'est une suppression ,si en haut de la case c'est un ajout , si par la diagonal c'est une modification et on continue toujours par la case qu'a la petite valeur jusqu'à $M(0,0)$,si la valeur ne change pas on applique le même algorithme sur la case suivante qui contient la même valeur que l'indice en rouge Figure ??.

		C	O	N	D	O	R
	0	1	2	3	4	5	6
C	1	0	1	2	3	4	5
O	2	1	0	1	2	3	4
M	3	2	1	1	2	3	4
D	4	3	2	2	1	2	3
O	5	4	3	3	1	1	2
R	6	5	4	4	3	2	1

FIGURE 3.3 – DL Étape 3

Algorithme Jaccard :[?]

Un algorithme qui permet de détecter la similarité entre deux textes mais certainement pas le contenu dupliqué

Le principe est un des plus simples et beaucoup de personnes utilisent cet indice sans le savoir pour la détection de duplication.

L'indice de Jaccard appliqué à un texte correspond au rapport entre le nombre de mots présents dans les deux textes (l'intersection) et le nombre de mots des deux textes (l'union). L'indice de Jaccard est normalisé (entre 0 et 1), plus il est proche de 1 ou 100 plus les deux textes comparés sont similaires.

Avant d'appliquer la méthode il faut nettoyer les textes de tout qui pourrait nuire à la comparaison des textes. Pour cela on utilise une fonction qui supprime les accents, met les caractères en minuscule et remplace tout ce qui n'est pas une lettre par un espace.

Critique de l'indice de Jaccard

Utiliser un indice de Jaccard dans le but de détecter le contenu dupliqué ou de limiter la similarité de textes spinnés est une première étape. Malheureusement, il ne faut pas confondre similarité et duplication de contenus, si les moteurs de recherche

utilisaient un algorithme aussi simple beaucoup d'articles dans la même thématique seraient détectés en tant que duplique.

À la fin on peut dire que l'indice de Jaccard est un bon moyen de calculer la similarité entre deux textes mais qu'il n'est pas suffisant pour être utilisé dans une problématique de détection de contenant dupliqué.

WordNet :[?]

Est une grande base de données lexicales en anglais. Les noms, les verbes, les adjectifs et les adverbes sont regroupés en ensembles de synonymes cognitifs (synsets), chacun exprimant un concept distinct. Les syntaxes sont inter-connectées au moyen de relations conceptuelles sémantiques et lexicales. Le réseau résultant de mots et de concepts pertinents peut être navigué avec le navigateur. WordNet comprend les noms, les verbes, les adjectifs et les adverbes des catégories lexicales, mais ignore les prépositions, les déterminants et les autres mots fonctionnels.

Historique[?]

WordNet a été créé au Laboratoire de la science cognitive de l'Université de Princeton sous la direction du professeur de psychologie George Armitage Miller à partir de 1985 et dirigé ces dernières années par Christiane Fellbaum. Le projet a reçu des fonds d'organismes gouvernementaux, y compris la National Science Foundation, DARPA, le Bureau des technologies perturbatrices (anciennement l'activité de recherche avancée et de développement) et REFLEX.

Contenu de la base de données

WordNet contient plus de 118 000 formes de mots différents et plus de 90 000 sens de mot différents, ou plus de 166 000 (mot, sense) paires. Environ 17% des mots dans WordNet sont polysémiques; approximativement 40% possèdent un ou plusieurs synonymes.

Structure

La relation principale entre les mots dans WordNet est la synonymie. Les synonymes - mots qui désignent le même concept et sont interchangeable dans de nombreux contextes - sont regroupés en ensembles non ordonnés (syntaxe).

Chacun des 117 000 syntaxes de WordNet est lié à d'autres syntaxes au moyen d'un petit nombre de "relations conceptuelles". De plus, un synset contient une brève définition et, dans la plupart des cas, une ou plusieurs phrases courtes illustrant l'utilisation Des membres synset. Les formes de mots avec plusieurs significations distinctes sont représentées en autant de syntaxes distinctes. Ainsi, chaque paire form-meaning dans WordNet est unique.

Application

WordNet a été utilisé pour un certain nombre d'objectifs différents dans les systèmes d'information, y compris la désambiguïsation des mots, la récupération d'informations, la classification automatique des textes, la synthèse automatique des textes, la

traduction automatique et même la génération automatique de mots croisés.

Une utilisation courante de WordNet consiste à déterminer la similitude entre les mots.

Divers algorithmes ont été proposés, notamment la mesure de la distance entre les mots et les syntaïlles dans la structure graphique de WordNet, par exemple en comptant le nombre d'arêtes parmi les syntaxes.

L'intuition est que plus les deux mots ou syntaxes sont proches, plus leur sens est proches.

Principe de fonctionnement ChatterBot :

Par défaut une instance simple de ChatterBot commence avec aucune base de connaissance et un "Json File Storage Adapter" pour le stockage de la base de connaissance , "BestMatchAdapter" pour la sélection de la réponse à partir de la base de connaissance , "Terminal input adapter" pour la récupération d'entrée depuis terminal, "Terminal output adapter" pour l'affichage de la réponse sélectionné par le logic adapter. Chaque fois qu'un utilisateur entre un texte la librairie sauvegarde ce dernier dans la base de données sous la forme d'un objet Json avec le nombre d'occurrences de même pour la réponse. On peut aussi paramétrer la prise en compte ou non de stopwords : ce sont les mots très courant dans un langage donné qui vont finalement avoir peu d'influence sur le sens de la phrase. Exemple : en français, le, de, la, est peuvent être définis comme stopwords dans la phrase "le chat de la voisine est parti dehors".

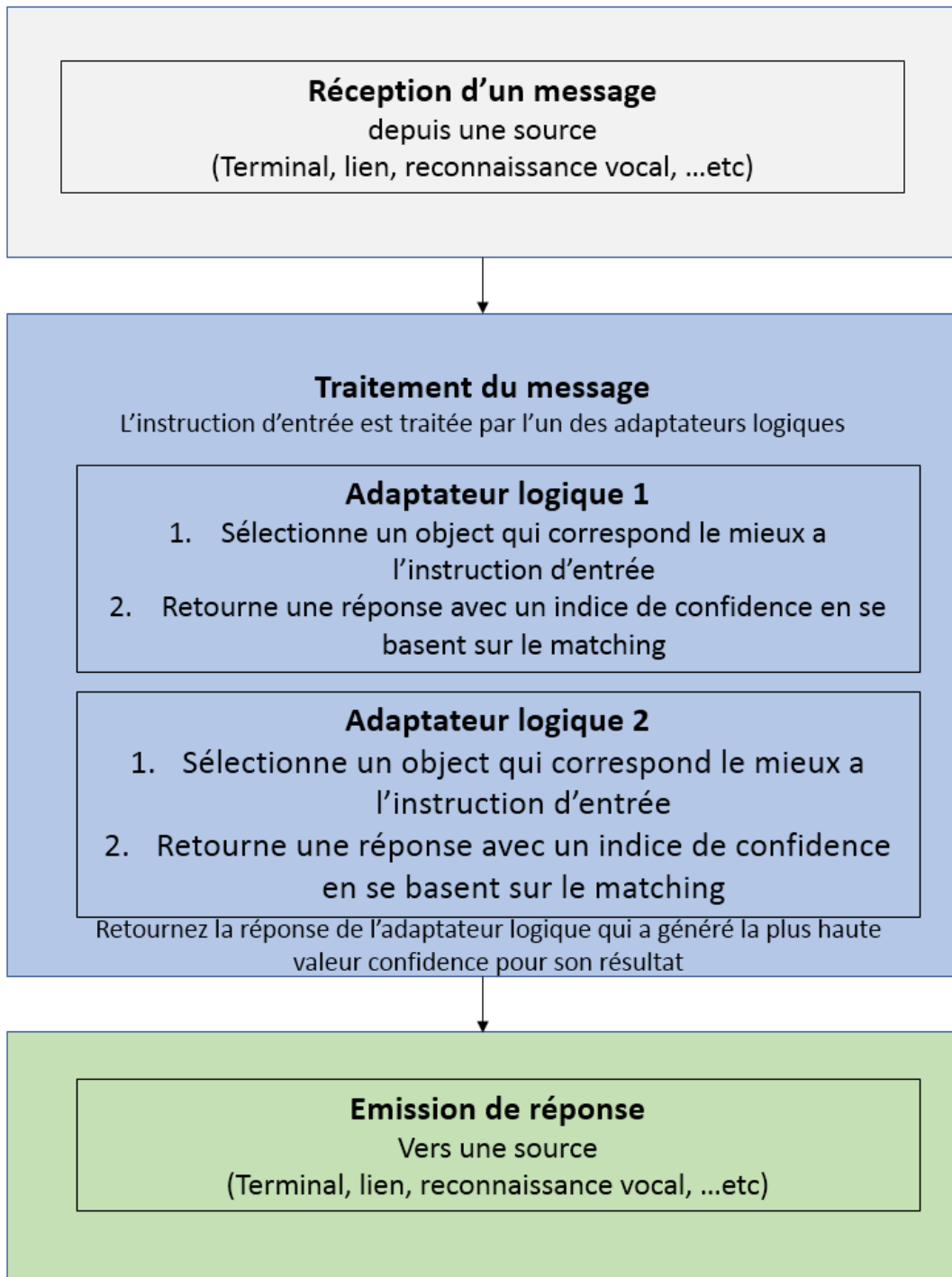


FIGURE 3.4 – Organigramme fonctionnement ChatterBot[?]

3.6.3 Réseaux de neurones

[?] Il y a certains problèmes qui ne peuvent être formulés sous forme d'un algorithme, des problèmes qui dépendent de nombreux facteurs subtils. Sans algorithme, un ordinateur se retrouve alors bloqué et ne peut pas résoudre ce genre de problématiques. Comment sommes-nous alors capable de répondre à ces problèmes et non les ordinateurs ? Nous apprenons, une capacité que ne possèdent évidemment pas les ordinateurs. Les êtres humains ont un cerveau, les ordinateurs ont quant à eux des unités de traitement et de la mémoire. L'ordinateur peut donc effectuer des calculs complexes dans un temps très court, mais n'est pas adapté à la résolution d'autres types de problématiques.

Historique :

[?]

Nous nous intéressons ici à une branche de l'Informatique fondamentale qui, sous l'appellation d'Intelligence Artificielle, a pour objectif de simuler des comportements du cerveau humain. Les premières tentatives de modélisation du cerveau sont anciennes et précèdent même l'ère informatique. C'est en 1943 que Mc Culloch (neurophysiologiste) et Pitts (logicien) ont proposé les premières notions de neurone formel.

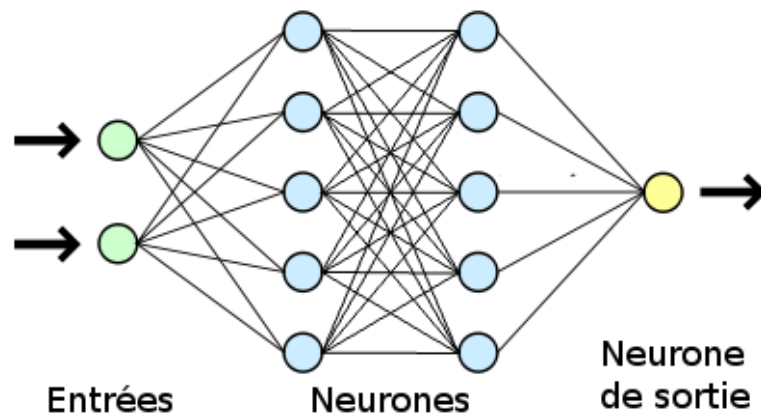


FIGURE 3.5 – Représentation d'un réseau de neurones [?]

