



Université de Tunis El Manar

Faculté des Sciences de Tunis



## RAPPORT DE STAGE

Elaboré par :

Mariem ZOUARI

---

Sujet : Conception et développement d'un agent conversationnel pour la génération d'applications interactives

---

Organisme d'accueil :



Superviseur du stage :

Sofiène GHARBI

Période :

Début : 01/07/2023

Fin : 31/08/2023

## Remerciements

Je tiens à remercier toutes les personnes qui ont contribué au succès de mon stage et qui m'ont aidé lors de la réalisation de ce site et la rédaction de ce rapport. J'adresse tout d'abord mes remerciements, à toute l'équipe de ST2I pour leur accueil chaleureux et leur esprit d'équipe. Je tiens à remercier vivement mon maitre de stage, Monsieur Sofiene GHARBI pour ses conseils et le partage de son expertise. Grâce aussi à sa confiance j'ai pu m'accomplir totalement dans mes missions. Il fut d'une aide précieuse dans les moments les plus délicats. Enfin, je tiens à remercier tous les contributeurs externes, les experts et les ressources open-source qui ont été utiles dans le cadre de ce projet.

## Table des matières

Remerciements.....	2
INTRODUCTION GÉNÉRALE .....	6
CHAPITRE 1 : ÉTUDE PRÉLIMINAIRE .....	7
1. Introduction.....	7
2. Présentation de l'entreprise .....	7
3. Contexte .....	7
4. Objectif .....	7
CHAPITRE 2 : la méthodologie adoptée .....	9
1. Détection des intentions .....	11
a. Présentation des intentions .....	11
b. Processus de détection .....	12
c. Evaluation du modèle .....	13
d. Les intentions .....	13
CHAPITRE 3 : La Base de Connaissances .....	22
1. Introduction.....	22
2. Structure de la Base de Connaissances .....	22
3. Exemple .....	22
4. Assistance Supplémentaire.....	23
5. Conclusion .....	23
CHAPITRE 4 : Développement de l'Interface Graphique et Intégration avec le Chatbot .....	24
1. Introduction.....	24
2. Conception de l'Interface Graphique en Angular .....	24
a. Choix de la Technologie Angular.....	24
b. Architecture de l'Application Angular .....	24
c. Conception de l'Interface Utilisateur.....	24
3. Liaison avec le Chatbot Python .....	25
a. Communication entre l'Interface Graphique et le Chatbot.....	25
b. Intégration de l'API REST .....	25
CHAPITRE 5 : Limites du Chatbot.....	26
1. Introduction.....	26
2. Limites de la Base de Données .....	26
a. Volume de Données .....	26
b. Mises à Jour .....	26

c. Personnalisation Limitée .....	26
d. Complexité des Questions .....	26
e. Méconnaissance des Scénarios Rares .....	26
3. Autres Limites du Chatbot.....	27
a. Apprentissage Continu .....	27
b. Pas d'Intelligence Artificielle Générale .....	27
4. Conclusion .....	27
CHAPITRE 6 : Conclusion Générale.....	28

## Table des illustrations

Figure 1 Logo de l'entreprise ST2I .....	7
Figure 2 Fonctionnement de l'application ChatBot .....	9
Figure 3 Diagramme d'activités du processus global .....	10
Figure 4 Scénarios des intentions.....	12
Figure 5 Métriques d'évaluation du modèle .....	13
Figure 6 Annotation manuelle des entités avec spaCy.....	15
Figure 7 Résultat d'annotation avec le modèle .....	15
Figure 8 Résultat d'extraction des entités sur l'exemple .....	15
Figure 9 Nature des exemple pour l'entraînement du modèle .....	17
Figure 10 Exemple pour le test du modèle.....	17
Figure 11 Réponse du Chatbot sur l'exemple.....	17
Figure 12 Exemple pour le test du modèle.....	18
Figure 13 Contenu du fichier "output.json" .....	19
Figure 14 Contenu du fichier "output.json" après test .....	19
Figure 15 Réponse du chatbot.....	19
Figure 16 Exemple pour tester le modèle .....	20
Figure 17 Contenu du fichier "output.json" .....	21
Figure 18 Réponse du Chatbot .....	21
Figure 19 Contenu de la base de connaissances .....	23
Figure 20 exemples pour l'assistance supplémentaire.....	23
Figure 21 Interface graphique du chatbot.....	24
Figure 22 Exemple de génération de la réponse .....	25

## INTRODUCTION GÉNÉRALE

De nos jours, l'évolution rapide de la technologie a donné naissance à des outils de communication avancés qui facilitent l'interaction entre les utilisateurs et les applications. Parmi ces technologies, les agents conversationnels ont suscité un intérêt croissant en raison de leur capacité à comprendre le langage naturel et à fournir des réponses adaptées aux requêtes des utilisateurs. Dans le cadre de mon stage d'été au sein de ST2I, j'ai entrepris de créer un agent conversationnel spécifique visant à générer des applications de manière interactive.

