



ÉCOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE ET D'ANALYSE DES  
SYSTÈMES - RABAT

---

**Rapport de Projet de fin d'année :  
ENSIBOT un chatbot qui répond aux questions  
concernant l'ensias**

---

*Réalisé par :*

Ouaziz YASSIR  
Benkachcha YAHYA

*Encadré par :*

Pr. AHMED FAQIHI





## *Remerciements :*

Nous tenons tout d'abord à exprimer notre profonde gratitude ainsi que nos sincères remerciements à notre encadrant de projet : **Mr. AHMED FAQIHI** pour son soutien, son aide, sa présence pépétuelle et sa bienveillance durant toute la durée de l'accomplissement de ce travail.

Nous souhaitons également adresser nos vifs remerciements à nos professeurs pour le savoir, l'assistance, l'orientation, et les précieux conseils qui ont mis sur notre chemin durant toute les phases de réalisation de ce projet. En outre, nous adressons nos sincères remerciements aux membres de Jury qui ont accepté d'évaluer notre projet et de nous orienter par leurs retours élogieux et constructifs. Enfin, nos remerciements vont aussi au corps administratif de l'Ecole Nationale Supérieure de l'Informatique et de l'Analyse des Systèmes - ENSIAS pour l'atmosphère bienveillante dans laquelle ils nous ont permis de travailler. Et nous espérons que le présent écrit présentera notre travail décemment et que notre projet sera à la hauteur des espérances.



# Résumé

Le choix d'une école d'ingénieurs après l'obtention du diplôme est une étape cruciale dans la carrière de tout aspirant ingénieur. Pour s'assurer que les étudiants inscrits à l'ENSIAS ont fait le bon choix, ENSIBOT est un chatbot d'assistance qui facilite la tâche d'orientation. Après avoir collecté les questions les plus fréquemment posées sur l'ENSIAS, afin de remplir sa mission, ENSIBOT a été soigneusement construit en utilisant le célèbre assistant Watson d'IBM pour l'aspect backend des choses, et sur une interface utilisateur supportée par ReactJS, notre robot est capable de répondre aux multiples questions que les étudiants candidats se posent sur l'école. Cet article vous guidera à travers les outils et les méthodes que nous avons utilisés pour atteindre notre objectif et construire un chatbot assistant à but spécifique entièrement fonctionnel sur une interface web.

Mots-clés : Chatbot, IBM watson, Backend, ReactJs, Frontend, interface web.

# Abstract

The choice of an engineering school after graduation is a crucial step in the career of every aspiring engineer. To make sure students enroled to ENSIAS have made the right choice, ENSIBOT is an assisstant chatbot that makes the orientation task easier. After data collection of the most commonly asked questions about ENSIAS, in order to acheive it's mission, ENSIBOT was carefully built using IBM's well known watson assistant for the backend side of things, and on a ReactJS-supported UI, our bot is capable of answering multiple questions that applying students have about the school. This will guide you through the tools and methods we have used in order to acheive our goal and build a fully functional specific purpose assistant chatbot on a web interface.

**Keywords :** Chatbot, IBM watson, Backend, ReactJs, UI, Frontend, web interface.

# Liste des abréviations

Dans ce document, de multiples abréviations ont été utilisées, cette liste a pour but de les montrer :

JS : JavaScript

UI : User Interface

API : Application Programming Interface

NLP : Natural Language Processing

ML : Machine Learning

SDK : Software Development Kit

# Table des figures

1.1	Diagramme bête à corne de l'application . . . . .	6
1.2	IBM Watson . . . . .	9
1.3	HTML . . . . .	10
1.4	CSS . . . . .	10
1.5	Bootstrap . . . . .	10
1.6	JavaScript . . . . .	10
1.7	React . . . . .	10
1.8	PHP . . . . .	11
1.9	Postman for API Development . . . . .	11
1.10	PHPMyAdmin . . . . .	11
1.11	XAMPP . . . . .	11
2.1	Schema de fonctionnement . . . . .	15
2.2	Les composants du facteur positionnement . . . . .	16
2.3	Business intent . . . . .	18
2.4	Casual intent . . . . .	18
2.5	Sujets Principaux . . . . .	21
2.6	Exemples Intents de ENSIBOT . . . . .	22
2.7	Boite de dialogue de ENSIBOT . . . . .	23
2.8	Testing à l'aide de Postman . . . . .	24
2.9	Logo NodeJS . . . . .	25
2.10	Serveur lancé sur XAMPP . . . . .	25
2.11	Table user . . . . .	26
2.12	BD sur PhpMyAdmin . . . . .	26
2.13	Page login . . . . .	27
2.14	Page signup . . . . .	27
2.15	logo React . . . . .	28
2.16	UI de ENSIBOT . . . . .	28
3.1	Logo de Scrapy . . . . .	29
3.2	Logo Tensorflow . . . . .	30
3.3	Logo Python . . . . .	30
3.4	Logo IBM Watson Assistant . . . . .	30

# Table des matières

Liste des abréviations

Table des figures

Liste des tableaux

<b>Introduction générale</b>	<b>5</b>
<b>1 Analyse et conception</b>	<b>6</b>
Introduction . . . . .	6
1.1 Problématique . . . . .	6
1.2 Cahier des charges . . . . .	7
1.2.1 Contexte et objectifs . . . . .	7
1.2.2 Les besoins fonctionnels . . . . .	7
1.2.3 Les besoins non fonctionnels . . . . .	8
1.2.4 Caractéristiques de l'application . . . . .	8
1.3 État de l'art . . . . .	9
1.3.1 Les outils utilisés . . . . .	9
1.3.2 Les caractéristiques des outils utilisés . . . . .	12
Conclusion . . . . .	12
<b>2 Conception et Developpement</b>	<b>13</b>
Introduction . . . . .	13
2.1 Partie Chatbot . . . . .	13
2.1.1 Définition . . . . .	13
Définition . . . . .	13
2.1.2 Théorie des chatbots . . . . .	14
Théorie des chatbots . . . . .	14
2.1.3 Autour de IBM Watson . . . . .	19
Autour de IBM Watson . . . . .	19
2.2 Partie Backend/Frontend . . . . .	24
2.2.1 Partie Backend . . . . .	24
2.2.2 Partie Frontend . . . . .	27
<b>3 Remarques et conclusion</b>	<b>29</b>
3.1 Remarques . . . . .	29
3.2 Conclusion générale . . . . .	31
<b>Ressources</b>	<b>32</b>

# Introduction générale

Après avoir réussi leurs parcours scolaires dans les diverses institutions : classes préparatoires, FST, EST et autres. Vient l'étape où les étudiants sont ammenés à choisir l'école supérieure qui satisfait leur désir et au même temps compatible avec leurs résultats et notes.