Définition :

Un réseau de neurones artificiels est une technique de calcul qui se base sur le fonctionnement des neurones biologiques. Les neurones biologiques (ou neurones formels) opèrent mécaniquement à partir de règles précises.

Le principe du réseau de neurones est de reproduire schématiquement le cerveau humain et son mode d'association afin de l'automatiser dans une machine : c'est ce

qu'on appelle l'intelligence artificielle. Comme notre cerveau, le réseau de neurones débute par une phase d'apprentissage, avec l'association de données d'entrées (image, son, texte...) à des catégories spécifiques comme références.

Principe de fonctionnement :

Le principe du réseau de neurones est de créer un raisonnement statistique. En fonction de ce qu'il a appris par le passé, grâce à une « base » d'apprentissage, il va prendre des décisions par rapport aux données d'entrées qu'il reçoit. Le choix de classification se portera sur la probabilité de ressemblance à une classe qu'il connaît déjà. On appelle aussi ceci l'apprentissage par expérience, le réseau de neurones intervenant comme une aide statistique à la décision. Principe étape par étape :

* données en entrées (flèches noires à gauche sur le graphique),

* analyse des données avec des critères de pondération (cercles et liaisons sur le graphique),

* choix de classification en sortie avec une probabilité de certitude. (flèche noire à droite sur le graphique)

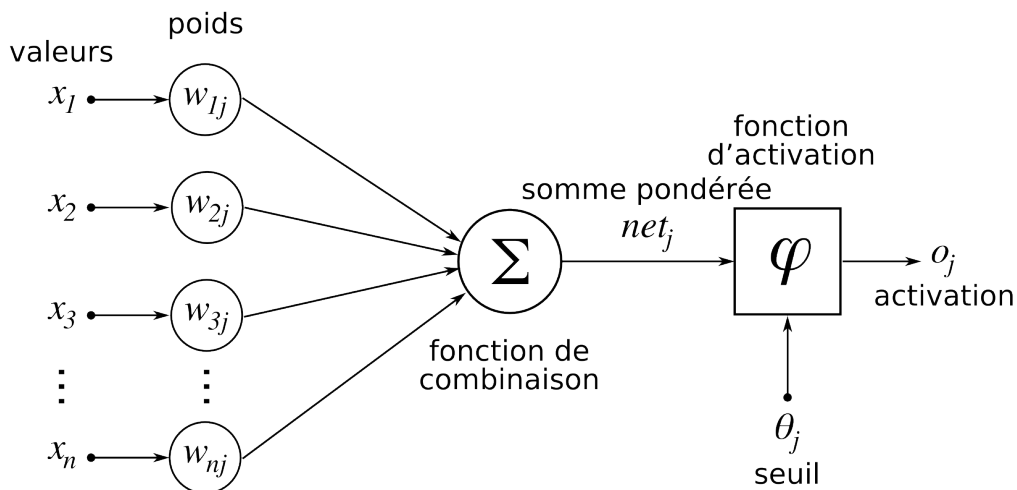


FIGURE 3.6 – Représentation d'un réseau de neurones

Un réseau de neurones possède deux états, un état d'apprentissage et un état de fonctionnement optimal. Afin de former le réseau, il doit y avoir une période d'enseignement et de répétition, tout comme un enfant qui forme son cerveau grâce à l'enseignement de ces parents tout au long de son enfance. Les réseaux de neurones apprennent par un processus de rétro propagation. Cela implique la comparaison de la sortie du réseau de neurones avec la sortie pour laquelle il a été conçu. Cette différence sert à modifier les poids des connexions entre les neurones du réseau et à le rendre plus précis. Une fois le réseau suffisamment calibré, il atteint un niveau où il n'est plus nécessaire de le superviser. Par exemple, supposons que nous avons enseigné à un réseau de neurones à distinguer une table d'une chaise. Pour cela nous lui avons montré 25 chaises et 25 tables différentes en lui expliquant bien à quoi correspondait

chaque image. Il atteint alors un état autonome et est capable de distinguer par lui-même si les nouvelles entrées correspondent plutôt à une table ou à une chaise. Plus fort encore, si on lui montre une variante, une chaise longue pour notre exemple. Il va être capable de catégoriser ce meuble en chaise et non en table. Simplement sur la seule base de son expérience et sans aucune aide, tout comme un être humain. . .

Apprentissage :

Mode supervisé

Un apprentissage est dit supervisé lorsque le réseau est forcé à converger vers un état final précis, en même temps qu'un motif lui est présenté.

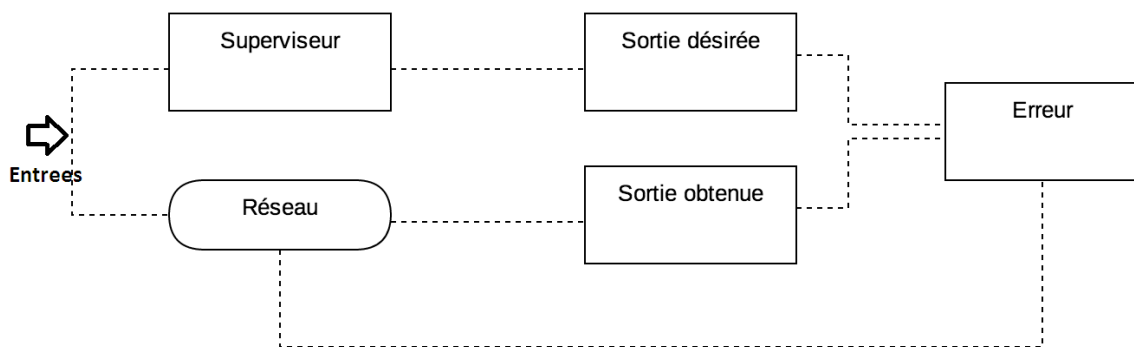


FIGURE 3.7 – Mode supervisé

Mode non-supervisé

Un apprentissage non-supervisé, le réseau est laissé libre de converger vers n'importe quel état final lorsqu'un motif lui est présenté.

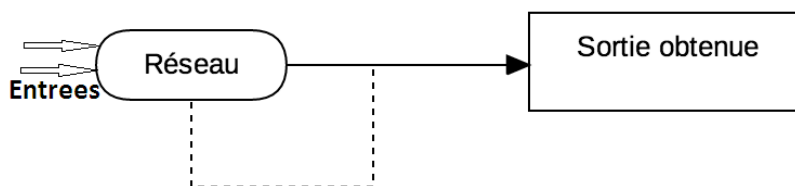


FIGURE 3.8 – Mode non supervisé

Surapprentissage

Il arrive souvent que les exemples de la base d'apprentissage comportent des valeurs approximatives ou bruitées. Si on oblige le réseau à répondre de façon quasi parfaite

relativement à ces exemples, on peut obtenir un réseau qui est biaisé par des valeurs erronées.

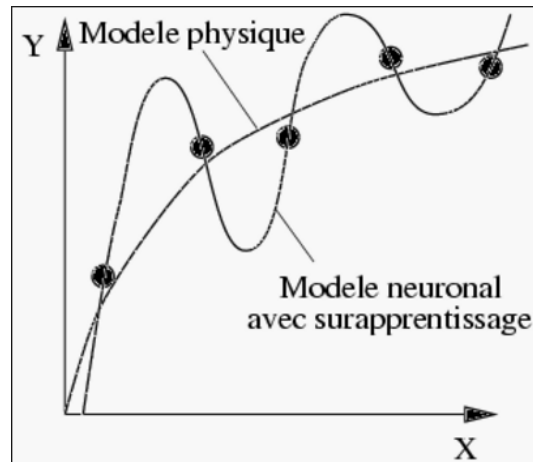


FIGURE 3.9 – Mode surapprentissage

Rétropropagation

La rétropropagation consiste à rétropropager l'erreur commise par un neurone à ses synapses et aux neurones qui y sont reliés

Élagage

L'élagage est une méthode qui permet d'éviter le surapprentissage tout en limitant la complexité du modèle. Elle consiste à supprimer des connexions (ou synapses), des entrées ou des neurones du réseau une fois l'apprentissage terminé. En pratique, les éléments qui ont la plus petite influence sur l'erreur de sortie du réseau sont supprimés. Les deux algorithmes d'élagage les plus utilisés sont :

- * Optimal brain damage (OBD) de Yann LeCun et al.
- * Optimal brain surgeon (OBS) de B. Hassibi et D. G. Stork.

3.7 Avantages et Inconvients

AIML , ChatterBot et réseau de neurones possèdent chacun des avantages et des inconvients.

3.7.1 AIML

Il est très difficile de maintenir un chatBot crée avec AIML. Le gros problème de l'AIML en plus d'une syntaxe XML particulièrement lourde, est l'énorme quantité d'instructions qu'il faut réaliser pour obtenir toutes les réponses possibles.

Les chatBots AIML ne maintiennent pas leur état cela signifiait qu' il répondront indépendamment à chaque question mais grâce une ontologie, nommée WOLF¹⁰. Cette base de connaissances en langage naturel se concentre surtout sur l'analyse grammaticale, mais possède aussi un dictionnaire de concepts très poussé .

3.7.2 ChatterBot

Avec ces multiples Adapters ,Cette librairie offre une base forte pour un chatBot maintenable. Chatterbot a tout de même ses limites car avec plusieurs scénarios de conversation en langage naturel comme « base d'apprentissage » ,Le chatBot prendra un peu plus de temps pour répondre mais la librairie offre la possibilité de changer le logic adapter afin de réduire le temps de réponse et implémenté d'autres algorithmes de sélection . Grâce a cette librairie ,Un chatBot peut comprendre les phrases comportant des fautes d'orthographe ou de syntaxe.

3.7.3 Réseau de neurones

Plus le problème est complexe, plus il nécessite une base de données importante pour la « base d'apprentissage » du chatBot ,additionnellement le réseau de neurones est coûteux en temps , en argent et la détermination des descripteurs nécessite une analyse réalisée par un expert. Selon les descripteurs qui en sont extraits, peut engendrer des réseaux de neurones aux performances très différentes. Chaque changement dans la « base d'apprentissage » nécessite un nouveau descripteur (coûteux en temps et en argent) ,additionnellement un réseau de neurones avec une petite « base d'apprentissage » serait incapable de répondre parfaitement.

10. WOLF : Une ressource lexicale sémantique (wordnet) libre pour le français.

3.8 Test de Turing

Le test de Turing est un test qui a été mis en place pour essayer de définir si une intelligence artificielle est capable de se comporter comme un humain et permettant d'imiter le comportement humain.

Le test consiste à confronter un humain face à deux interlocuteurs, une intelligence artificielle et un humain. S'il n'est pas capable d'identifier l'humain de la machine alors le test est validé pour le chatBot. Ce fameux test de Turing qui définit une intelligence artificielle comme une intelligence indistinctement de celle de l'homme dans de telles discussions par le biais d'échanges textuels[?].

Tout les chatBots crée avec AIML , ChatterBot ou réseaux de neurones ont passé le test du turing avec succès .

3.9 Conclusion

Dans ce chapitre nous avons commencer par définir un chatBot ensuite on avons présenté les différents types et familles des chatBot ainsi que l'utilité de ce dernier dans une entreprise et enfin nous avons illustré les différentes techniques existantes qui permettent la création d'un chatBot, dans ce projet on a utilisé la librairie ChatterBot on apportant des modifications et nous avons utilisé facebook comme interface d'interaction que nous allons détaillé dans le chapitre suivant.

Chapitre 4

API Facebook

4.1 Introduction

Parmi les sujets technologie tendances, les bots ,qui font beaucoup parler d'eux depuis cette année. Ce sujet est devenu à la mode et ce qui a poussé Facebook a passé à la vitesse supérieure dans ses efforts pour intégrer d'autres services à son application de messagerie Messenger . Afin qu'un chatBot puisse entretenir une conversation avec un humain, Facebook à met en place un Api que nous allons découvrir dans ce chapitre .