L'objectif principal de ce projet était de répondre à un défi majeur dans le processus de développement d'applications, à savoir la complexité et le temps nécessaire pour concevoir des interfaces utilisateur conviviales. Notre agent conversationnel cherche à simplifier cette tâche en permettant aux utilisateurs de spécifier leurs exigences fonctionnelles et esthétiques par le biais de conversations naturelles. Grâce à des algorithmes de traitement du langage naturel et à des techniques d'intelligence artificielle, l'agent est capable de comprendre les intentions des utilisateurs et de générer automatiquement des maquettes d'interfaces utilisateur en temps réel.

Ce rapport présente la combinaison du travail durant deux mois de stage. Ce dernier est organisé de la manière suivante :

- Le premier chapitre expose l'étude préliminaire du projet qui donnera une vision générale sur ce dernier.
- Le deuxième chapitre détaillera la méthodologie que nous avons adoptée pour la conception et le développement de notre agent.
- Le troisième chapitre présentera la solution finale de l'agent conversationnel.
- Le quatrième et le dernier chapitre résumera les principales réalisations du projet et présentera des pistes pour de futures améliorations.

# CHAPITRE 1 : ÉTUDE PRÉLIMINAIRE

## 1. Introduction

Ce premier chapitre a pour objectif de situer le projet dans son cadre général. Pour cela, nous commençons par définir l'organisme d'accueil et les concepts de base de notre projet. Ensuite, nous établissons une étude afin de définir les grandes lignes et les besoins du projet.

## 2. Présentation de l'entreprise

Créée début des années 90 et membre d'un groupe international d'ingénierie multidisciplinaire, le Groupe STUDI, ST2i est un bureau de consulting se positionnant sur des créneaux à forte valeur ajoutée dans le domaine de l'Informatique, de la géomatique et des nouvelles Technologies de l'Information et de la Communication.

Forte d'une équipe dirigeante, ST2i a su être à l'écoute constante de ses clients et à l'affût des innovations technologiques lui garantissant une amélioration continue et une meilleure adaptation aux changements. Ces atouts ont permis à ST2i de mener à bien multiples projets d'envergure dans plus de quinze pays : en Tunisie, en Algérie, au Mali, au Côté d'Ivoire, au Sénégal, à Burkina Faso, au Bénin, au Togo, au Cameroun, en Guinée, en RDC, en Mauritanie, en Arabie Saoudite, au Canada, etc.



*Figure 1 Logo de l'entreprise ST2i*

## 3. Contexte

Le contexte de ce projet repose sur la nécessité de simplifier et d'améliorer le processus de développement d'applications. Actuellement, la conception d'interfaces utilisateur conviviales à travers un formulaire interactif peut être complexe et demande beaucoup de temps. Afin de relever ce défi, le projet s'inspire d'une initiative précédente de génération d'applications via un formulaire interactif, en proposant une approche novatrice basée sur un agent conversationnel intelligent. L'objectif est de faciliter l'interaction entre les utilisateurs et le système de génération d'applications en remplaçant le formulaire traditionnel par un agent conversationnel intuitif, capable de comprendre les demandes des utilisateurs et de guider efficacement le processus de conception.

## 4. Objectif

Le principal objectif de ce rapport de stage est de concevoir et développer un agent conversationnel intelligent destiné à remplacer le formulaire interactif existant pour la génération d'applications. Cet agent conversationnel agira comme une interface utilisateur intuitive permettant aux utilisateurs de spécifier leurs exigences fonctionnelles et esthétiques de manière naturelle, en utilisant des conversations en langage courant.

Le projet vise à améliorer l'expérience des utilisateurs lors de la conception de leurs applications en remplaçant la saisie traditionnelle dans le formulaire par une interaction plus conviviale et interactive avec l'agent conversationnel. L'agent sera capable de comprendre et d'interpréter les intentions des utilisateurs, guidant ainsi efficacement les utilisateurs tout au long du processus de spécification de leurs besoins.

En combinant les capacités de traitement du langage naturel et l'intelligence artificielle, l'agent conversationnel permettra de réduire les obstacles liés à la saisie de données dans un formulaire, offrant une approche plus souple et facile pour les utilisateurs lors de la création de leurs applications personnalisées. Le projet vise ainsi à faciliter le processus de conception d'applications et à améliorer l'efficacité du système de génération d'applications existant.