En effet, devant l'ensemble des écoles et la contrainte de temps pour la prise de décision, le renseignement devient un fardeau pour la plupart des étudiants. Par conséquent, ces derniers ont énormément besoin d'une solide orientation pour ne pas opter simplement pour la méthode de "la bouche à l'oreille" ou bien naviguer vaguement sur le site des écoles en perdant du temps précieux. C'est dans ce sens que le projet "ENSIBOT" s'avère utile et nécessaire pour remédier à ce genre de situation et aider à prendre la bonne décision.

Le but du projet est de rendre l'orientation et la recherche faciles et fluides pour les étudiants qui souhaitent intégrer l'école. Cette application sera capable d'orienter les étudiants en répondant à leurs questions et en présentant automatiquement les informations lorsqu'elles sont demandées.

# Chapitre 1

## Analyse et conception

### Introduction

La présente partie nous permet de présenter la problématique, identifier toutes les fonctionnalités de notre application, et donner les outils de la résolution.

### 1.1 Problématique

Il s'agit de réaliser une application Web qui permet de répondre aux questions des élèves qui souhaitent intégrer l'ENSIAS, en présentant automatiquement les informations lorsqu'elles sont demandées

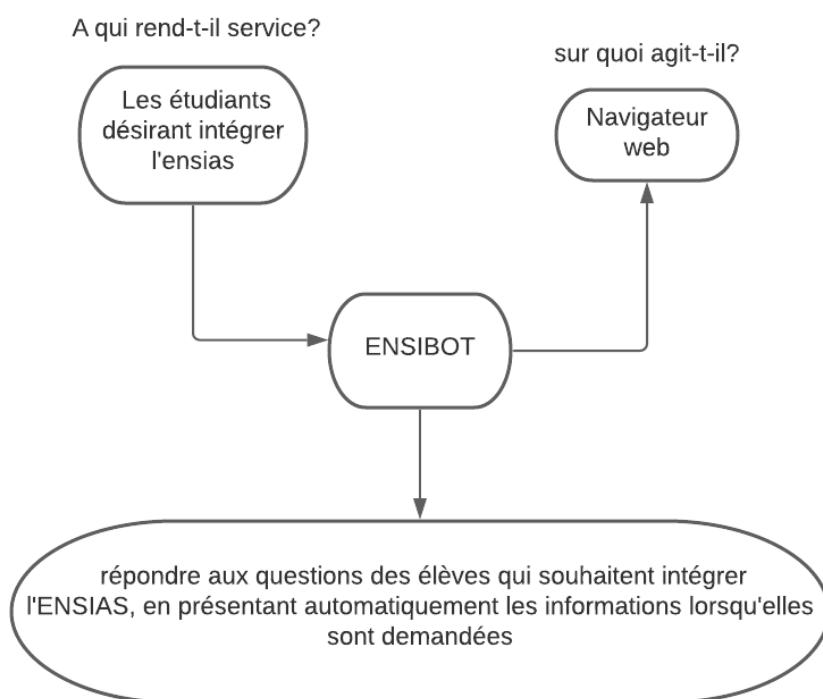


FIGURE 1.1 – Diagramme bête à corne de l'application

## 1.2 Cahier des charges

### 1.2.1 Contexte et objectifs

De nos jours, l'orientation est une nécessité pour assurer un bon avenir pour tout élève. Or il y a des outils informatiques pour faciliter la tâche et gagner du temps. On est obligé de finaliser ce projet qui consiste à assurer des réponses à toute question concernant notre école. Notre objectif à travers ce travail est de construire une application Web qui répond aux besoins de l'utilisateur.

### 1.2.2 Les besoins fonctionnels

Après une étude détaillée du système, cette partie est réservée à la description des exigences fonctionnelles des différents acteurs de l'application. Ces exigences sont :

Un utilisateur peut faire :

- **Approche général** : Notre application web propose une méthode interactif et automatisé de recherche des informations concernant notre école sous forme un chatbot.
- **Approche spécifique** : L'utilisateur peut obtenir des informations en tout ce qui concerne les filières, le parascolaire, la qualité de formation et encore plus. Ainsi qu'il sera fourni avec les documents et les liens nécessaires à donner des informations plus spécifiques.

### 1.2.3 Les besoins non fonctionnels

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

- **La disponibilité** : L’application Web doit être disponible pour être utilisé par n’importe quel utilisateur.
- **La performance** : L’application Web doit réagir dans un temp convenable, quel que soit l’action de l’utilisateur.
- **La synchronisation** : Plusieurs utilisateurs peuvent chercher dans l’application Web sans perdre de performance.

### 1.2.4 Caractéristiques de l’application

- Design simple et conviviale.
- Langue utilisée : Français.

## 1.3 État de l'art

### 1.3.1 Les outils utilisés

- o Les outils et plateformes de réalisation des chatbots :



FIGURE 1.2 – IBM Watson

**IBM Watson** est le générateur des chatbots personnalisés le plus populaire. Selon une recherche de Mindbowser en association avec Chatbots Journal, c'est la plateforme incontournable pour 61 % des entreprises. Watson possède des capacités de traitement du langage naturel inhérentes et peut comprendre plusieurs langues comme l'anglais, le français et le japonais. L'assistant virtuel Watson fourni avec la plateforme peut être utilisé pour répondre aux requêtes spontanées des utilisateurs et peut être personnalisé en fonction des besoins d'une entreprise individuelle. Watson fournit des kits de développement logiciel (SDK) pour Node.Js, Java, Python, iOS et Unity. Il est facile de démarrer le développement et permet de déployer des robots sur des plateformes de messagerie, des appareils mobiles et même des robots. Les données sur le bot sont également sécurisées car la plateforme permet aux utilisateurs de refuser le partage de données. Il offre également une intégration rapide avec un large éventail de réseaux, de canaux et d'environnement

- o Les outils de frontend :

Il ne fait aucun doute que **JavaScript** est le roi du développement web et probablement le langage le plus populaire parmi les développeurs web. C'est aussi le seul langage qui vous permet de créer des applications web, aussi bien frontales que dorsales, ainsi que des applications mobiles (React Native).

La force de Javascript ne réside pas seulement dans le fait qu'il peut être exécuté sur un navigateur et un serveur à l'aide de Nodejs, mais aussi dans les formidables cadres et bibliothèques dont il dispose pour le développement de sites Web et d'applications.

Par exemple, vous pouvez utiliser React.js et Angular pour le front-end, Nodejs pour le back-end et React Native pour créer des applications mobiles multiplateformes (Android et iOS).