4.2 Qu'est ce que Facebook ?

[?]

Créé par Mark Elliot Zuckerberg, Facebook est à l'origine destiné aux étudiants de l'université d'Harvard . Devant son succès, le site s'est ouvert en septembre 2006 à tous les internautes. Le nom « Facebook » qui signifie « trombinoscope », vient des photos de classe distribuées en fin d'année scolaire aux étudiants. Le site était donc conçu au départ comme un immense trombinoscope virtuel présentant chaque élève. Facebook est un réseau social très populaire qui vous permet d'être en lien avec des amis, partager des centres d'intérêts et rejoindre des groupes. Facebook est le plus grand site de réseautage social au monde avec plus de 600 millions d'utilisateurs. L'utilisateur interagit avec en moyenne environ 130 amis sur leur page.

4.3 Facebook, pourquoi faire ?

Facebook change la façon dont nous interagissons sur le web. Les possibilités offertes par le réseau social sont très larges :

- Lier les amis à travers le globe
- Partager des photos, des vidéos avec ses amis et sa famille
- Inviter des amis à un événement

- Dialoguer avec des amis et administrateurs de page par messageries instantanées
- Demander de l'aide par messageries instantanées ...
- Poser des questions aux pages des sociétés ou associations par messageries instantanées ...

4.4 Facebook Messenger Platform

Les 12 et 13 avril 2016 se tenait la conférence F8¹ de Facebook[?], lors de cette dernière Facebook a annoncé l'API Send/Receive pour Messenger. L'API Send/Receive permettra de relier des chatbots à des pages ou profil sur Facebook.

L'API Send/Receive ne supportait pas que les envois et les réceptions des messages textes, mais également des photos et des bulles interactives contenant de multiples actions contextuelles ainsi que des listes . « Les développeurs peuvent également définir un écran d'accueil pour leurs discussions pour définir le contexte ainsi que les différents contrôles. À compter d'aujourd'hui, les développeurs et les entreprises auront accès aux documents pour concevoir des bots pour Messenger et les soumettre pour être examinés. Nous allons progressivement accepter et approuver les soumissions afin d'assurer les meilleures expériences pour tout le monde sur Messenger ».

4.5 Principe de fonctionnement

Avant de pouvoir relier une page avec un chatBot, Facebook exige certaines étapes afin de garantir que les interlocuteurs seront en sécurité et auront une bonne expérience :

4.5.1 Mise en place :[?]

- Se rendre sur <https://developers.facebook.com> .
- Sélectionnez la page créée précédemment et créer une application Facebook afin de générer un PAGE-ACCESS-TOKEN².
- Configurer les webhooks³ juste en inscrivent le lien de rappel en HTTPS⁴ plus un mot de passe .

1. F8 : Conférence mondiale annuelle des développeurs

2. Ce jeton sera utilisé pour authentifier les demandes à chaque envoi de message

3. webhooks :Servent notifier des scripts externes à une application web qu'un événement a eu lieu

4. HTTPS : HyperText Transefert Protocole Sécurisé

- Facebook lance un défi similaire à celui d'Hell-man⁵ par la suscription d'un mot de passe sur la page et sur le script ou lien de rappel pointe, après la validation ; Facebook sollicite le script par la méthode HTTP GET afin de vérifier la similarité du mot de passe inscrit dans la page et sur le script.
- Lors de la validation Facebook envoi par la méthode HTTP GET (GET est la méthode permettant de récupérer le contenu d'un fichier) le mot de passe vers le lien de rappel, si le même mot de passe est sur le script, il doit retourner un ok, sinon une erreur.

4.5.2 Fonctionnement :

Après avoir validé le mot de passe et le lien de rappel sur le tableau de bord , Chaque message reçu sur la page est re-directionné vers le lien de rappel par méthode HTTP POST(Il a le même effet que GET sauf que, les données sont envoyés via le corps de la requête) en un texte structuré en Json.

Exemple :

```
{ "object": "page",
  "entry":
  [ {
    "id": "1348801641866967",
    "time": 1494552858618,
    "messaging":
      [
        {
          "sender": { "id": "1547809251959501" },
          "recipient": { "id": "1348801641866967" },
          "timestamp": 1494552858212,
          "message":
            {
              "quick_reply": { "payload": "BOT_OFF_PAYLOAD" },
              "mid": "mid.$cAASQTr0t_AZiK5AWZFB-krOAK3rQ",
              "seq": 742398,
              "text": "Oui"
            }
          }
        ]
      }
    ]
  }
}
```

5. Hell-man :est une méthode de cryptographie

- Dans cet exemple nous pouvons observer qu'il y a 2 membres (object, entry). object qui détermine la provenance du message et entry est constitué d'un seul membre qui est un tableau d'un seul objet qui contient 3 membres . le plus important dans ce json est messaging dans entry[0], il permettra de savoir qui a envoyé et quand il a envoyé et qu'il a envoyé .

pour répondre c'est simple il suffit d'envoyer par la méthode HTTP POST un text structuré en Json à `https://graph.facebook.com/v2.6/me/messages?access_token=<PAGE_ACCESS_TOKEN>` .

4.6 Types de contenu

Facebook a met un format texte avec une certaine structure Json pour tous les types de contenu brut (non structuré) qui peuvent être envoyés aux utilisateurs tel que texte , fichier , vidéo ou audio et aussi une structure Json pour le contenu structuré comme les boutons ou les listes . Pour envoyer un message du type quelconque, une requête POST à `https://graph.facebook.com/v2.6/me/messages?`

`access_token=<PAGE_ACCESS_TOKEN>` doit être achevé avec les message en question.

4.6.1 Non structuré

Texte

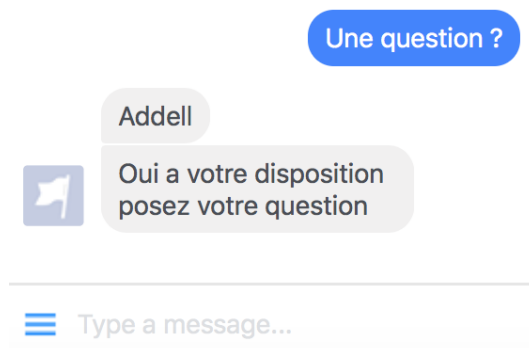


FIGURE 4.1 – Exemple Texte

```
{
  "recipient":{"id":"1547809251959501"},
  "message":{"text":"Oui a votre disposition posez votre question"}
}
```

- recipient{"id"} obligatoire.
- message{"text"} obligatoire et doit être au format UTF-8 et ne doit pas dépasser 640 caractères .

Fichier

FIGURE 4.2 – Exemple Fichier

```
{
  "recipient":{"id":"1547809251959501"},
  "message":
  {
    "attachment":
    {
      "type":"audio",
      "payload":{"url":"https://condorbot.com/c7_ring.mp3"}
    }
  }
}
```

- recipient{"id"} obligatoire.
- message{attachment{"type"}} il indique le type de donnée à envoyer, voici les valeurs qui peut prendre :
 - "audio" : pour les fichiers audio.
 - "file" : pour tous type des donnes .
 - "vidéo" : pour les vidéos.
- message{attachment{"payload"}} métadonnée indique le chemin complet ou se trouve ce fichier audio.

Sinon par upload par exemple une vidéo :

```
curl /
-F 'recipient={"id":"1547809251959501"}' /
-F 'message={"attachment":{"type":"video","payload":{}}}' /
-F 'filedata=@/tmp/clip.mp4;type=video/mp4' /
```

"https://graph.facebook.com/v2.6/me/messages?access_token=PAGE_ACCESS_TOKEN"
 – filedata est le chemin absolu ou se trouve la vidéo sur la machine .

4.6.2 Structuré

Bouton

Pour envoyer un texte et des boutons joints, voici un exemple :



FIGURE 4.3 – Exemple Boutons

```
{
  "recipient":{"id":"USER_ID"},
  "message":{"
    "attachment":{"
      "type":"template",
      "payload":{"
        "template_type":"button",
        "text":"Avec quel produit avez vous un probleme ?",
        "buttons":[
          {
            "type":"postback",
            "payload":"MOBILE_PAYLOAD",
            "title":"Smartphone"},
          {
            "type":"postback",
            "title":"Informatique",
            "payload":"INFO_PAYLOAD"
          }
        ]
      }
    }
  }
}
```

Le bouton est caractérisé par un titre "title" qui s'affiche dans la discussion et un type qui peut prendre que deux valeurs "postback" ou "web_url".

Si l'interlocuteur clique sur un bouton de type "postback" le titre de ce bouton sera renvoyé vers le lien de rappel avec un indice qui est le paramètre "payload" .

Si facebook envoie seulement le titre, le script qui interprètera le message il lui sera impossible de savoir si l'interlocuteur a écrit ou il a cliqué sur un bouton donc si un y a un clique une métadonnée "payload" est associée au message a envoyé sinon le message sera envoyé sans le paramètre "payload".

Si l'interlocuteur clique sur un bouton de type "web_url" , Messenger lui ouvrira directement le navigateur par défaut avec le paramètre "url" .

Liste

Pour envoyer une liste avec des éléments voici un exemple avec une image de couverture :

```
{
  "recipient": {"id": "RECIPIENT_ID"},
  "message": {
    "attachment": {
      "type": "template",
      "payload": {
        "template_type": "list",
        "elements": [
          {
            "title": "Classic T-Shirt Collection",
            "image_url": "https://.../img/collection.png",
            "subtitle": "See all our colors",
            "default_action": {
              "type": "web_url",
              "url": "https://..../",
              "messenger_extensions": true,
              "webview_height_ratio": "tall",
              "fallback_url": "https://..../"
            }
          },
          {
            "title": "View",
            "type": "web_url",
            "url": "https://.../",
            "messenger_extensions": true,
            "webview_height_ratio": "tall",
            "fallback_url": "https://..../"
          }
        ]
      }
    }
  }
}
```

```

    } , ..... ] ,
    "buttons": [
      {
        "title": "View More" ,
        "type": "postback" ,
        "payload": "payload"
      }
    ]
  }
}

```

- Le "type" et le "payload" sont obligatoires dans la liste .
- Pour chaque élément dans payload{"elements"} le "title" et le "subtitle" ne peuvent pas excéder les 80 caractères.
- Les buttons sont limités à 1 pour chaque élément de la liste.

4.7 Divers[?]

API Send/Receive ne se limite pas à offrir des méthodes d'échange pour tout types de contenu mais aussi pour améliorer l'expérience utilisateur, Facebook a ajouté des réponses rapides et un menu persistant.

4.7.1 Réponse rapide

Les réponses rapides fournissent un moyen de présenter des boutons à l'utilisateur en réponse à un message et apparaissent en évidence au-dessus du message. Une fois que l'utilisateur clique sur la réponse rapide cette dernière est envoyée dans la conversation mais aussi vers le lien de rappel avec les métadonnées définies .

voulez vous raiment me
desactiver ?

Oui

Non

FIGURE 4.4 – Exemple Réponses rapides


```

{
  "recipient":{"id":"USER_ID"},
  "message":{"
    "text":"Voulez vous desactiver le chatbot?",
    "quick_replies":[
      {
        "content_type":"text",
        "title":"Oui",
        "payload":"BOT_OFF_PAYLOAD"
      },
      {
        "content_type":"text",
        "title":"Non",
        "payload":"BOT_ON_PAYLOAD"
      }
    ]
  }
}

```

- "quick_replies" liste des réponses rapides .
- "content_type" peut prendre que deux valeurs "text" ou "location" (permet de récupérer la latitude, la longitude et l'altitude) .
- message{quick_replies{0}{payload}} métadonnée .

