

# **Description du projet**

## **Description :**

Vu le besoin permanent de l'entreprise pour le recrutement de nouveaux employés, ainsi que l'augmentation de nombre de postulants ; les responsables des Ressources Humaines rencontrent des problèmes dans le recrutement. Dans le but d'alléger cette tâche qui pèse sur ces derniers, on s'est vu la pertinence d'introduire un Chat Bot qui serait configuré à poser certaines questions. Ainsi, il peut mener des entretiens d'embauche de présélection avec les candidats et leurs poser toutes sortes de questions qui sont généralement prisées par le responsable des Ressources Humaines. Avec les réponses fournies par les candidats, le chatbot peut dénicher ceux qui sont les plus compétents pour le poste proposé. Pour cela, le chatbot analyse les mots-clés de la requête et peut répondre à l'aide des réponses préenregistrées. Les candidats sélectionnés peuvent ensuite passer un entretien d'embauche avec un vrai recruteur.

Dans le domaine de la Ressource Humaine, le chatbot peut automatiser le processus de recrutement. En effet, il est un outil adéquat pour réaliser la première étape de sélection des candidats en analysant leur curriculum vitae et leur lettre de motivation. Il peut aussi procéder à un questionnaire d'usage avec des questions simples de type « Pouvez-vous présenter votre parcours ? ». L'étape de présélection permet d'éliminer les candidatures non pertinentes et de faire gagner un temps précieux aux recruteurs.

# Chapitre 2

---

## **Etude fonctionnelle**

*Ce chapitre s'attache à présenter dans un premier temps le contexte général du projet suivi d'une élaboration de cahier des charges ainsi que les cas d'utilisations de ce dernier.*

## Introduction

Une étude approfondie de notre projet a été conçue pour dégager par la suite les spécifications fonctionnelles et techniques et sortir avec une analyse.

## 1 Cahier des charges (Spécification générale)

Chaque entreprise aura la possibilité de gérer plusieurs fonctionnalités :

- Configurer le Chat Bot selon les préférences de chaque entreprise.
- Configurer les questions posées par le Chat bot.
- Définir la barre minimale d'acceptation des candidats.
- Voir tous les candidats acceptés par le chat bot pour chaque poste.
- Visualiser les profils des candidats acceptés.

Pour ce qui concerne le candidat, il devra passer par la même procédure et aura comme fonctionnalités :

- Inscription.
- Répondre aux questions posées par le Chat Bot dans un temps limité

### 1.1 Spécification fonctionnelles détaillées :

#### 1.1.1 Spécification de cas d'utilisation

##### ➤ Acteurs

- ✓ Représentant : L'acteur qui va représenter l'entité entreprise et qui va configurer le chat bot.
- ✓ Candidat : On inclut chaque personne qui veut postuler pour une offre.

##### ➤ Cas d'utilisation

- ✓ Pour le représentant
-

<b>Cas d'utilisation</b>	<b>Description</b>
Configurer le chat bot	L'entreprise doit s'inscrire pour pouvoir accéder au portail.
Authentification	Avant d'accéder au portail, le représentant s'authentifie
Ajouter une question	Le représentant a la possibilité d'ajouter une question.
Modifier les questions	Le représentant a la possibilité de modifier les questions que le chat bot va poser aux candidats.
Consulter les candidats	Le représentant consulte les candidats qui ont répondu au chat bot.
Modifier la barre d'acceptation et le nombre de candidats à accepter.	Le représentant choisit la barre que doit un candidat dépasser pour être accepté
Visualiser les candidats acceptés	Le représentant peut voir les candidats qui ont été acceptés par le chat bot dans un premier lieu.

**Tableau 1: les cas d'utilisation pour le représentant**

### ✓ Pour le candidat

<b>Cas d'utilisation</b>	<b>Description</b>
Description	Le candidat doit s'inscrire pour pouvoir accéder au portail.
Authentification	Avant d'accéder au portail, le candidat s'authentifie.
Choisir l'offre d'emploi	Le candidat consulte les offres et en choisit une.
Répondre aux questions du Chat bot	Le candidat répond aux questions du chat bot dans un temps limité.

**Tableau 2: les cas d'utilisation pour le candidat**

### 1.1.2 Spécification des services

- **Authentification** : l'objet de cette offre de service est de s'authentifier afin d'accéder au contenu du portail.