**React** est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, la dernière étant une application web accessible via une page web unique. Le but est d'éviter le chargement d'une nouvelle page à chaque action demandée, et de fluidifier ainsi l'expérience utilisateur.[1]



FIGURE 1.3 – HTML



FIGURE 1.4 – CSS



FIGURE 1.5 – Bootstrap

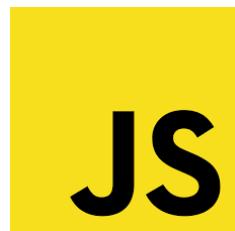


FIGURE 1.6 – JavaScript

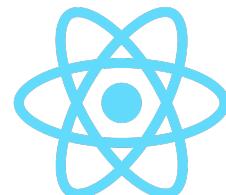


FIGURE 1.7 – React

- Les outils de backend :

**PHP** est le langage de programmation de backend web le plus populaire. Un code PHP s'exécute en tant que module de serveur Web ou en tant qu'interface de ligne de commande. Il existe plusieurs serveurs web pour exécuter des programmes PHP, comme WAMP et XAMPP.

**Xampp** est l'environnement de développement PHP le plus populaire pour les plateformes Windows, OS X et Linux.

**Xampp** signifie Cross platform(**X**), Apache(**A**), Maria db(**M**), PHP(**P**), Pearl(**P**). Il s'agit d'un serveur de distribution de logiciels qui facilite le travail des développeurs pour les tests et le déploiement en créant un serveur web local.

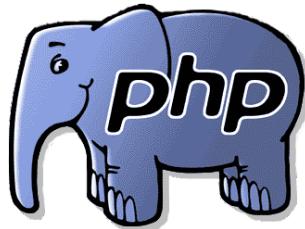


FIGURE 1.8 – PHP



FIGURE 1.9 – Postman for API Development



FIGURE 1.10 – PHPMyAdmin



FIGURE 1.11 – XAMPP

### 1.3.2 Les caractéristiques des outils utilisés

- **IBM Watson** : Plus rapide et plus performant que d’autres outils comme Amazon Lex ou bien Dialogflow.
- **phpMyadmin** : une interface facile d’utilisation pour gérer une base MySQL
- **React** : Rapide en term de loading.
- **Postman for API Development** pour assurer le bon fonctionnement d’API tout au long de sa vie

## Conclusion

Après avoir dégagé les besoins fonctionnels et opérationnels et tous les critères qu’on doit prendre en considération, et afin de modéliser les besoins attendus de notre application et que les objectifs soient atteints, on va suivre la démarche du processus unifié qu’on va le détailler dans la prochaine partie.

# **Chapitre 2**

# **Conception et Développement**

## **Introduction**

Le développement des agents conversationnels n'est pas un fruit d'hier. En effet, le premier chatbot a été créé en 1966 par Joseph Weizenbaum au MIT . Il s'agissait d'un système qui simulai un psychothérapeute rogérien, les utilisateurs étaient donc des patients. Un renouveau dans ce domaine a été possible grâce d'une part aux avancées phénoménales dans le traitement du langage naturel et les techniques d'intelligence artificielle de la dernière décennie. Et d'autre part, l'utilisation des applications de messagerie a explosé ; Le nombre d'utilisateurs mensuel des quatre plus grandes applications de messagerie a dépassé celui des quatre plus grands réseaux sociaux au premier trimestre de 2015 .D'où la propagation et la popularité des chatbots.

## **2.1 Partie Chatbot**

### **2.1.1 Définition**

Un agent conversationnel, ou un chatbot, est un programme informatique conçu pour simuler une conversation avec un utilisateur humain. Il s'agit d'un assistant virtuel qui communique avec l'utilisateur grâce à des messages textuels généralement dans une application de messagerie (par exemple Facebook Messenger, Telegram ou Skype) ou sur des sites webs. Un tel système peut vouloir imiter une vraie personne ou alors juste aider à accomplir des tâches diverses telles qu'ajouter un événement dans notre calendrier ou commander des habits en ligne. Les buts d'un chatbot peuvent donc être très variés[2]. Il existe deux types de chatbots :

- **À domaine limité (Closed Domain) :**

Ce genre de chatbot se contente d'un nombre de sujets particuliers déjà pré-définis par le programmeur. Il ne peut alors répondre qu'à des questions ayant une relation avec son domaine. Ce qui le rend certes plus facile à créer mais en contrepartie très limité car il ne satisfait pas tous les besoins de l'utilisateur[3].

- **À domaine ouvert (Open Domain) :**

Contrairement au précédent, ce chatbot est ouvert à toute catégorie de question d'où le nom domaine ouvert car il est basé sur des algorithmes d'intelligence artificielle. Cela augmente le niveau de difficulté de leur création, toutefois ils sont plus avancés et intéressants. Dans un temps où les consommateurs sont devenus hyper connectés devant leurs écrans, et dans le but d'automatiser les actions répétitifs réalisées par les humains, d'optimiser et humaniser la relation client, de gagner aussi en temps et en efficacité dans la gestion des demandes des utilisateurs, les chatbots viennent répondre à ces besoins.

**ENSIBOT** est un Closed Domain chatbot, simplement par ce qu'il a un but exact, d'aider à répondre aux questions concernantes spécifiquement l'école, et donc n'a pas besoin d'être développé afin d'avoir une normale conversation ou d'être Open Domain. Ainsi, la partie suivante fera l'objet de la théorie de réalisation de ce fameux agent conversationnel.

### 2.1.2 Théorie des chatbots

On entend dire par la théorie des chatbots tous les concepts et les fondements théoriques sur lesquels se basent un agent conversationnel. Considérons tout d'abord les deux facteurs suivants : le "Positionnement" et le "Ton et Personnalité".