4.7.2 Menu persistant

Le menu Persistant apporte les utilisateurs directement dans les fonctions de haut niveau, ce dernier se trouve toujours en bas de la discussion il suffit de le customiser et l'activer par exemple :

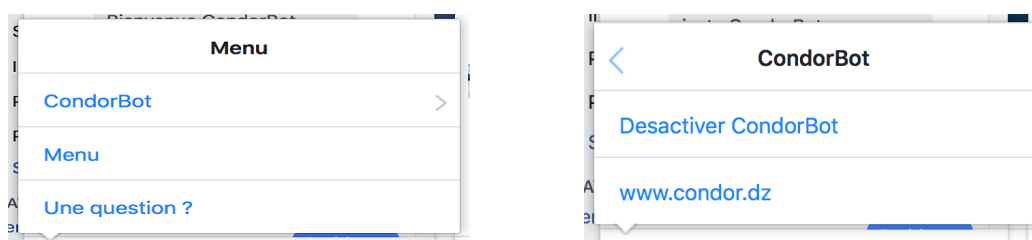


FIGURE 4.5 – Exemple Menu persistant

```

{
  "persistent_menu ":[
    {
      "locale ":" default " ,
      "composer_input_disabled ": false ,
      "call_to_actions ":[
        {
          "title ":" CondorBot " ,
          "type ":" nested " ,
          "call_to_actions ":[
            {
              "title ":" Desactiver CondorBot " ,
              "type ":" postback " ,
              "payload ":" BOT_OFF "
            } ,
            {
              "title ":" www.condor.dz " ,
              "type ":" web_url " ,
              "url ":" https://www.condor.dz /"
            }
          ]
        } ,
        {
          "type ":" postback " ,
          "title ":" Menu " ,
          "payload ":" condorbot "
        } ,
        {
          "type ":" postback " ,
          "title ":" Une question ? " ,
          "payload ":" QUESTION_PAYLOAD "
        }
      ]
    } ,
    {
      "locale ":" zh_CN " ,
      "composer_input_disabled ": false }
  ]
}

```

4.8 Conclusion

Dans ce chapitre nous avons vu comment relier le Messenger d'une page avec un lien de rappel, le principe de fonctionnement de Messenger et les méthodes de transférer des différents types du contenu. Dans le chapitre suivant nous allons concevoir notre chatBot afin qu'il puisse répondre aux besoins.

Chapitre 5

Conception et réalisation

La réalisation d'un système doit principalement passer d'une phase importante et indispensable qui est la conception. Cette phase a pour objectif d'expliquer le déroulement de notre système, de lister les résultats attendus en termes de fonctionnalités et d'assurer une bonne compréhension des besoins des utilisateurs. Avant de se pencher sur la conception de système, La première étape sera la présentation d'un cahier des charges afin de spécifier le besoin : que va faire le bot ? Quelle est son intérêt, quelles fonctions va-t-il remplir ? La seconde étape sera la conception du système toute en respectent le cahier des charges .

5.1 Méthodologie et approche adoptée :

La réalisation d'un système être principalement passée d'une phase importante et indispensable qui est la conception.

Il existe plusieurs langages de modélisation et de conception des systèmes d'informations. Pour mieux présenter l'architecture de notre système, nous avons choisi la modélisation objet avec le langage UML.

5.1.1 Définition d'UML

[?][?]

UML est l'abréviation de Unified Modeling Language, c'est-à-dire langage unifié pour la modélisation. C'est une notation graphique destinée à la création de modèles orientés objet. Ce n'est pas une méthode, c'est un ensemble d'outils permettant la modélisation de la future application informatique.

5.2 Cahier des charges

Le cahier des charges devra décrire précisément les opérations à prendre en charge, Voici les fonctionnalités attendues du projet :

- Création d'un chatBot Facebook prédéfini afin de ne pas reproduire la faute de Microsoft lors du lancement du chatBot auto apprenant "TayTwetts" devenu en 24 heures un Chatbot raciste par la manipulation des internautes.
- Création d'un chatBot de service pour des usages très précis, pour répondre à des questions simples et s'en tenir à une mission unique et renseigner les clients et les guider .
- En cas d'incapacité de réponse ou de dialogue, le chatBot doit orienter le client vers un conseiller.
- Pour chaque nouveau client sur Facebook le chatBot doit expliquer ce qu'il sait faire et ce qu'il peut apporter comme informations.
- Afficher des boutons, des listes afin de guider la conversation et de s'assurer que l'utilisateur reste dans le flux de la conversation.
- ChatBot doit gérer le contexte et mémorise des informations afin de ne pas les re-demander.
- Afficher des composants graphiques beaucoup plus parlants que du texte tel que les images .
- Mettre en place un système qui permet d'activer ou désactiver individuellement le chatBot , Cette action peut être déclenchée par le client ainsi que l'administrateur .
- L'administrateur aura la possibilité d'activer ou désactiver le chatBot pour tout les clients.
- Afficher un menu persistant comme celui de "The gardian".
- Mettre en devance d'autres liens de contact comme appel téléphonique, site web.
- Mettre en place un système qui permet de visualiser le chatBot avec tout les clients a qu'il parle et qui permet aussi l'intervention en temps réel.
- Garder un historique sur toutes les conversations .

5.3 Conception

5.3.1 introduction

Après avoir défini les problèmes et préciser les besoins auxquels notre projet doit répondre, nous arrivons à la phase de conception. Dans cette section, nous présentons quelques diagrammes de modélisation en explicitant leurs rôles. Nous allons principalement présenter les diagrammes qui définissent les principales fonctionnalités de notre système sans se préoccuper de comment il va le faire .

5.3.2 Présentation du projet :

Notre projet est de réaliser un système qui a pour objectif d'améliorer le service clientèle de Condor Electronics, ce système est construit par un chatBot intelligent, et une application de gestion et messagerie de ce dernier .

Ce diagramme montre tous les éléments qui interagissent entre eux afin de garantir la mise en marche de notre système.

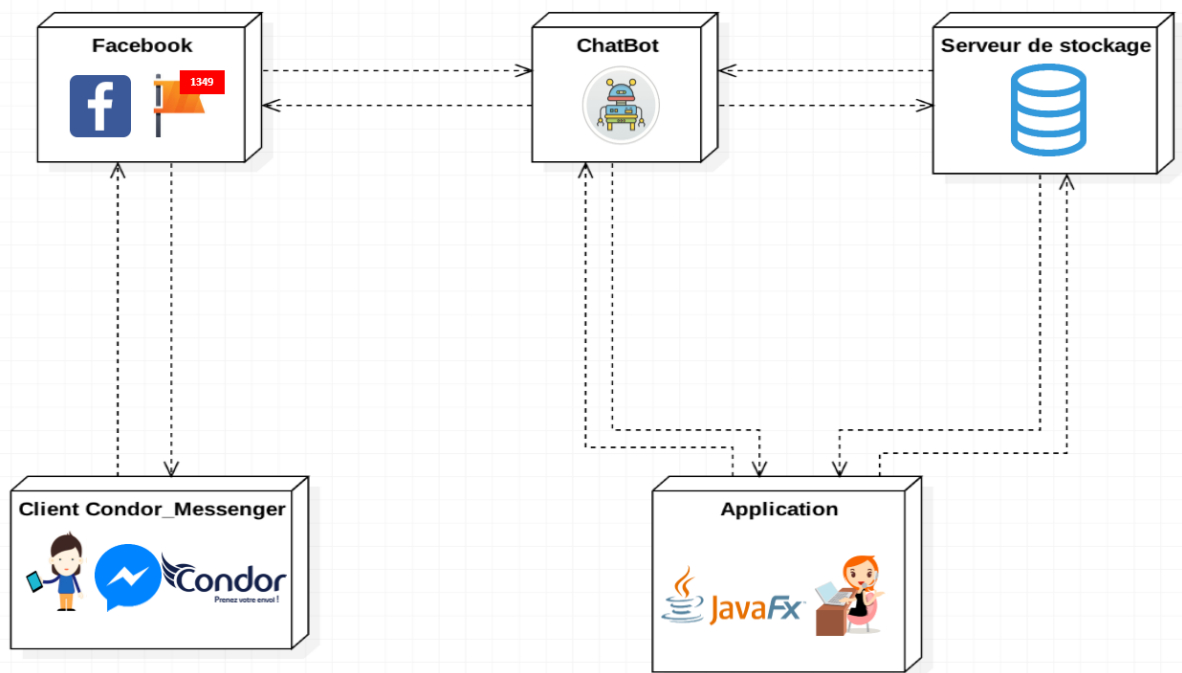


FIGURE 5.1 – Diagramme de déploiement

Présentation des diagrammes :**Diagramme de cas d'utilisation :**

Les diagrammes de cas d'utilisation sont des diagrammes utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet.

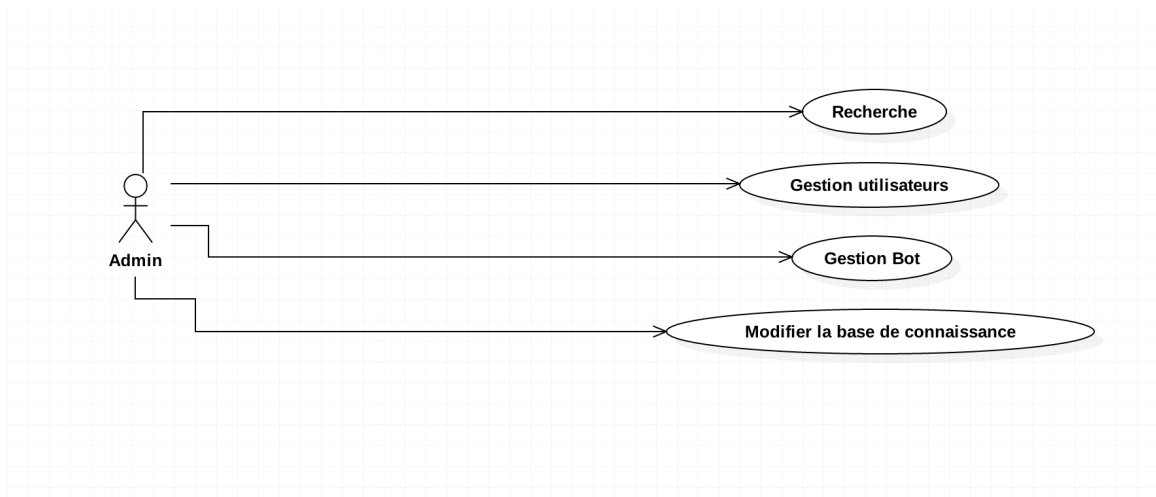


FIGURE 5.2 – Diagramme de cas d'utilisation

Accessibilité aux pages (Confidentialité)

L'application doit être ouverte aux administrateurs de service pour gérer la base de connaissance du chatBot et gérer les messages reçus. Les administrateurs effectuent aussi des opérations de contrôle sur le chatBot tel que activation ou désactivation individuel ou global en cas de besoin.

Diagramme de séquence :

Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique. Le diagramme de séquence permet de montrer les interactions d'objets dans le cadre d'un scénario prédit.

— **Authentification**

Procédure consistant à vérifier ou à valider l'identité d'une personne ou l'identification de toute autre entité, pour contrôler l'accès à l'application.

Acteur : Tous les administrateurs.

Pré-condition : Aucune.

Post condition : l'administrateur est identifié et accède .

Scénario Principal :

- L'administrateur ouvre l'interface.
- L'administrateur saisit son nom d'utilisateur et son mot de passe.
- Le système affiche l'interface principale de l'application.

Exceptions :

Si les informations sont erronées, l'application affiche un message d'erreur.