- **Paramètres en entrée :**

Paramètre	type	Description
Email	Char <i>Obligatoire</i>	Le candidat ou bien le recruteur doit remplir ce champ afin de pouvoir accéder au portail.
Mot de passe	Varchar <i>Obligatoire</i>	L'accès à l'espace entreprise ou l'espace candidat est sécurisé par un mot de passe

**Tableau 3: paramètres en entrée pour l'authentification**

- **Résultat retourné :**

Espace entreprise : après l'authentification le recruteur accède à son espace afin de modifier les questions du chat bot et choisir celle qui le convient ou visualiser les candidats acceptés par le chat bot pour chaque offre et même ceux qui ont été refuser ,comme il peut consulter leurs réponses aux questions.

Espace candidat : l'authentification du candidat afin de consulter les offres rejoignant son secteur d'activité et répondre au questionnaire du chat bot lié à chaque offre d'une entreprise.

- **S'inscrire dans la plateforme** : l'objet de cette offre de service est de s'inscrire dans la plateforme, les recruteurs ainsi que les candidats sont invités à remplir un formulaire afin de s'inscrire.

## ➤ Paramètres en entrée:

## ➤ Candidat :

Paramètre	Type	Description
nom	Char <i>Obligatoire</i>	Le candidat doit remplir ce champ afin de pouvoir accéder au portail.
Mot_passe	Varchar <i>Obligatoire</i>	L'accès à l'espace entreprise ou l'espace candidat est sécurisé par un mot de passe.
Secteur_activité	char <i>obligatoire</i>	Le candidat est amené à indiquer le secteur ou les secteurs d'activités dont il est intéressé.
Niveau_etude	char <i>Obligatoire</i>	Le candidat doit mentionner son niveau d'étude.

Tableau 4: paramètres en entrée pour l'inscription de candidat

## ➤ Entreprise

Paramètre	type	Description
Nom-organisme	Char <i>Obligatoire</i>	Le recruteur doit remplir ce champ afin de pouvoir accéder au portail
Mot_passe	Varchar <i>Obligatoire</i>	L'accès à l'espace entreprise ou l'espace candidat est sécurisé par un mot de passe.
Secteur_activité	char <i>obligatoire</i>	L'entreprise est amenée à indiquer son secteur d'activité.

Tableau 5: paramètres en entrée pour l'inscription de l'entreprise

- **Le Chat bot :** l'objet de cette offre est de passer un test au candidat qui doit répondre aux questions posées par le chat bot.

- **Paramètres en entrée:**

- Candidat

Paramètre	type	Description
Réponse à la question posée	Varchar	Le candidat répond au chat bot dans un temps bref , sinon sa réponse ne serait pas accepté .

**Tableau 6: Paramètres en entrée pour la**

- **Résultat retourné :**

Dès que le test est lancé, le candidat doit répondre aux questions posées par le chat bot dans un bref délai, chaque réponse est noté par le chat bot, ainsi le client saura si il est accepté dans l'offre pour un éventuel entretien.

- Entreprise

Paramètre	type	Description
Question	Varchar <i>Obligatoire</i>	Le recruteur choisit une question parmi les questions disponibles dans le chat bot
La barre d'acceptation	Int	Le recruteur pose une barre minimal qu'un candidat doit dépasser pour être accepté pour un éventuel entretien.

**Tableau 7: Paramètres en entrée pour la configuration des questions**



## 1.2 Règles de gestion

N° de règle	Règle de gestion
RG001	Chaque candidat aura la possibilité de postuler à plusieurs entreprises.
RG002	Chaque entreprise a le droit de poster plusieurs offres et annonces de travail.
RG003	Pour l'accès au portail, le représentant de l'entreprise ainsi que le candidat doivent obligatoirement être inscrit (rejoindre les utilisateurs du portail).
RG004	Le candidat ne peut pas répondre au questionnaire du chat bot qu'une seule fois

**Tableau 8: Règles de gestion**

## Conclusion

Le but de ce chapitre consiste à faire une étude approfondie pour extraire un cahier de charges et ensuite pour définir les besoins fonctionnelles. L'étape suivante consiste en une étude technique selon la deuxième branche du processus Y qui était menée en parallèle à l'étude fonctionnelle.