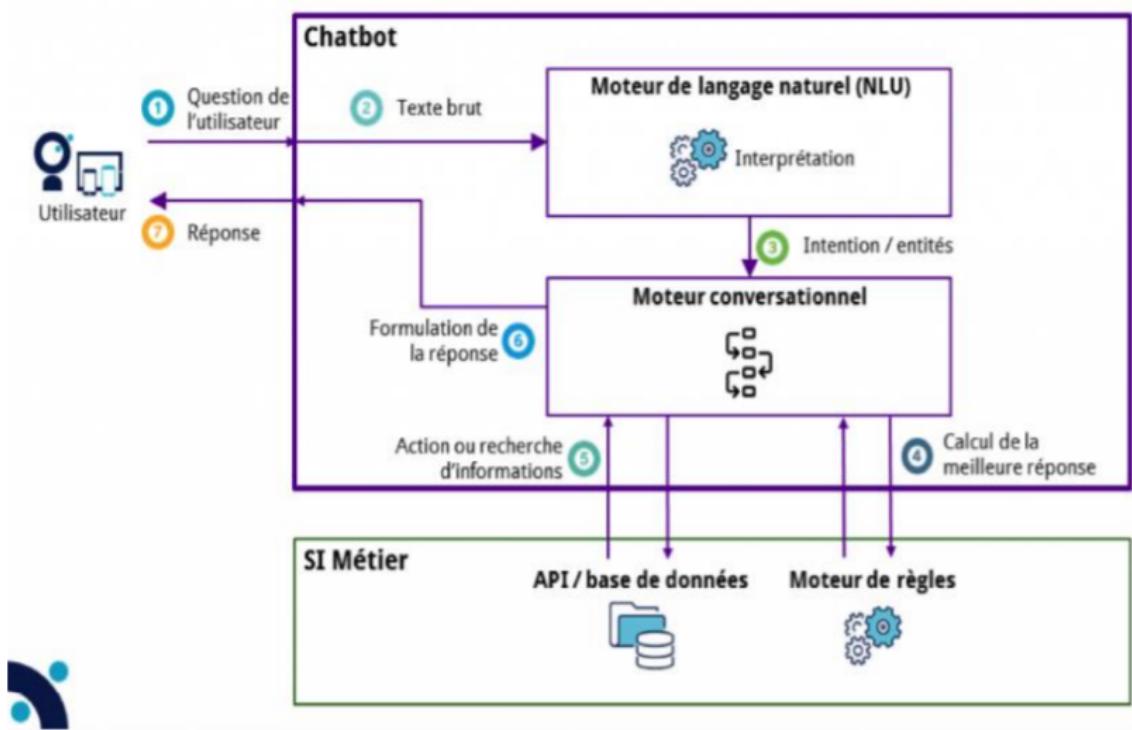


FIGURE 2.1 – Schema de fonctionnement

## 1. Le positionnement :

Le positionnement du chatbot est un terme collectif qui couvre les 3 aspects suivants : "Le but de la réponse" (Purpose), "La perspective de la réponse" (View-point) et "La proactivité" (proactivity). Notre projet vise à créer un chatbot capable de répondre à ces trois axes.

### • Le but de réponse (Purpose) :

Une interaction conversationnelle doit toujours avoir un objectif clair, c'est-à-dire un travail bien défini à faire. Par exemple clarifier ou répondre aux questions de l'utilisateur, guider les utilisateurs à trouver les informations qu'ils cherchent, aider les utilisateurs à terminer un processus et collecter les informations nécessaires pour signaler un problème.

- **La perspective de la réponse (Viewpoint) :**

C'est le rôle que la solution de chatbot doit adopter vis-à-vis de l'utilisateur. Comme un développeur, nous devons préciser le public envisagé et qu'est ce qui représente notre chatbot pour l'utilisateur (un coach, un influenceur, un ami assistant, un vendeur etc).

- **La proactivité (Proactivity) :**

C'est le degré auquel la solution engage et guide proactivement l'utilisateur. C'est-à-dire attendre que l'utilisateur pose une question, puis réagir. Le chatbot tend parfois la main à l'utilisateur et pose des questions, et parfois il attend que l'utilisateur pose une question. Dans des cas spécifiques, il utilise les deux techniques.

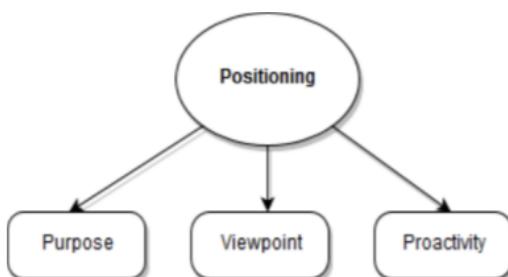


FIGURE 2.2 – Les composants du facteur positionnement

## **2. Ton & Personnalité :**

Représente la voix par laquelle le chatbot parle. Il ne faut pas confondre le ton et la personnalité avec le point de vue, qui est le rôle adopté par le chatbot. On doit déterminer le type de personnalité que les utilisateurs préfèrent ou se sentent plus à l'aise d'interagir avec. La personnalité est affectée par le choix des mots. En effet on peut créer un personnage complet pour le chatbot. De surcroit, l'humour est une caractéristique commune souvent présente dans celui-ci. Il peut augmenter la compréhension et le pardon de l'utilisateur lorsque le chatbot ne peut pas comprendre quelque chose. Cela pourrait même ravir l'utilisateur et augmenter sa satisfaction. On définit alors ces trois concepts :

- **Intentions :**

Elles font référence aux buts de l'utilisateur quand il tape une question ou un commentaire, elles permettent de catégoriser l'intention d'un utilisateur dans le cadre d'un tour de conversation. Il existe deux types d'intentions :

- Casual intents :

Ce sont ceux qui débutent et finissent la conversation tels que 'salut', 'bonjour', 'bye', 'au revoir' etc. Ils doivent diriger le bot à répondre par 'bonjour que puis-je faire pour vous ? ou 'Au revoir merci de m'avoir parler' et ainsi de suite. Ils ont aussi un rôle d'affirmation ou de refus suite à une certaine question que le bot a posé et à laquelle l'utilisateur doit répondre par exemple par 'ok' , 'oui', 'non' ...

- Business intents :

Ces intentions sont généralement sous forme d'écrits : textes et images. Elles font référence aux informations qu'un utilisateur voudrait savoir et dont la réponse n'est pas un simple oui ou non mais une rédaction. Comme par exemple les filières existantes au sein de l'école, ou les mobilités de chaque filière.

**ENSIBOT** a été créé pour se concentrer sur les "Business intents", mais il est également capable de traiter et de répondre correctement à des "Casual intents".

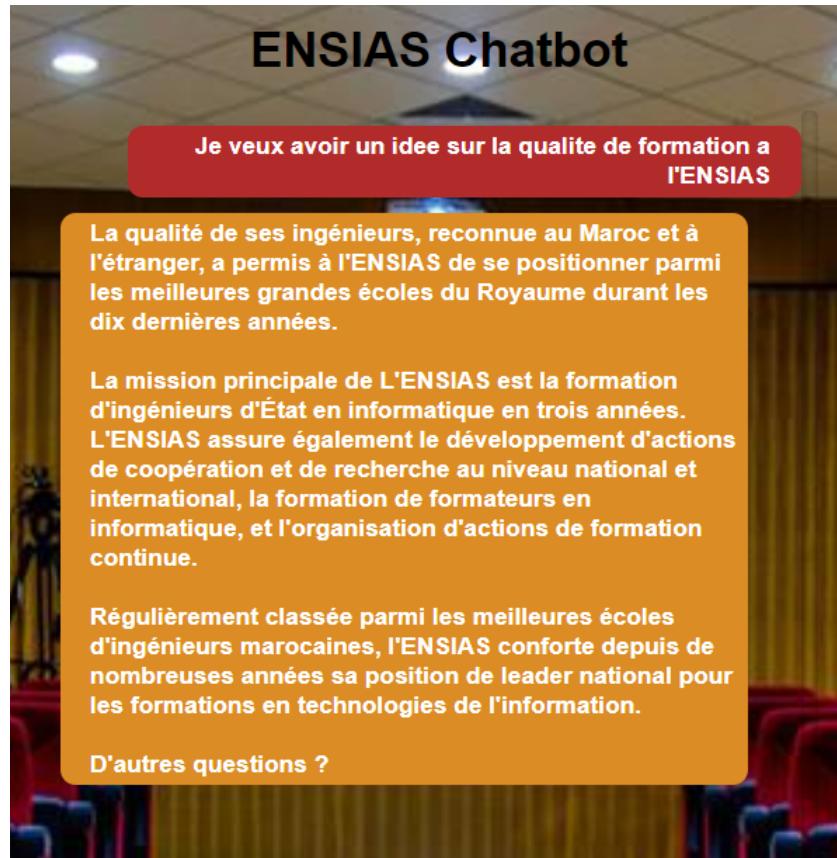


FIGURE 2.3 – Business intent



FIGURE 2.4 – Casual intent

- **Entités :**

Chaque paramètre d'intention est associé à un type, appelé type d'entité, qui détermine la manière exacte dont les données sont extraites d'une expression d'utilisateur. Une entité nous indique de quoi parle-t-on et regroupe plusieurs mots qui lui appartiennent, par exemple location, pays, horaire sont des entités.