L'évaluation de l'agent conversationnel se fera en recueillant les retours des utilisateurs concernant leur expérience d'interaction avec l'agent. Ces commentaires seront analysés pour identifier les points forts de l'agent conversationnel, ainsi que pour identifier d'éventuelles améliorations et optimisations nécessaires pour rendre l'agent plus performant et plus adapté aux besoins des utilisateurs.

En développant cet agent conversationnel innovant, le projet vise à contribuer à l'avancement de la recherche dans le domaine de la génération d'applications et de l'interaction homme-machine, ouvrant ainsi de nouvelles perspectives pour la conception d'applications personnalisées et intuitives.



## CHAPITRE 2 : la méthodologie adoptée

On peut adapter certaines parties du processus CRISP-DM (Cross-Industry Standard Process for Data Mining) :

1. Compréhension des objectifs : Identifier clairement les tâches qu'on souhaite que notre agent puisse accomplir. Dans notre cas, générer automatiquement des applications, fournir des informations sur les fonctionnalités, aider les utilisateurs à personnaliser leurs applications, etc.
2. Préparation des données : Collectez et préparez les données nécessaires pour entraîner votre agent conversationnel, en incluant des exemples de conversations utilisateur-agent, des exemples de génération d'applications, etc.
3. Modélisation : Conception l'architecture de l'agent conversationnel et choisir les algorithmes appropriés pour l'entraîner. Dans notre cas on va construire notre propre modèle.
4. Évaluation : c'est la phase des tests. On va évaluer les performances de l'agent conversationnel en utilisant des métriques appropriées. On vérifie également si notre agent répond correctement aux requêtes des utilisateurs, s'il génère des applications fonctionnelles, etc. Dans un stade avancé, on peut collecter les commentaires des utilisateurs pour améliorer notre agent.

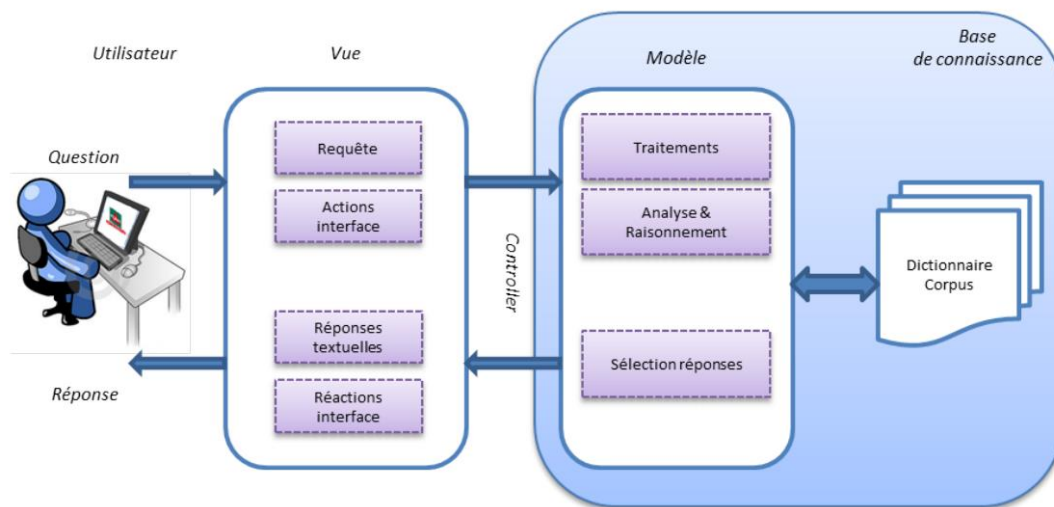


Figure 2 Fonctionnement de l'application ChatBot

La figure 2 présente le fonctionnement global de notre agent conversationnel.

Une fois connecté à sa session, l'utilisateur peut interagir avec le chatbot à travers une interface graphique : chatBox. Il peut introduire plusieurs informations que se soit pour fournir des informations pour la création de l'application ou bien demander au chatbot de faire quelques traitements.

Après avoir saisi le message et appuyé sur le bouton Envoyer, le texte sera converti en une requête et ensuite, elle sera transmise au modèle qui va réaliser différents traitements selon l'intention de l'utilisateur.

A la suite de ces traitements, la réponse sera générée à partir de la base de connaissance et sera affichée pour l'utilisateur à la fin.