# Dossier technique

---

ETUDE COMPARATIVE DES DIFFERENTES SOLUTIONS DE  
L'INTELLIGENCE ARTIFICIELLE UTILISEES POUR LA REALISATION  
DU CHATBOT

REALISE PAR :

-ZOUHAIR HAITAM

-JAKA YASSIR

-MANSOURI AYMAN

ENCADRE PAR : MME.TIKITO KAWTAR

## **Introduction générale**

Dans ce présent rapport, nous avons essayé de faire une étude comparative entre les différentes solutions utilisées pour la réalisation d'un chatbot. Cette étude nous a emmené à analyser plusieurs plateformes, en déterminer les avantages et les inconvénients, mais aussi les ordonner suivant le rapport qualité/prix et le niveau de difficulté. Par conséquent, cette étude a été concentrée finalement sur trois solutions représentant les meilleures plateformes d'intelligence artificielle et qui sont :

- Microsoft Language Understanding Intelligent Service(LUIS)
- Amazon Lex
- IBM Watson

## **I-Microsoft Language Understanding Intelligent Service(LUIS)**

Cette plateforme utilise un modèle de langage pour analyser un énoncé (l'unité linguistique minimale analysée par LUIS).

Un modèle de langage est constitué de deux parties : l'intention et l'entité. La première désigne ce que l'utilisateur désire faire, par exemple : " je suis devant vous pour candidater au poste d'ingénieur d'étude " ,ici l'intention est postuler. La deuxième représente le centre d'intérêt de l'intention, c'est-à-dire ce sur quoi le chatbot prendra des actions. Dans notre exemple, c'est ingénieur d'étude. Parmi les avantages de LUIS, c'est que vous pouvez utiliser des modèles préexistants de classe mondiale de Bing et Cortana, rendre JSON facile à utiliser et stocker toutes les expressions entrantes dans la section *Logs* et fournir des fonctions d'apprentissage semi-automatiques avec suggestion, lorsque le système tente de prédire les bonnes intentions déjà présentes dans le modèle.

**Tarification:** LUIS API - Gratuit: 10 000 transactions gratuites par mois; LUIS API - Basic: jusqu'à 10 transactions par seconde; 0,75 \$ pour 1 000 transactions.

**Plateformes:** Facebook, Kik, Slack, Télégramme, Twilio, Microsoft équipes, Skype, k GroupMe, Web Chat, Email, ligne directe.

**Langues supportées:** anglais, français, italien, allemand, espagnol, portugais brésilien, japonais, coréen et chinois.

**Langages de programmation:** C# SDK, Python SDK, Node JS SDK, Android SDK.

**Fonctionnalité préconfigurée :** cette plateforme comprend des entités prédéfinies comme les nombres (en chiffres ou en texte), la température, les dimensions, l'argent, l'âge, la géographie, la date et l'heure.

## **2-Amazon Lex**

Amazon Lex est un service de création d'interfaces conversationnelles dans n'importe quelle application utilisant la voix et le texte. Amazon Lex fournit les fonctionnalités avancées d'apprentissage en profondeur de la reconnaissance vocale automatique (ASR) pour la conversion de la parole en texte, et la compréhension du langage naturel (NLU) pour reconnaître l'intention du texte. En tant que service entièrement géré, Amazon Lex évolue automatiquement, vous n'avez donc pas besoin de vous soucier de la gestion de l'infrastructure.

### **Tarification :**

- 0,004 \$ par demande vocale, et .00075 \$ par demande de texte. Votre utilisation est mesurée en "demandes traitées", qui sont additionnées à la fin du mois pour générer vos frais mensuels.
- Offre gratuite - jusqu'à 10 000 demandes de texte et 5 000 demandes de discours par mois gratuitement pour la première année.

**Plateformes:** intégration intégrée avec Facebook, Slack et Twilio; mais si vous souhaitez vous connecter à un service de messagerie non pris en charge, il existe une API pour cela: l'API Amazon Lex.

**Langues prises en charge:** Actuellement, Amazon Lex se limite à la prise en charge de l'anglais américain.

**Langages de programmation:** Vous pouvez créer des robots à l'aide de SDK: Java, JavaScript, Python, CLI, .NET, Ruby on Rails, PHP, Go et CPP.

**Fonctionnalité préconfigurée :** interopérabilité native avec AWS Lambda, AWS MobileHub et Amazon CloudWatch et intégration facile avec de nombreux autres services sur la plate-forme AWS, notamment Amazon Cognito et Amazon DynamoDB.

### **3-IBM Watson**

IBM a placé la technologie derrière Watson dans IBM Cloud et a publié l'ensemble des API que vous pouvez utiliser dans vos propres applications conversationnelles.

L'ensemble d'API comprend des offres de compréhension du langage, depuis un classificateur de langage naturel jusqu'aux aperçus de concepts et au traitement des dialogues.