- **Boîte de Dialogue :**

C'est une conversation entre l'utilisateur et le chatbot modélisée par un arbre qui a une racine et des noeuds. Un noeud unique avec une condition et une réponse peut gérer les demandes utilisateur simples. Pour gérer des tâches plus complexes, on imbrique des noeuds (noeud enfants) pour demander à l'utilisateur des informations supplémentaires. Il s'avère donc que pour réaliser un chatbot, il faudra définir l'ensemble des intentions des utilisateurs, les entités, et la boîte de dialogue. Et tout ceci doit être cadré au sein d'un outil.

### 2.1.3 Autour de IBM watson

IBM Watson est un outil d'intelligence artificielle conçus par IBM. Il englobe un ensemble de services sous forme d'APIs qui permettent au développeur d'ajouter de fonctionnalité et d'intégrer des services à leurs propres applications, sans avoir à savoir les détails du code derrière ces API. L'outil IBM Watson a été notre choix parce qu'il a été créé pour répondre à des questions formulées en langage naturel à l'aide du NLP, et puis a été développé pour contenir plusieurs services qui sont les suivants :

1. Watson Discovery
2. Natural Language Understading
3. Natural Language Classifier
4. Visual Recognition
5. Speech To Text
6. Text To Speech
7. Language Translator
8. Personality Insights
9. Tone Analyzer
10. Watson studio
11. Watson Machine Learning
12. Watson Assistant :

C'est l'outil auquel on va s'intéresser dans notre projet surtout dans la section suivante. Il permet de réaliser des interfaces en langage naturel pour automatiser les interactions avec les utilisateurs c'est à dire réaliser des chatbots.

Il s'avère donc que l'outil IBM Watson est très riche par ses fonctionnalités et ses services multiples qui visent la production d'applications innovantes et intelligentes.[4]

### •IBM Watson Assistant

Les assistants sont des bot cognitifs qu'on peut personnaliser selon le besoin de notre travail. IBM Assistant est un service qui crée dynamiquement un modèle d'apprentissage automatique conçu pour comprendre les demandes d'utilisateurs et des demandes similaires. Il fournit des informations sur les sujets ou les renseignements que les utilisateurs demandent. La personnalisation de l'assistante est assurée par l'ajout d'une compétence de dialogue qui comprendra et répondra aux requêtes des utilisateurs. On verra ceci en profondeur dans ce qui suit.

#### -Compétences :

La compétence de dialogue anciennement appellée Workspace, agit comme un récipient qui contient les données de formation et la logique permettant à un assistant d'aider les clients. Une compétence contient les types d'artefacts suivants :

- Intentions
- Entités
- Boîte de dialogue

Lorsqu'on ajoute des informations, la compétence utilise ces données uniques pour créer un modèle d'apprentissage automatique capable de reconnaître ces entrées utilisateur et celles similaires. Chaque fois qu'on ajoute ou qu'on modifie les données, le processus de d'apprentissage se déclenche pour garantir que le modèle sous-jacent reste à jour selon les besoins des clients et les sujets dont il souhaite discuter.

#### A.Intentions

Une intention représente le but de la saisie de l'utilisateur, c'est à dire ce que ce dernier souhaite savoir en posant sa question, autrement dit son objectif. Comme par exemple une question sur les emplacements commerciaux ou le paiement d'une facture. Celà se fait selon deux étapes définies ci-dessous :

- **Planifier et définir les intentions d'une application :**

Réfléchir à ce que les utilisateurs voudraient faire et à ce que nous voulons que notre application puisse gérer en leur nom. On les définit en utilisant le signe # au début de chaque intention.

- **Enseigner à IBM Watson vos intentions :**

Après avoir décidé les besoins que le chatbot gère pour les utilisateurs, on informe IBM Watson en définissant ces intentions dans l'interface dédié à ceci. Tel que #Acheter\_Quelque\_Chose. Le catalogue de contenu est intégré dans l'outil IBM Watson. Il fournit quelques catégories générales, telles que General et Bot Control, qui sont de bons démarreurs pour quiconque pour créer un chatbot.