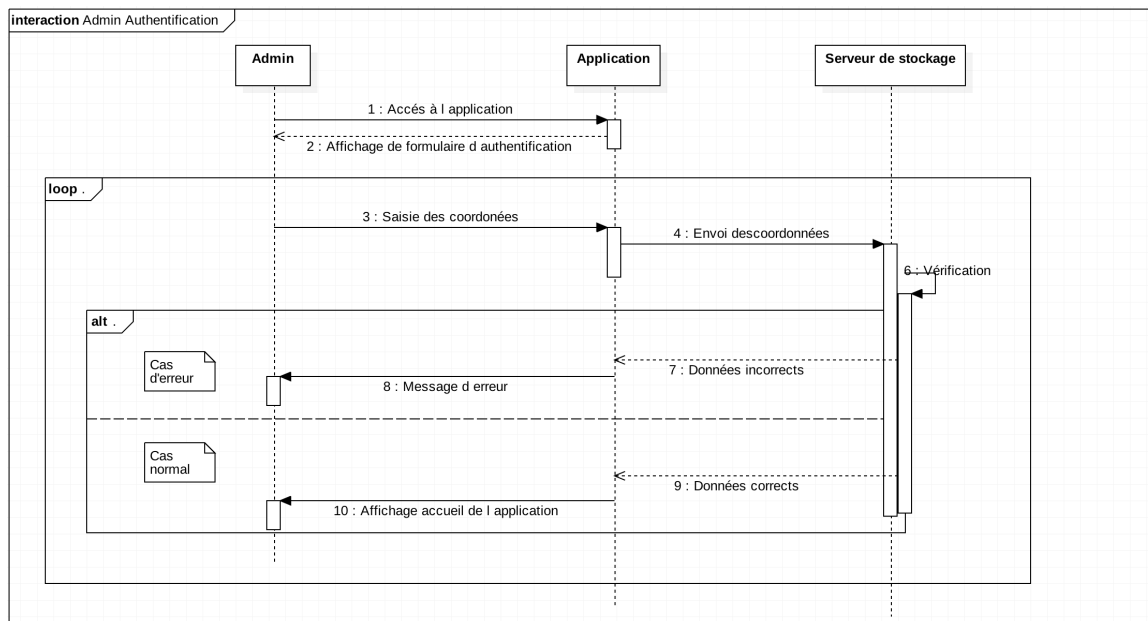


FIGURE 5.3 – Diagramme de séquence d'authentification

— **Recherche**

Cette procédure consiste à chercher dans l'application soit sur un profile ou une discussion .

Acteur :Les administrateurs.

Pré-condition :l'administrateur est connecté, liste complète récupérer

Scénario Principal :

- L'administrateur accède au formulaire de recherche.
- L'administrateur spécifie le mot à rechercher.
- Le système vérifie les informations saisies.
- Le système affiche le résultat de la recherche.

Exceptions :

En cas d'erreur de saisie,l'application affiche un message d'erreur. Si les informations n'existent pas,l'application affiche un message d'erreur.

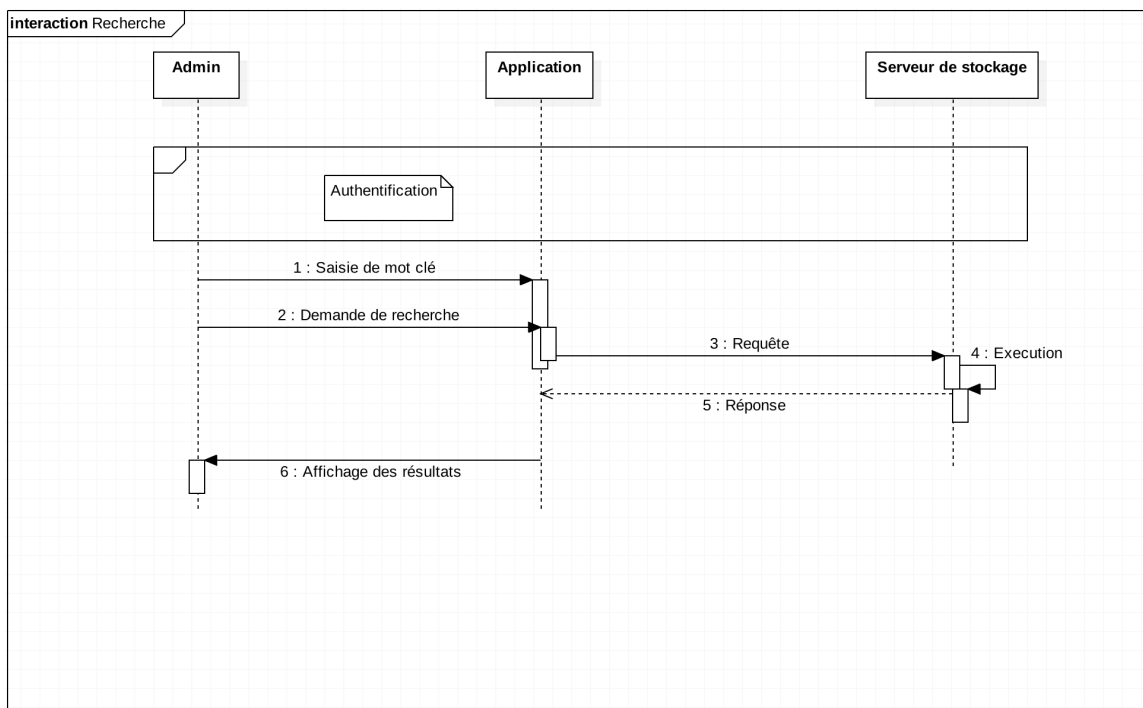


FIGURE 5.4 – Diagramme de séquence de recherche

— **Contrôle du chatBot**

Cette procédure consiste à désactiver ou activer individuellement le chatBot pour l'un des clients selon le besoin.

Acteur : Les administrateurs.

Pré-condition : l'administrateur est connecté, la liste complète des clients récupérer

Scénario Principal :

- L'administrateur accède à l'interface principale de l'application.
- L'administrateur spécifie le client .
- L'administrateur clique sur le bouton d'activation/désactivation.

Exceptions :

En cas d'erreur de connexion, l'application affiche un message d'erreur.

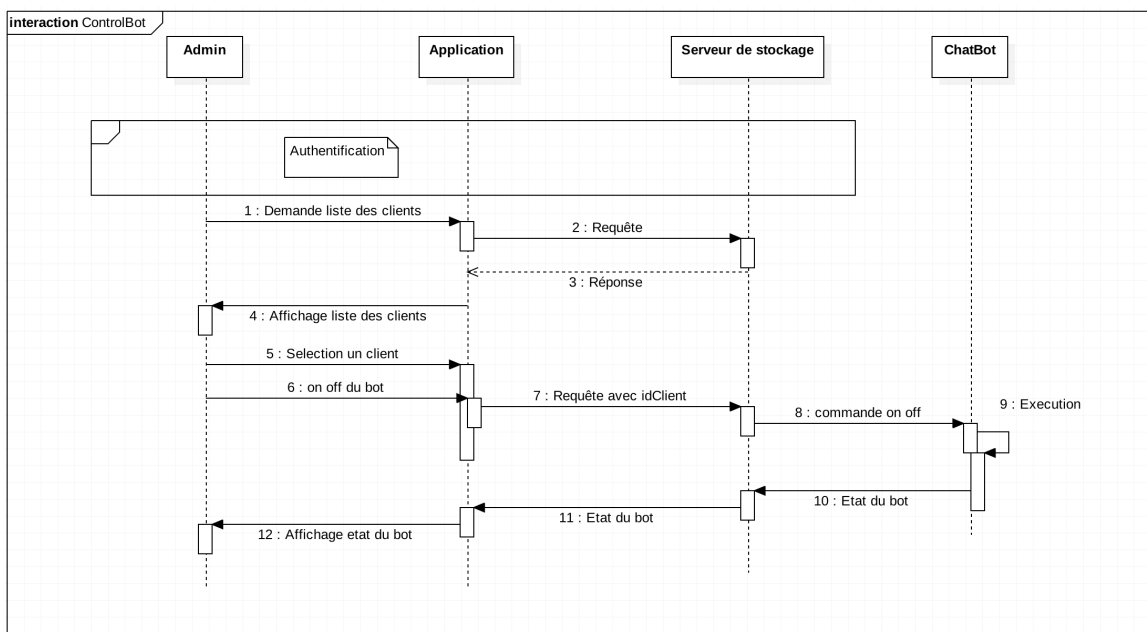


FIGURE 5.5 – Diagramme de séquence contrôle du chatBot

— Gestion du chatBot

Cette procédure consiste à gérer (ajouter , modifier , supprimer)la base de connaissance du chatBot.

Acteur :Les administrateurs.

Pré-condition :l'administrateur est connecté.

Scénario Principal :

Ajout :

- L'administrateur accède à l'interface de traitement.
- L'administrateur saisit les informations nécessaires et clique sur ajouter.
- L'application vérifie les propriétés à ajouter.
- L'application enregistre l'entré ajouté.
- Un message affirmant l'opération effectuée s'affiche.

Modification :

- L'administrateur accède à l'interface de traitement.
- L'administrateur cherche la donnée à modifier.
- L'administrateur saisit les informations nécessaires à modifier et clique sur modifier utilisateur
- L'application vérifie les propriétés à modifier.
- L'application enregistre les modifications..
- Un message affirmant l'opération effectuée s'affiche.

Suppression :

- L'administrateur accède à l'interface de traitement.
- L'administrateur cherche la donnée à supprimer.
- L'administrateur sélectionne les données à supprimer et clique sur supprimer.
- L'application vérifie les données à supprimer.
- L'application enregistre les modifications.
- Un message affirmant l'opération effectuée s'affiche.

Exceptions :

En cas d'erreur de connexion, l'application affiche un message d'erreur.

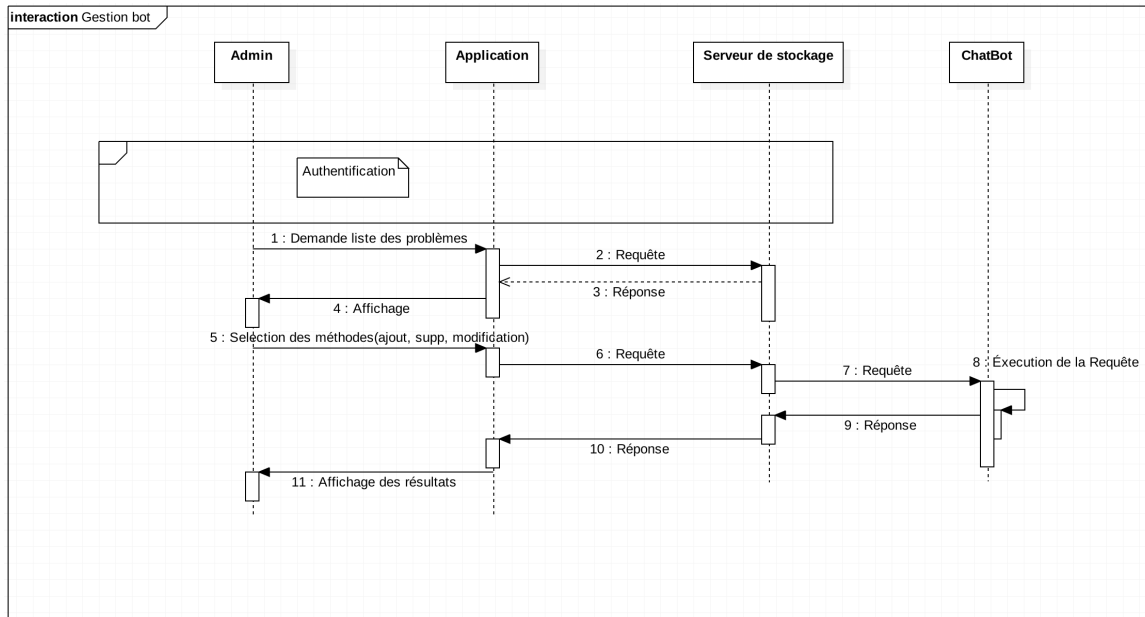


FIGURE 5.6 – Diagramme de séquence gestion du chatBot

— Gestion d'utilisateurs

Cette procédure consiste à gérer (ajouter , modifier , supprimer)les profile des client.