**Tarification:** Les plans Lite, Standard et Premium sont disponibles.

-Lite: 10 000 appels API gratuits par mois

Jusqu'à 5 espaces de travail

Jusqu'à 25 intentions

Jusqu'à 25 entités

-Standard: \$ 0.0025 (USD) par appel API

Requêtes API illimitées / mois

Jusqu'à 20 espaces de travail

Jusqu'à 2 000 intentions

Jusqu'à 1 000 entités

-Premium: les forfaits Watson Premium offrent un niveau de sécurité et d'isolation supérieur pour aider les clients ayant des besoins en données sensibles sur demande.

**Plates-formes:** toute plate-forme de messagerie prenant en charge les paroles, les images et le texte.

**Langues prises en charge:** anglais, japonais.

**Langages de programmation:** Node SDK, Java SDK, Python SDK, iOS SDK, Unity SDK.

**Fonctionnalité préconfigurée :**

IBM Watson est construit sur un réseau de neurones (un milliard de mots Wikipedia). Il a trois composants principaux: Intents, Entités, Dialog

Vu que nous sommes nouveaux sur les Chatbots en général, il est recommandé d'utiliser le service de conversation Watson pour notre premier chatbot. Il dispose d'une interface utilisateur graphique permettant de créer notre premier chatbot sans grande difficulté. De plus, nous pouvons facilement combiner le service de conversation avec d'autres services d'intelligence artificielle, tels que le traitement du langage naturel ou d'autres services d'apprentissage automatique.



## Code de l'application :

```
import requests
import time
import threading
import msvcrt

def check():
    time.sleep(2)
    if text != None:
        return
    print("temps fini")

point=0
barre=30
domaine=["computer","technology","software","engineering"]

requis=["ingenieur","developpement web"]

print(" BONJOUR ,Presentez vous")
text=input()
text=text.replace(" ","%20")
r = requests.get('https://gateway.watsonplatform.net/natural-
language-understanding/api/v1/analyze?version=2017-02-
27&text='+text+'.&features=sentiment,keywords',
auth=('f5d6aceb-2a29-48a4-a658-892fa480926d',
'4IhrKw27mXoI'))
dict=r.json()
usage=dict.get("usage","none")
number=usage.get("text_characters")
doc=dict.get("sentiment","none").get("document","none")
```

```
sentiment=doc.get("label","none")
score=doc.get("score","none")
```

```
#voir les sentiments
if(sentiment=="positive"):
    point+=10*score
if(sentiment=="negative"):
    point-=5
if(number>200):
    point+=5
```

```
r = requests.get('https://gateway.watsonplatform.net/natural-
language-understanding/api/v1/analyze?version=2017-02-
27&text='+text+'.&features=categories', auth=('f5d6aceb-
2a29-48a4-a658-892fa480926d', '4IhrKw27mXoI'))
a=r.json()
```

```
#voir la correspondance avec les domaines
categories=a.get("categories","none")
nom=[]
scoring=[]
for i in categories:
    nom.append(i.get("label","none"))
    scoring.append(i.get("score","none"))
for i,str in enumerate(nom):
    for j in domaine:
        if j in str:
            point+=10*scoring[i]

print(point)
```

```
#-2 eme question
timeout = 5 # number of seconds your want for timeout
text= None
prompt=""
def direct_input():
    global text
    text=input()
```

```
class KeyboardThread(threading.Thread):
    def run(self):
        direct_input()
    def stop(self):
        self.stopped=True
```

```
def raw_input_with_timeout(prompt, timeout=30.0):
    global text
    print(prompt)
    finishat = time.time() + timeout
    it=KeyboardThread()
    it.start()
    while True:
        if time.time() > finishat:
            it.stop()
            return text
```

```
text=raw_input_with_timeout("quelles sont vos experiences et
vos diplomes 120 sec",120)
print("temps termine")
if text!= None:
    text=text.replace(" ", "%20")
    r = requests.get('https://gateway.watsonplatform.net/natural-
language-understanding/api/v1/analyze?version=2017-02-
```

```

27&text='+text+'.&features=keywords&keywords.limit=2&ke
ywords.sentiment=true', auth=('f5d6aceb-2a29-48a4-a658-
892fa480926d', '4IhrKw27mXoI'))
a=r.json()
keyword=a.get("keywords","none")
competences=[]
for i in keyword:
    competences.append(i.get("text","none"))

for str in competences:
    for j in requis:
        if j in str:
            point+=5

print(point)

```

## Références :

- <https://console.bluemix.net/catalog/?search=label:lite&category=watson>
- <https://chatbotsmagazine.com/worlds-biggest-ai-engines-comparison-46e421413ab>
- <https://stackoverflow.com/questions/44793687/ibm-watson-vs-amazon-lex-chatbot>
- <https://amethyste16.wordpress.com/2017/02/19/tutoriel-luis/>