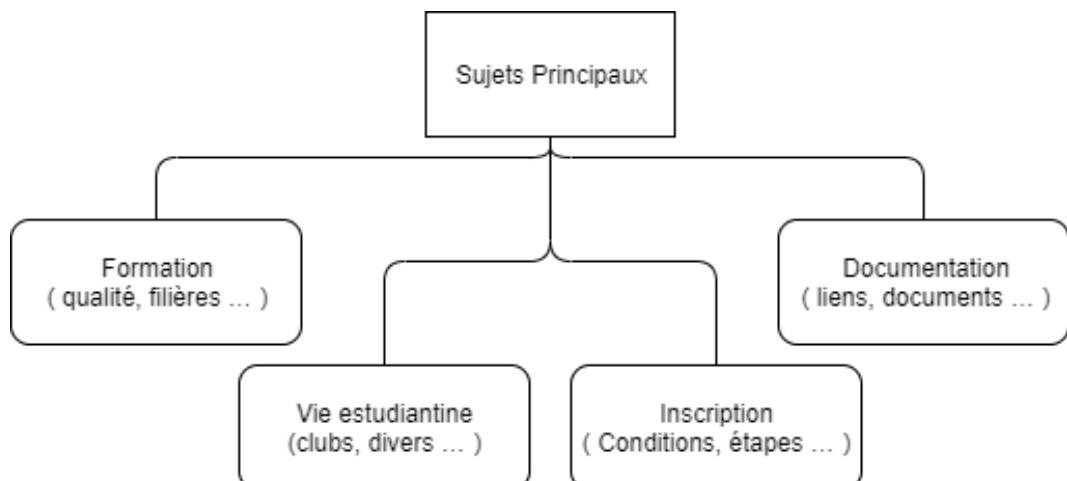


FIGURE 2.5 – Sujets Principaux

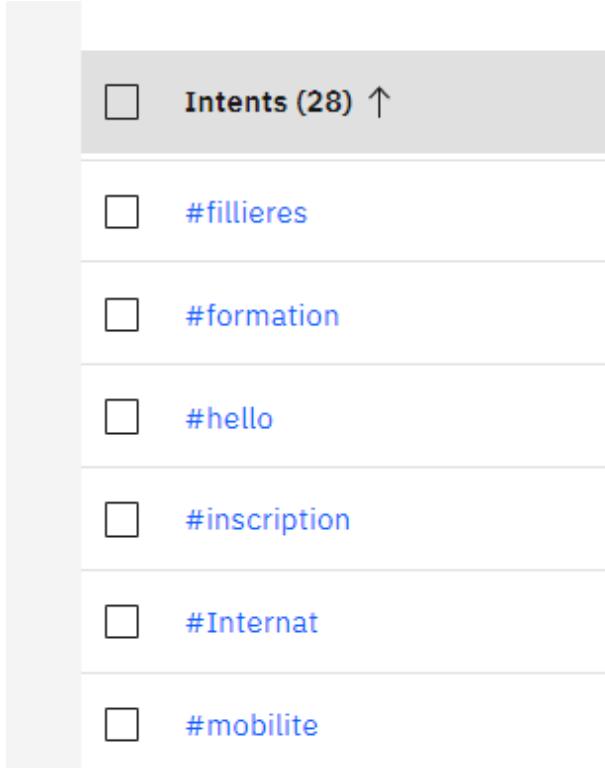


FIGURE 2.6 – Exemples Intents de ENSIBOT

## B.Entités

Les entités représentent une classe d'objet ou un type de données qui est pertinent pour l'objectif d'un utilisateur. En reconnaissant les entités mentionnées dans l'entrée de l'utilisateur, le service effectue l'action requise et donne une réponse plus ciblée et précise. Il existe deux types d'entités :

- **Entité synonyme :**

Il s'agit de spécifier un groupe de synonymes, c'est-à-dire des mots qui ont une signification exacte ou similaire. Par exemple, pour la valeur d'entité Blue, ses synonymes peuvent être Aqua, Navy et Cyan.

- **Entité modèle :**

On spécifie une expression régulière qui définit le modèle pour les mentions.

La correspondance floue ou le Fuzzy-Matching est une option qu'on trouve lors de la création des entités. C'est une fonction activée si on souhaite que le service reconnaissse les termes avec une syntaxe similaire à la valeur d'entité et aux synonymes qu'on spécifie mais sans exiger une correspondance exacte. Cette fonctionnalité est disponible pour toutes les langues prises en charge par l'assistant Watson.

## C. Boîte de dialogue

C'est un flux de branchement qui définit la façon dont le chatbot répond lorsqu'il reconnaît les intentions et les entités définies. La boîte de dialogue est composée de noeuds qui définissent les étapes de la conversation. Les noeuds du dialogue sont enchaînés dans une structure arborescente (graphiquement). Chaque noeud comprend une ou plusieurs conditions pour qu'il soit actif, ainsi qu'un objet de sortie qui définit la réponse fournie. On considère le noeud comme une construction "Si-Alors" c'est-à-dire si cette condition est vraie, le chatbot retourne la réponse appropriée. Ainsi on travail avec la structure suivante au sein du noeud :

- **Condition :**

spécifie les informations qui doivent être présentes dans l'entrée de l'utilisateur pour que le noeud soit déclenché dans la boîte de dialogue.

- **Réponse :**

précise l'énoncé avec lequel le service répond l'utilisateur.

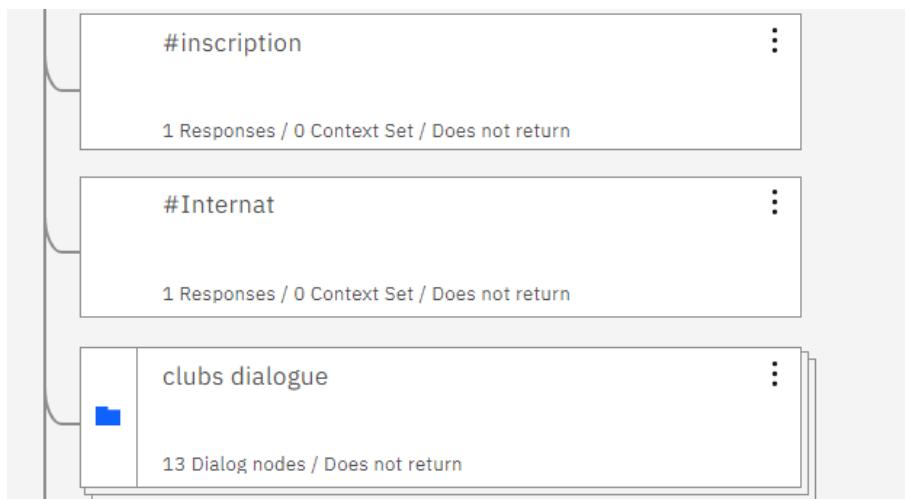


FIGURE 2.7 – Boîte de dialogue de ENSIBOT

Donc une compétence dans Watson assistant comporte trois éléments primordiales : entités, intentions et boîte de dialogue. Et ces derniers ont été utilisés afin que ENSIBOT soit capable de s'entraîner, traiter et puis répondre aux questions qu'on lui pose avec efficacité et erreur minimale.

## 2.2 Partie Backend/Frontend

### 2.2.1 Parite Backend

L'élément le plus essentiel dans ce projet est le fonctionnement sans défaut du bot, ce qui nécessite la gestion d'envoi et réception des commandes entre l'application et l'API d'IBM Watson Assistant.

Après création du serveur, la première essentielle partie est celle du testing, effectuée à l'aide de l'outil **Postman**, on peut contrôler la communication entre le serveur et l'API :

The screenshot shows two separate Postman requests side-by-side.

**Left Panel (POST Request):**

- URL: `http://localhost:5000/api/watson/message`
- Method: `POST`
- Body tab selected, showing raw JSON:

```
1
2   ...
3   "input": "bonjour"
```

**Right Panel (GET Request):**

- URL: `http://localhost:5000/api/watson/session`
- Method: `GET`
- Query Params table:

KEY	VALUE
Key	Value

Both panels have tabs for Body, Cookies, Headers (7), and Test Results. The Body tab is selected in both, showing the JSON response.

**Left Panel Response (Pretty JSON):**

```
1
2   "output": {
3     "intents": [
4       {
5         "intent": "hello",
6         "confidence": 1
7       }
8     ],
9     "entities": [],
10    "generic": [
11      {
12        "response_type": "text",
13        "text": "Bonjour, comment puis-je vous aider ?"
14      }
15    ]
16  }
```

**Right Panel Response (Pretty JSON):**

```
1
2   "session_id": "cf8e5620-3ba2-45fd-b63b-d72dc9508b53"
```

FIGURE 2.8 – Testing à l'aide de Postman

Après avoir assuré la connexion, L'etape suivante est de creer le lien entre l'API et le serveur à l'aide de **NodeJS** afin que l'envoi et reception des commandes soit automatique.