Acteur :Les administrateurs.

Pré-condition :l'administrateur est connecté.

Scénario Principal :

- L'administrateur accède à l'interface de traitement.
- L'administrateur sélection un client.
- L'administrateur applique une méthode.
- Un message affirmant l'opération effectuée s'affiche.

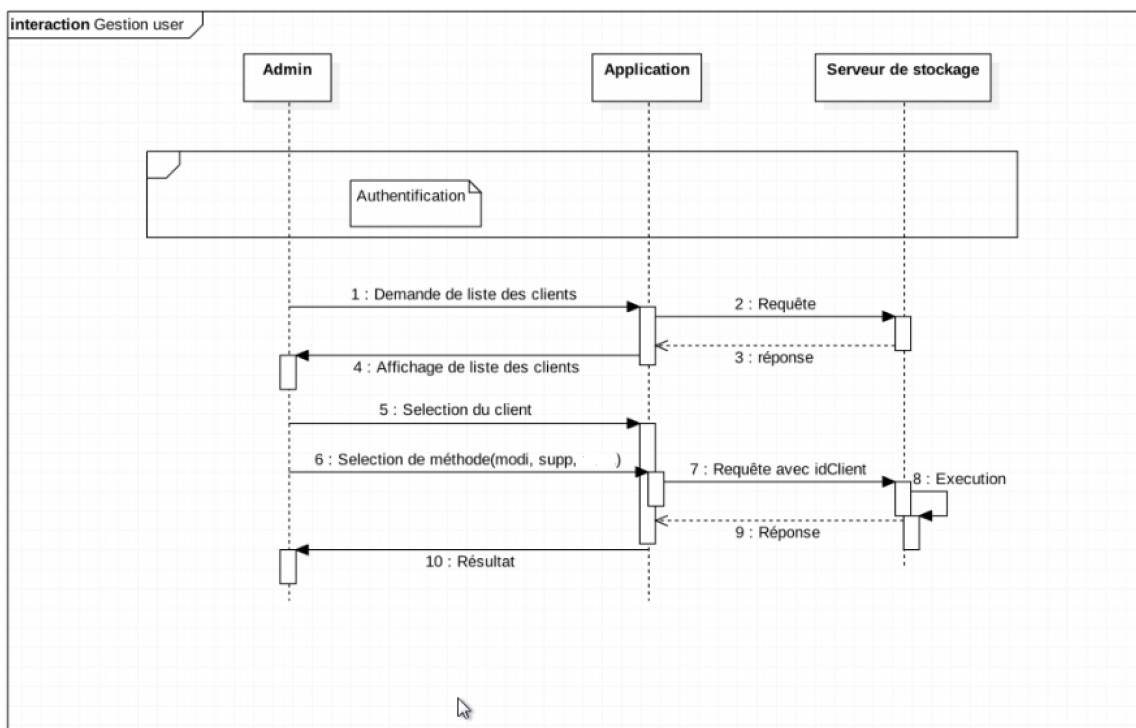


FIGURE 5.7 – Diagramme de gestion d'utilisateurs

— **Gestion globale du chatBot**

Cette procédure consiste a désactiver ou activer le chatBot pour tous clients selon le besoin.

Acteur : Les administrateurs.

Pré-condition : l'administrateur est connecté.

Scénario Principal :

- L'administrateur accède à l'interface principale de l'application.
- L'administrateur clique sur le bouton d'activation/désactivation.

Exceptions :

En cas d'erreur de connexion, l'application affiche un message d'erreur.

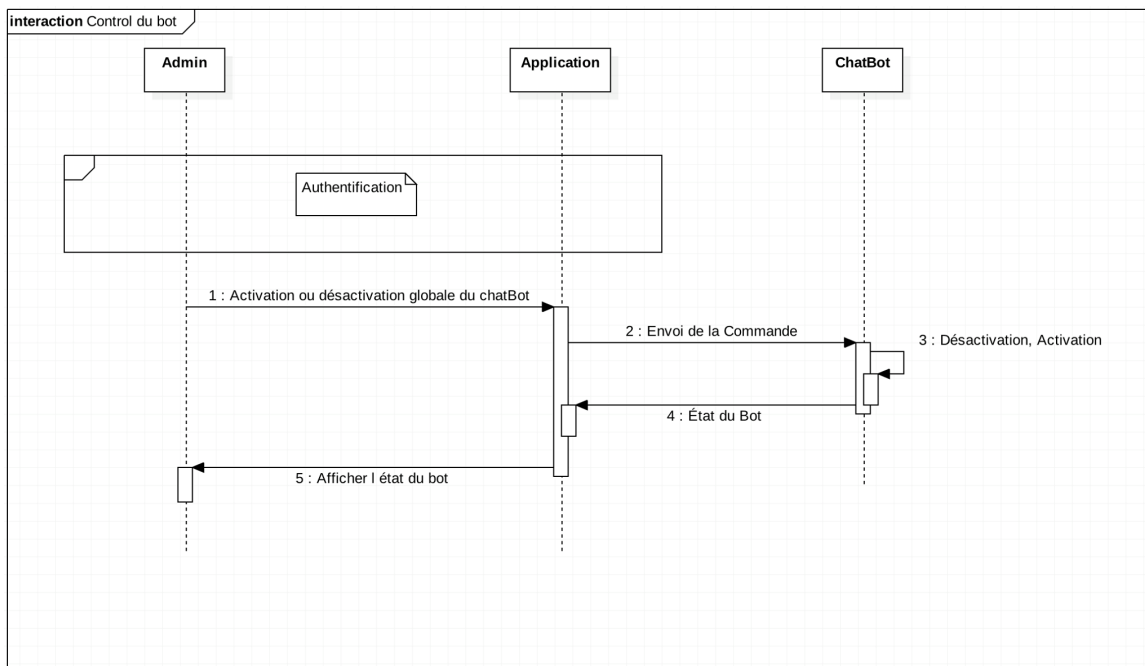


FIGURE 5.8 – Diagramme de séquence controle globale du bot

— **Envoi d'un message**

Cette procédure consiste à envoyer manuellement des messages .

Acteur :Les administrateurs.

Pré-condition :l'administrateur est connecté.

Scénario Principal :

- L'administrateur accède à l'interface principale de l'application.
- L'administrateur sélectionne le client et commence a rédiger.
- L'administrateur clique sur le bouton d'envoi.

Exceptions :

En cas d'erreur de connexion,l'application affiche un message d'erreur.

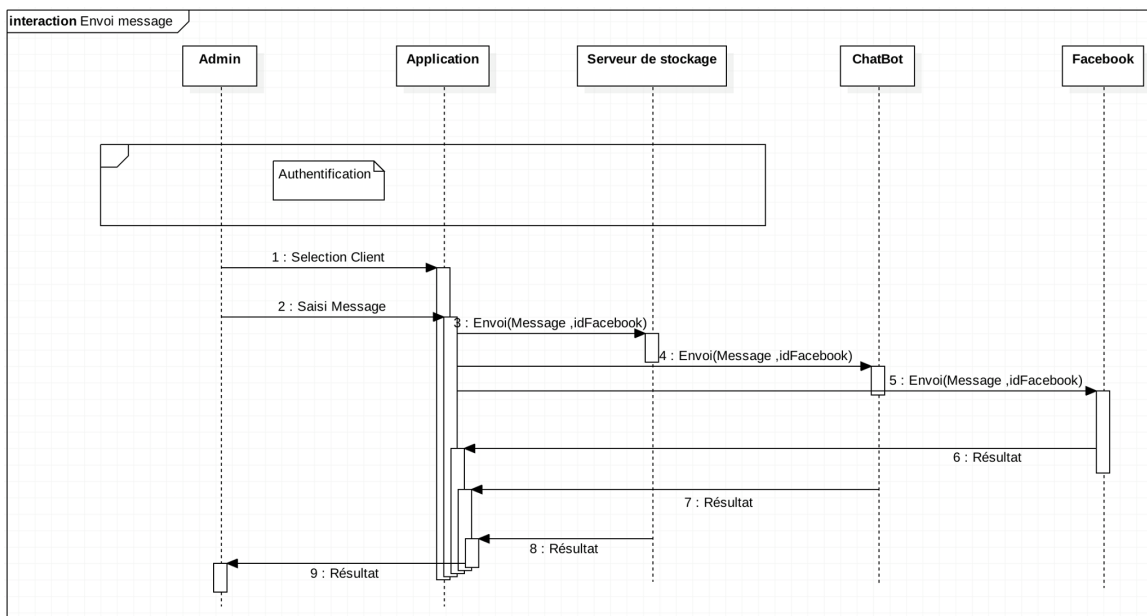


FIGURE 5.9 – Diagramme de séquence envoi de message

5.4 Réalisation

5.4.1 Introduction :

Cette phase décrit la réalisation proprement dite de notre système, c'est-à-dire citer les outils, langages de programmation et environnement de développement utilisé. Cette phase est consacrée à la présentation des outils utilisés pour réaliser le projet, et la représentation de l'application finale avec des illustrations en captures d'écran.

5.4.2 Environnement du travail :

Afin de mener à bien ce projet, nous avons utilisé un ensemble de matériels dont les principales caractéristiques sont les suivantes :

- Processeur : Intel(R) Core™ i7 @2.2 GHz
- RAM : 16.00 GO
- OS : macOS Sierra 10.12.2

5.4.3 Choix des langages et outils de développement

Le choix des outils et langages de développement revient à certains critères que l'on prend en considération, comme la facilité de manipulation et la fiabilité .

Choix de langage

Nous avons opté pour le langage python pour la programmation du chatBot car il est mieux placé pour :

- La simplicité d'écriture des scripts (facilité d'apprentissage dans un délai court).
- La plupart des bibliothèques IA sont écrites en Python. mais nous avons utilisé d'autres langages de programmation comme le PHP et Java .

Choix de l'outil de développement

Le choix de Visual Studio Code est effectué grâce à son support de plusieurs langages de programmation ainsi qu'à sa facilité , nous avons aussi utilisé d'autres outils pour le développement tel que IntelliJ IDEA pour l'interface graphique .

5.4.4 Langage de programmation

Python

[?] Python est un langage de programmation de haut niveau largement utilisé pour la programmation à usage général, créé par Guido van Rossum et publié pour la première fois en 1991. Une langue interprétée, Python a une philosophie de conception qui met l'accent sur la lisibilité du code (en utilisant notamment l'indentation des espaces blancs pour délimiter les blocs de code plutôt que des crochets ou mots-clés) et une syntaxe qui permet aux programmeurs d'exprimer des concepts en moins de lignes de code que possible.

Java

Java est un langage de programmation largement utilisé, conçu expressément dans l'environnement distribué d'Internet. C'est le langage de programmation le plus populaire.

PHP

PHP est un langage de script et un interprète librement disponible et utilisé principalement sur les serveurs Web Linux. PHP, dérivé à l'origine de Personal Home Page Tools, représente maintenant PHP : Hypertext Preprocessor.

JavaFx

JavaFX est une plate-forme logicielle pour créer et livrer des applications de bureau, ainsi que des applications Internet riches qui peuvent fonctionner sur une grande variété de périphériques.

JSON

JSON (Notation d'objet JavaScript) est un format léger d'échange de données. Il est facile pour les humains de lire et d'écrire. Il est facile pour les machines d'analyser et de générer. Il est basé sur un sous-ensemble du langage de programmation JavaScript.

XML

Un métalangage qui permet aux utilisateurs de définir leurs propres langages de balisage personnalisés, en particulier pour afficher des documents sur Internet.

5.4.5 Outils de développement

Intellij idea [?]

Est un environnement de développement intégré Java (IDE) pour le développement de logiciels informatiques. Il est développé par JetBrains (anciennement appelé IntelliJ) et est disponible sous la forme d'une édition de communauté Licence Apache 2, et dans une édition commerciale exclusive. Les deux peuvent être utilisés pour le développement commercial.