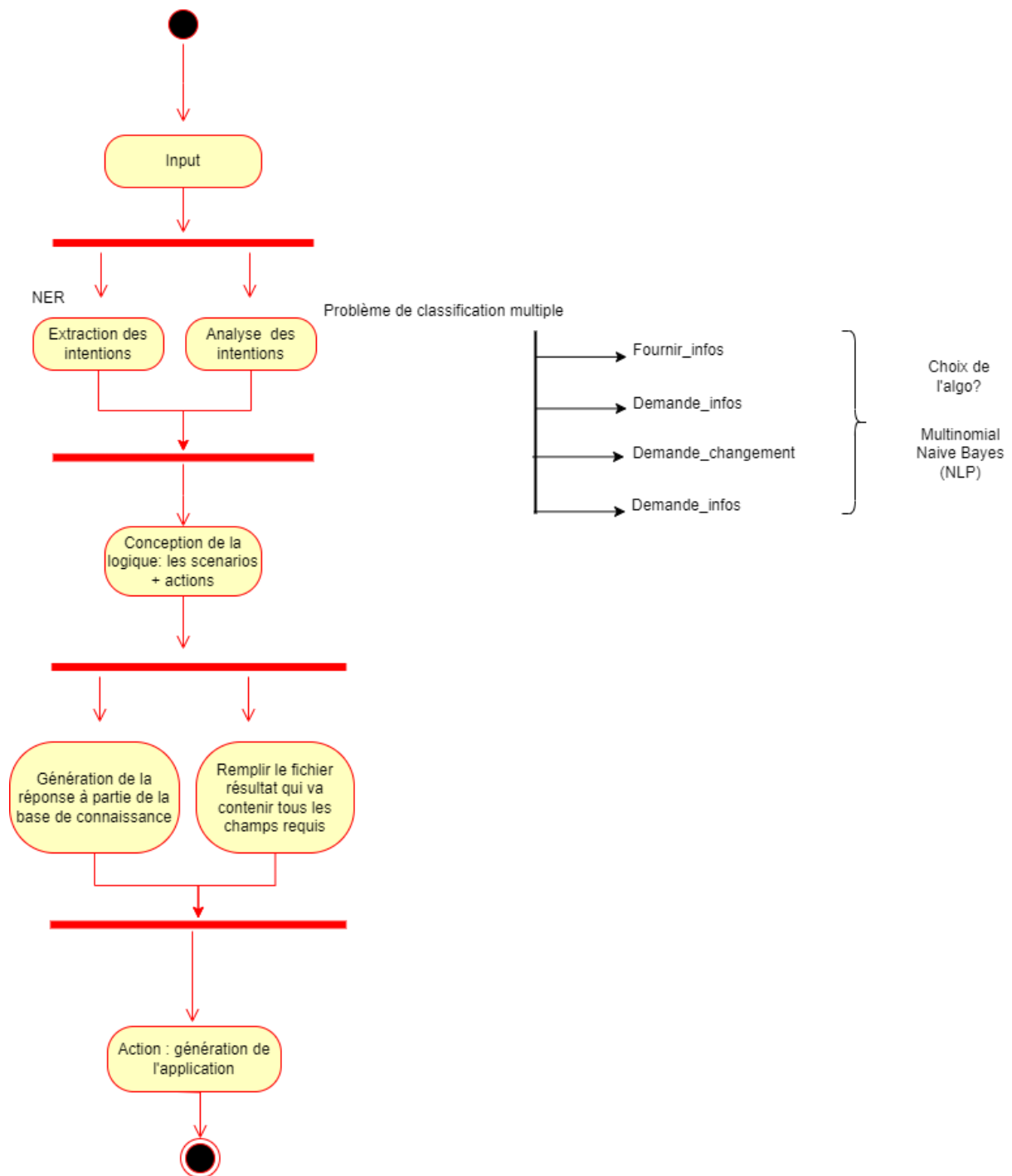


Figure 3 Diagramme d'activités du processus global

## 1. Détection des intentions :

### a. Présentation des intentions :

J'ai défini au total quatre catégories d'intentions qui couvrent la totalité des natures des demandes

Que l'on peut rencontrer dans une application de développement.

- ✓ **Fournir informations** : L'utilisateur fournit des informations pertinentes nécessaires au processus de développement de l'application. Cela peut inclure des détails sur le nom de l'application, les parties serveur et web, les langues supportées, les profils de la partie serveur, les micro-services, etc.
- ✓ **Demande aide** : L'utilisateur exprime le besoin d'obtenir de l'aide ou des clarifications sur une étape spécifique du développement de l'application. Cette intention peut être déclenchée lorsque l'utilisateur mentionne des mots-clés tels que "aide", "besoin d'aide", "explication", etc.
- ✓ **Demande changement** : L'utilisateur demande des modifications ou des ajustements à une étape précédemment spécifiée. Cela peut inclure des demandes de modification du nom de l'application, des parties serveur et web, des langues supportées, des profils de la partie serveur, des micro-services, etc.
- ✓ **Demande informations** : L'utilisateur cherche à obtenir des informations supplémentaires sur une étape spécifique du développement de l'application. Cela peut inclure des demandes de clarification, d'explication ou de détails sur les différentes étapes, les options disponibles, les bonnes pratiques, etc.