FIGURE 2.9 – Logo NodeJS

Le bot maintenant capable de fonctionner, l'étape suivante est d'effectuer un système d'identification, afin d'avoir une base de données des utilisateurs qui auront accès à ENSIBOT.

Ces taches ont été effectuées à l'aide de deux outils principaux :

Premièrement pour la creation du serveur local qui va héberger la base de données, nous avons utilisé **XAMPP**.

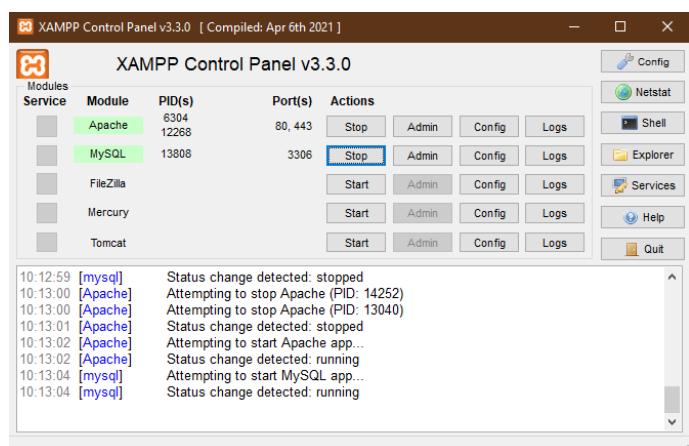


FIGURE 2.10 – Serveur lancé sur XAMPP

Le serveur lancé et prêt à être utilisé, la deuxième partie est de créer la base de données selon notre besoin, ce qui a été effectué à l'aide de **PhpMyAdmin**.

user	
*id	INT
username	VARCHAR
password	VARCHAR
email	VARCHAR
to_date	DATE

FIGURE 2.11 – Table user

	id	username	password	email	to_date
<input type="checkbox"/>	1	yahya	hehe	xyz@mail.com	2021-06-05
<input type="checkbox"/>	6	yasser	yasser777	test@mail.com	2021-06-09
<input type="checkbox"/>	4	test	mdp	wxyz@mail.com	2021-06-09

FIGURE 2.12 – BD sur PhpMyAdmin

Sur la figure 2.12, on peut constater que la table n'est pas vide, c'est parce qu'on a déjà testé le fonctionnement de la BD à travers le UI, ce qui nous mène à la deuxième grande partie : Le Frontend.

## 2.2.2 Parité Frontend

Afin d'accéder à l'interface du chatbot, l'utilisateur doit d'abord s'inscrire/s'identifier, d'où l'utilité d'une page de login, créée avec **CSS** et **HTML**.

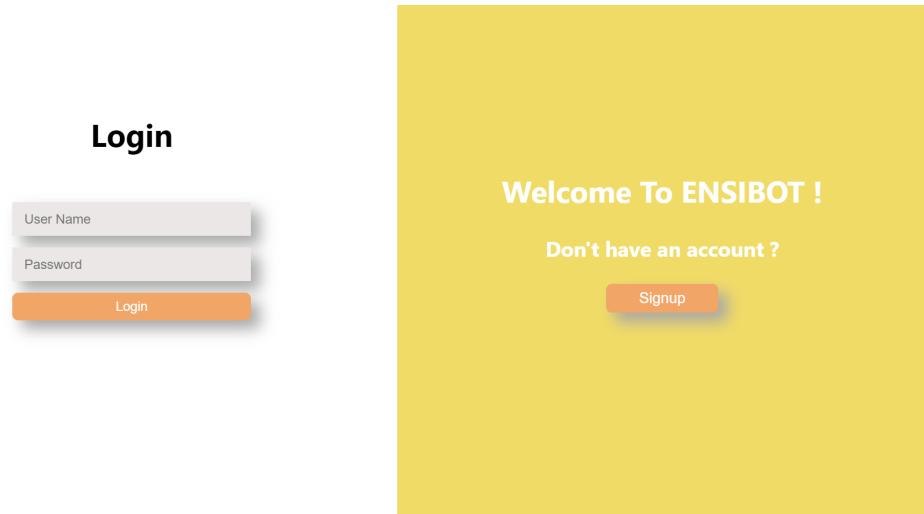


FIGURE 2.13 – Page login

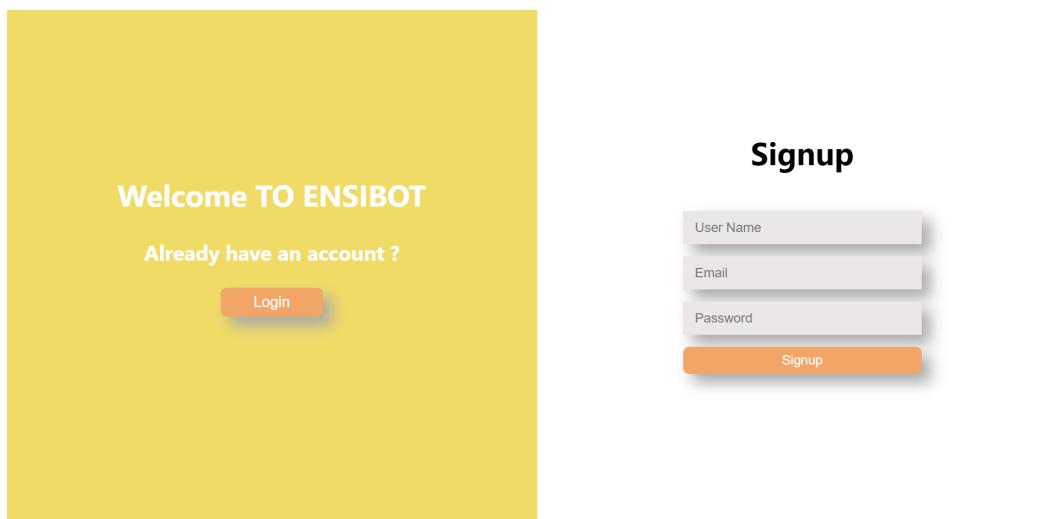


FIGURE 2.14 – Page signup

Apres une authentification réussie, l'utilisateur est enfin envoyé à la page où il pourra communiquer avec le chatbot, la dernière étant créée avec **React**.