PyCharm[?]

Est un environnement de développement intégré (IDE) utilisé dans la programmation informatique, en particulier pour la langue Python. Il est développé par la société tchèque JetBrains. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration avec des systèmes de contrôle de version (VCSes) et prend en charge le développement Web avec Django. PyCharm est multiplate-forme,

avec Windows, MacOS et Linux. L'édition communautaire est diffusée sous Licence Apache, et il existe également une Edition Professionnelle publiée sous licence exclusive - ceci comporte des fonctionnalités supplémentaires.

Serveurs

Heroku

Heroku[?] a été fondée en 2007 par James Lindenbaum, Adam Wiggins et Orion Henry. Le nom "Heroku" a été créé en tant que portmanteau des mots "héros" et "haiku". Heroku a été acquis par Salesforce.com en 2010. Le siège de la société est à San Francisco. Heroku est une plate-forme cloud en tant que service prenant en charge plusieurs langages de programmation utilisés comme modèle de déploiement d'applications Web. Heroku, l'une des premières plates-formes en nuage, est en développement depuis juin 2007, alors qu'elle ne supporte que le langage de programmation Ruby, mais prend désormais en charge Java, Node.js, Scala, Clojure, Python, PHP et Go.

Tout les tests des chatBots que nous avons réalisé on été testé sur ce dernier .

Send / Receive API Facebook

Facebook est un réseau social en ligne qui permet à ses utilisateurs de publier des images, des photos, des vidéos, des fichiers et documents, d'échanger des messages, joindre et créer des groupes via sa platform Messenger.

La Send / Receive API permet non seulement d'envoyer et de recevoir des messages à l'extérieur de Messenger, mais aussi d'enrichir les messages. Les développeurs pourront créer un message d'accueil et un contexte avec différents boutons pour l'utilisateur afin de répondre le plus simplement à sa requête.

SGBD

Un système de gestion de base de données (SGBD¹) est un logiciel système conçu pour créer et gérer des bases de données. Le SGBD offre aux utilisateurs et aux programmeurs les outils nécessaires pour mettre à jour leurs données, quelle que soit leur activité.

MongoDB

MongoDB est un système de gestion de base de données orientée documents, répartitionnable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++.

1. SGBD : Système de gestion de base de données

5.4.6 Mise en place du chatBot

L'implémentation technique

Après avoir établie une étude des concepts théoriques et rassembler le maximum d'informations sur les chatBots et Facebook ,Nous avons testé les trois techniques (Aiml , ChatterBot , réseaux de neurones) mais notre choix c'est fixé sur la librairie ChatterBot. Il apparait intéressant de créer un Object Python nommé "Bot" qui a pour but de gérer les transeferes de contenu depuis le chatBot vers les clients facebook. Une Instance de ce objet "Bot" sera en mesure d'envoyer tout types de contenu autorisé par facebook, voici queqlues méthodes d'instance réalisé :

- `send_text_message(self, facebook_client_id, message)` : permet d'envoyer un message vers le client qui possède ce "facebook_client_id".
- `send_button_message(self, facebook_client_id, message,buttons)` : permet d'envoyer un message avec des boutons vers le client qui possède ce "facebook_client_id".
- `send_image_url(self, facebook_client_id, image_url)` : permet d'envoyer une photo vers le client qui possède ce "facebook_client_id".
- `send_seen(self, facebook_client_id)` : permet de notifier le client qui possède ce "facebook_client_id" que son dernier message a été bien reçu et vu!.

Lors de la phase d'apprentissage du chatBot , la méthode `train()` est lancée :

```
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer

conversation = [
    "Salut ",
    "Bonjour ",
    "c'est quoi Condor?",
    "Condor est une societe specialisee dans la fabrication ."
]
chatbot = ChatBot("CondorBot")
chatbot.set_trainer(ListTrainer)
chatbot.train(conversation)
```

la liste "conversation" est une base de connaissances pour le chatBot ou conversation[0] est l'entrée de l'utilisateur et conversation[1] est la réponse de conversation[0] et aussi conversation[2] est l'entrée de l'utilisateur et conversation[3] est la réponse ainsi de suite . Dans cette conversation, la réponse n'est qu'un texte néanmoins, dans le cas de Facebook une conversation peut contenir une image, des boutons ... et la question peut avoir plusieurs réponses successives c'est pour cela que nous avons utilisé une réponse en format Json (JsonArray) :

```
from chatterbot import ChatBot
from chatterbot.trainers import ListTrainer

conversation = [
    'vous etes qui ?',
    '{"type": "image", "rep": "http://www.condor.dz/images/../../../../condorbot.jpg"}'
```

```
,{"type": "text", "rep": "CondorBot"}, {"type": "check", "rep": "condorbot"}]'],
'condorbot On',
'[{"type": "check", "rep": "condorbot"}]',
'Salut',
'[{"type": "text", "rep": "Salut *"}]',
'Hey',
'[{"type": "check", "rep": "Salut"}]',
'Bonjour!',
'[{"type": "check", "rep": "Salut"}]',
'Salm',
'[{"type": "check", "rep": "Salut"}]',
]
chatbot = ChatBot("CondorBot")
chatbot.set_trainer(ListTrainer)
chatbot.train(conversation)
```

L'idée consiste à effectuer des tests sur l'ensemble des sous réponses de la réponse qu'est un JSONArray. Si la valeur de la clé "type" de la sous-réponse est "image" la méthode d'envoi des images sera lancé avec les paramètres d'entrées qui sont des valeurs clés dans la sous-réponse :

```
bot = Bot(PAGE-ACCESS-TOKEN)
get_repond("1547809251959501", "vous etes qui ?"):

def get_repond(sender, message):
    rep = chatbot.get_response(message)
    data = json.loads(rep.text)
    for element in data:
        if element["type"] == 'image':
            bot.send_typing(sender)
            bot.send_image_url(sender, element["rep"])
        elif element["type"] == 'text':
            bot.send_typing(sender)
            bot.send_text_message(sender, element["rep"])
        elif element["type"] == 'check':
            get_repond(sender, element["rep"])
```

De cette façon, pour chaque valeur de la clé "type" une méthode d'envoi approprié sera appelée. Toutes les sous-réponses seront envoyés suivant la position dans la réponse globale .

Remarque : si une réponse a une clé "type" avec une valeur "check" , la valeur de la clé "rep" sera redirectionnée comme une question, cette procédure évitera une certaine redondance et que plusieurs questions avec le même sens auront toujours une réponse unifiée que la valeur de la clé "rep". Additionnellement la réponse de "Salut" dans la base d'apprentissage est clairement "Salut *" , chaque "*" contenu dans une réponse sera remplacé par le nom du client sur Facebook.

À la réception de chaque message depuis Facebook, le chatBot doit exécuter une procédure comme suit :

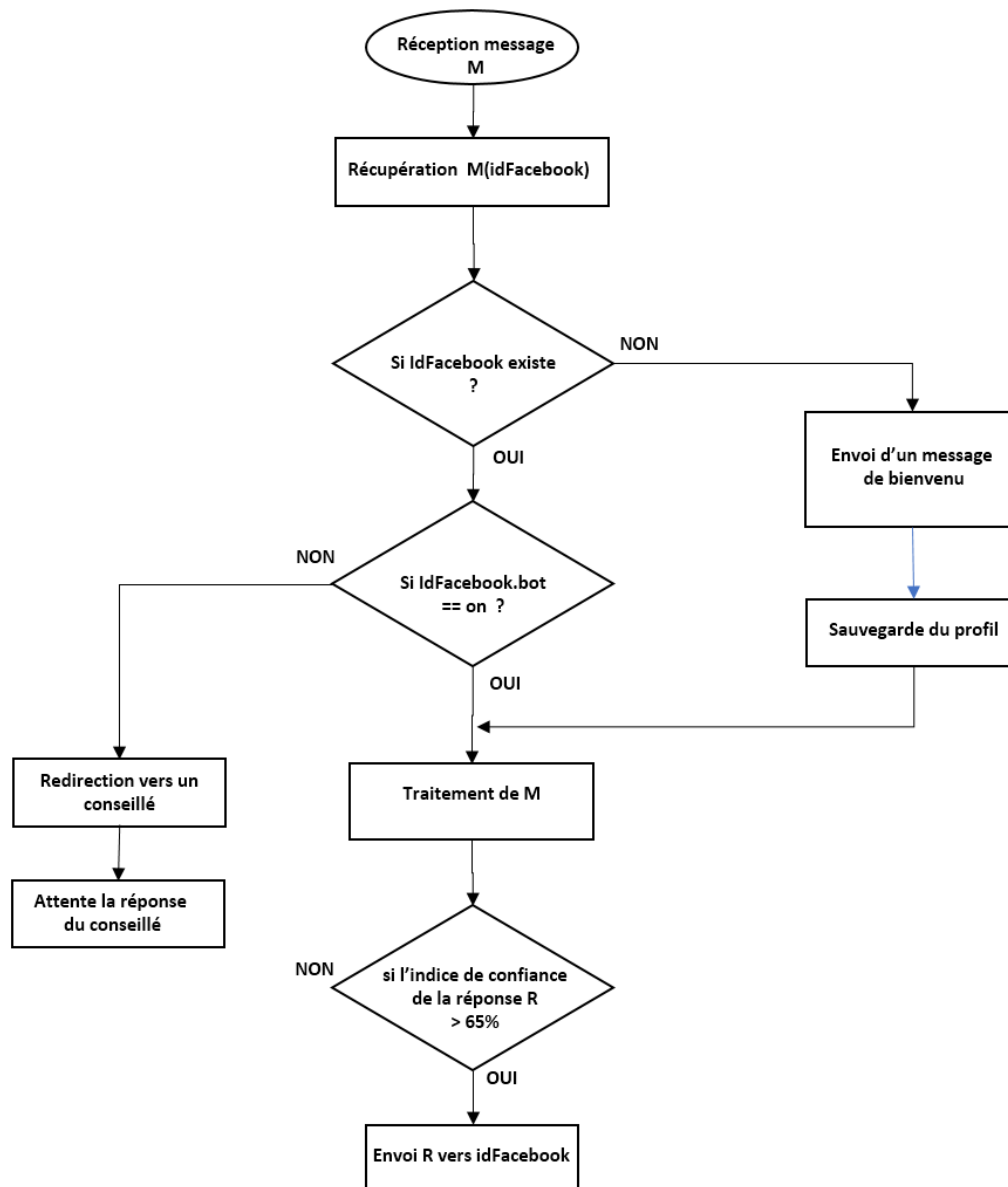


FIGURE 5.10 – Organigramme de fonctionnement du chatBot

Afin de respecter l'organigramme ?? précédant, l'instance du bot est créé avec plusieurs paramètres :

```

condorBot = ChatBot(
    'CondorBot',
    storage_adapter="SqlStorageAdapter.SQLiteDatabaseAdapter",
    logic_adapters=
    [
        {'import_path': chatterbot.logic.BestMatch'},
        {
            'import_path': 'chatterbot.logic.LowConfidenceAdapter',
            'threshold': 0.65,
        }
    ]
)
  
```

```

'default_response ':
'[{ "type": "text", "rep": "Pardon mais je vous comprend pas !!!" }
, { "type": "text_option", "rep":
  "Sinon, vous pouvez demander qu\'un conseiller de Condor
communique avec vous",
  "options": [ { "content_type": "text", "title": "Oui",
  "payload": "NEED_CONDOR" },
  { "content_type": "text", "title": "Non", "payload": "BOT_ON" } ] } ] '
],
trainer='chatterbot.trainers.ListTrainer',
silence_performance_warning=True,
database="condor-bot-database",
create=True
)

```

Présentation de l'application

Dans cette partie nous allons présenter quelques interfaces de l'application finale.

Interface Principale

Cette interface fournit à l'administrateur une liste des clients ainsi que la discussion avec ce dernier et le bouton (activer/désactiver) du chatBot pour le client en question, cette interface a été réalisé afin de superviser le chatBot et le contrôler a tout moment

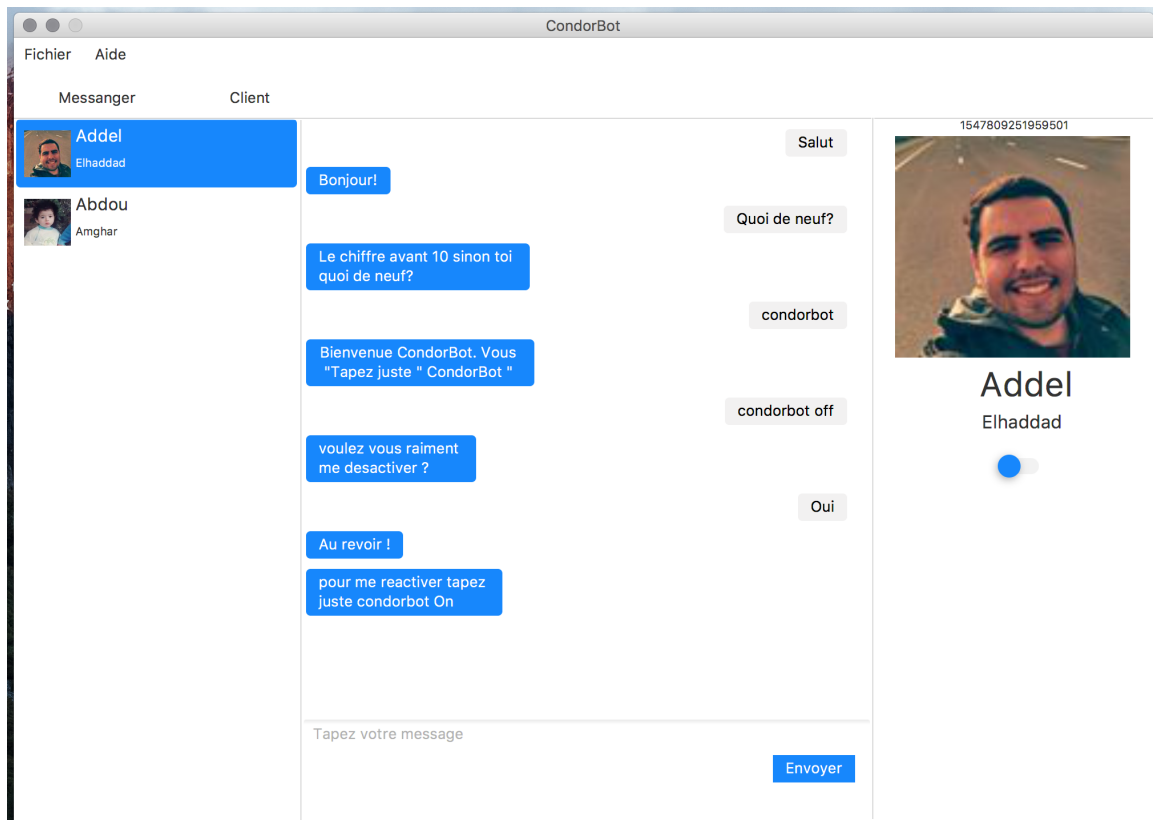


FIGURE 5.11 – capture d'écran de l'interface Principale

ChatBot côté Facebook

Quand l'administrateur accède à la page via facebook , il pourra aussi visualiser le chatBot comme sur facebook mais sans les options de contrôle .

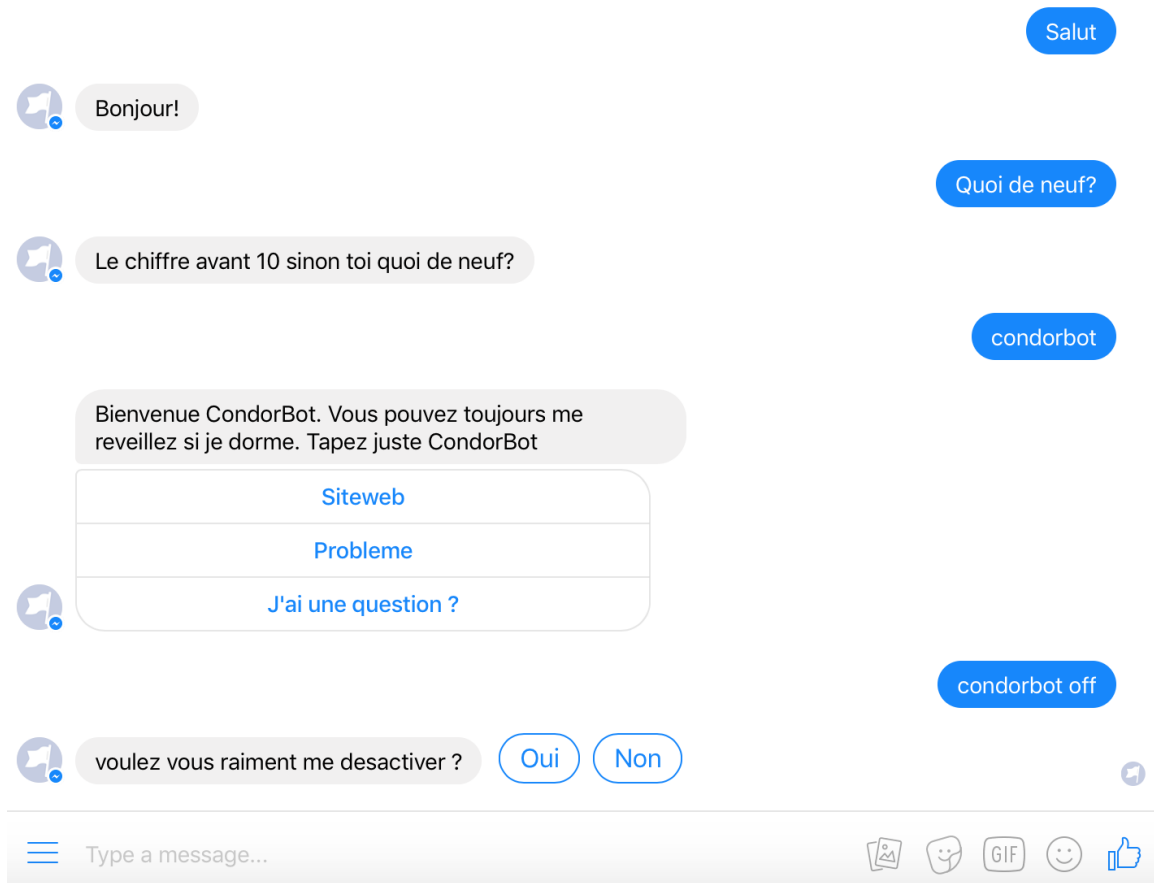


FIGURE 5.12 – capture d'écran d'une discussions a partir de Messenger

5.5 Conclusion

Dans ce chapitre nous avons commencer par réaliser un cahier des charges du chatBot ainsi que la conception de l'application de gestion de ce dernier . Nous avons aussi listé les outils de développement les langages de programmation afin de réaliser notre système .

Conclusion Générale :

Tout au long de la préparation de notre projet de fin d'études, nous avons essayé de mettre en pratique les connaissances acquises durant nos études universitaires et cela dans le but de réaliser un besoin technique de l'entreprise Condor electronics.

Ce projet de fin d'étude est le fruit d'un travail d'un stage de dix semaines, il décrit d'une façon détaillée la conception et la réalisation d'un chatBot .

Le but de ce travail a consisté principalement à concevoir et à réaliser un chatBot qui répond aux problèmes des clients avec fidélité et en temps record, nous avons en plus enrichi ce système avec une application pour contrôler le chatBot et gérer tout le flux des données .

Le système final a pu atteindre l'objectif principale demandé par l'entreprise, Comme perspective, nous souhaitons pour les prochaines améliorations d'élargir notre travail en rajoutant des techniques et des algorithmes d'optimisation afin d'assuré le confort et la satisfaction des clients ...etc . Améliorer l'application de gestion on supportent toutes les options existantes sur facebook tel que l'envoi des fichiers , photos

Sur le plan technique, cette expérience nous a permis de découvrir et d'acquérir des nouvelles connaissances, des nouveaux outils, méthodes et langages et surtout une expérience de mise en situation professionnel par la sortie avec les connaissances acquises lors de cursus de formation théorique au terrain de l'emploi.

Bibliographie

- [1] About wordnet. <https://wordnet.princeton.edu>. Dernière consultation : 2017-05-26.
- [2] Artificial intelligence markup language. <http://www.alicebot.org/aiml.html>. Dernière consultation : 2017-04-19.
- [3] Chatterbot tutorial. <http://chatterbot.readthedocs.io>. Dernière consultation : 2017-06-06.
- [4] Condor drh. <http://rh.condor.dz>. Dernière consultation : 2017-03-05.
- [5] Distance de levenshtein. <http://people.cs.pitt.edu>. Dernière consultation : 2017-04-19.
- [6] Facebook developer conference. <https://developers.facebook.com/>. Dernière consultation : 2017-06-01.
- [7] La plate-forme messenger 2.0 est arrivée. <https://developers.facebook.com/docs/messenger-platform>. Dernière consultation : 2017-06-12.
- [8] Les réseaux de neurones artificiels en informatique. <http://www.supinfo.com/articles/single/191-reseaux-neurones-artificiels-informatique>. Dernière consultation : 2017-05-15.
- [9] Official website. <https://www.python.org>. Dernière consultation : 2017-06-13.
- [10] Official website. <https://www.jetbrains.com/idea/>. Dernière consultation : 2017-06-13.
- [11] Official website. <https://www.jetbrains.com/pycharm/?fromMenu>. Dernière consultation : 2017-02-27.
- [12] Official website. <https://www.heroku.com>. Dernière consultation : 2017-06-13.
- [13] Plate-forme messenger. Dernière consultation : 2017-06-11.
- [14] Similarité et duplicate content : L'indice de jaccard. <http://b1n.sp1n.me/seo/similarite-duplicate-content-indice-jaccard.109.html>. Dernière consultation : 2017-06-07.
- [15] Tensorflow and deep learning – episode 3 – modifiez votre réseau de neurones en toute simplicité. <http://blog.xebia.fr/2017/04/11/tensorflow-deep-learning-episode-3-modifiez-votre-reseau-de-neurones-en-toute-s>. Dernière consultation : 2017-06-05.
- [16] wordnet lexical database for english. <https://wordnet.princeton.edu>. Dernière consultation : 2017-05-26.

- [17] AUDIBERT, L. Traitement automatique du langage naturel. novembre 2010.
- [18] CLIENT, L. B. T. S. Tout comprendre des chatbots. 2016.
- [19] CYBERCENTRE, S. *FACEBOOK – Présentation du réseau social*. Service Cybercentre, 22, faubourg St Michel – 44350 Guérande, 2014.
- [20] EZRATTY, O. *Les avancées de l'intelligence artificielle*. Mai 2016.
- [21] GELENBE, E. Réseaux neuronaux aléatoires stables. *Comptes rendus de l'Académie des sciences. Série 2, Mécanique, Physique, Chimie, Sciences de l'univers, Sciences de la Terre* 310, 3 (1990), 177–180.
- [22] GRAHNE, S. *UML 2 Pour modélisation orientée objet*. L'abeille, Boulevard Stiti - Tizi-Ouzou, 2009.
- [23] GRANT, A. Création d'un agent conversationnel pour orienter les clients de la maif. 03 juillet 2015.
- [24] ROQUE P, V. F. *Processus de développement UML par action de l'analyse des besoins à la conception*. EYROLLES, 2012. 4^{ème} édition 2004.