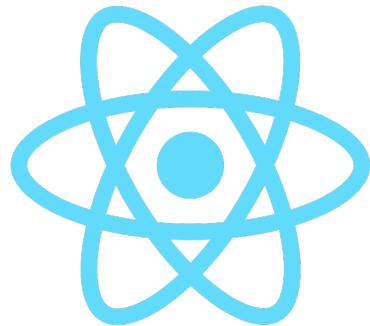


FIGURE 2.15 – logo React

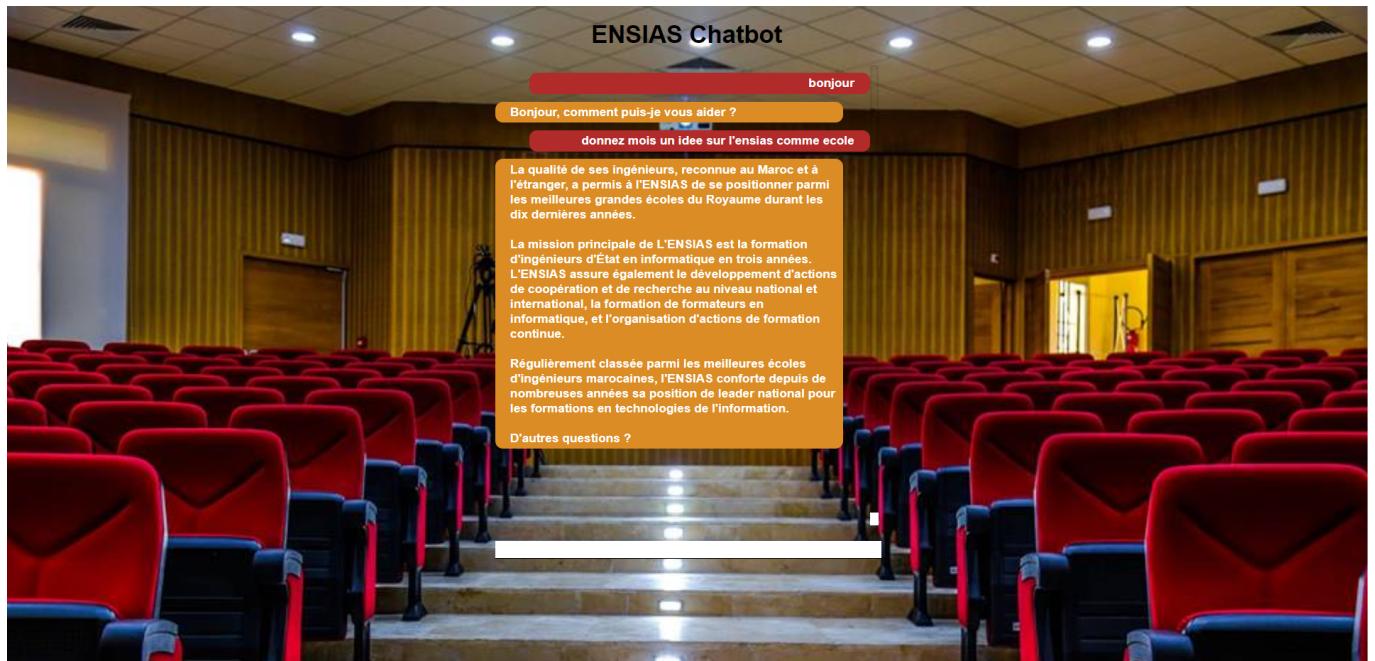


FIGURE 2.16 – UI de ENSIBOT

# Chapitre 3

## Remarques et conclusion

### 3.1 Remarques

Durant la réalisation de ce projet on a évidemment rencontré plusieurs difficultés, mais on était toujours très motivés et persévérandts pour les affronter.

- **Collection des données[6] :**

Après l'étude du cahier de charge, Nous avons compris que la collecte de données sera une partie très importante de notre projet. Notre premier recours a donc été d'essayer d'extraire des données à l'aide de la librairie Scrapy de Python à partir de médias sociaux tels que Facebook pour rechercher les questions fréquemment posées sur l'ENSIAS.



FIGURE 3.1 – Logo de Scrapy

Mais notre premier obstacle était la question de savoir s'il était autorisé de collecter des données à partir du site web, ainsi que la difficulté de la tâche elle-même, qui ne s'est pas avérée facile.

Nous avons donc abandonné l'idée et avons décidé de recueillir les questions en demandant aux étudiants de les soumettre dans un formulaire Google.

- **Creation du chatbot :**

Au départ, notre approche consistait à construire le robot à partir de zéro, en utilisant des modules d'apprentissage automatique, par l'intégration d'un NLP et de LSTM et python[5], et en entraînant le robot jusqu'à ce qu'il devienne capable de comprendre le langage humain.

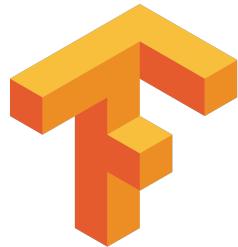


FIGURE 3.2 – Logo Tensorflow

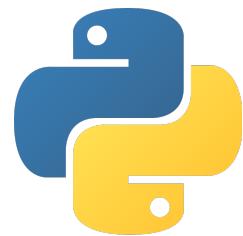


FIGURE 3.3 – Logo Python

Cependant, en raison de l'ampleur de la tâche et des contraintes de temps, nous avons eu recours à une approche plus directe en utilisant l'outil IBM Watson Assistant.



FIGURE 3.4 – Logo IBM Watson Assistant

## 3.2 Conclusion générale

En conclusion, ce projet sert l'objectif de créer un chatbot qui peut aider les étudiants qui souhaitent rejoindre les ensias ou qui ont des questions à ce sujet à trouver des réponses à leurs questions, rendant ainsi la tâche beaucoup plus accessible et plus facile.

A travers ce projet, nous avons mis à l'épreuve bon nombre de nos acquis tout au long de l'année, et ce fut une bonne occasion d'affiner nos compétences en matière de travail d'équipe, ainsi que d'apprendre de nouveaux langages et fonctionnalités que nous emporterons avec nous tout au long de notre cursus.

Le projet est encore loin d'être parfait, car il est encore possible d'améliorer de nombreux aspects, nous pouvons ajouter une plus grande gamme de questions et de réponses que le robot peut gérer, ainsi qu'améliorer l'interface utilisateur et ajouter encore plus de fonctionnalités et d'améliorations de la qualité de vie.

# Ressources

- [1] <<https://fr.wikipedia.org/>>.
- [2] **[Dana Doherty]**. Task-based interaction chatbot. *Ulster University*, 2018.
- [3] **[Jurgita Kapociute-Dzikiene]**. A domain-specific generative chatbot trained from little data. *Appl. Sci. 2 10(7), 2221*, 2020.
- [4] **[Khalil Faraj, Scott D'Angelo]**. Introduction to watson assistant, 2019.
- [5] **[Sumit Raj]**. *Building Chatbots with Python*. Apress, 2019.
- [6] **[Yasunobu Sumikawa, Masaaki Fujiyoshi, Hisashi Hatakeyama and Masahiro Nagai]**. Supporting creation of faq dataset for e-learning chatbot. *Intelligent Decision Technologies*, 2019.