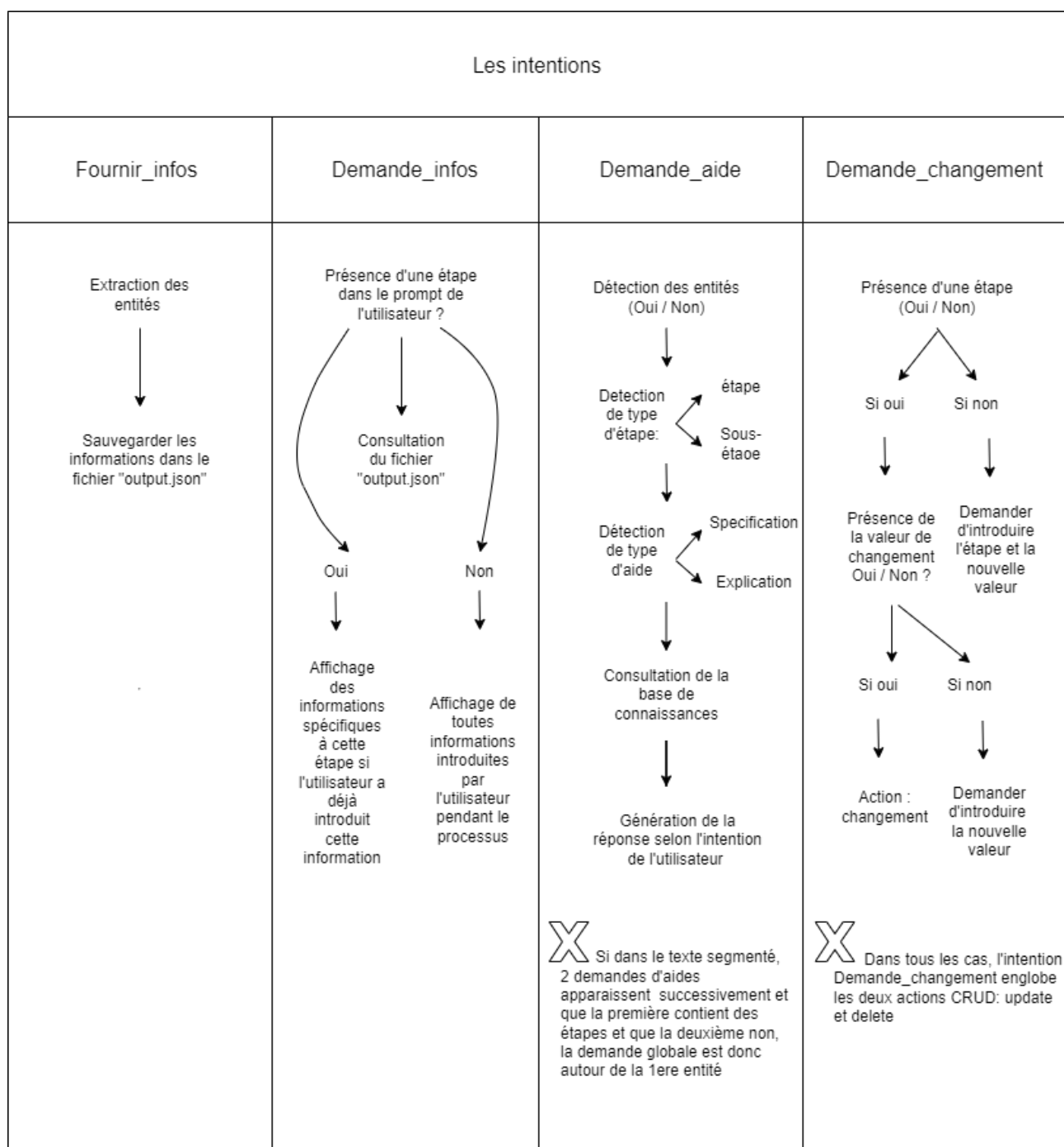


Figure 4 Scénarios des intentions

#### b. Processus de détection :

On a utilisé Multinomial Naive Bayes (MultinomialNB) comme algorithme de classification qui est basé sur le théorème de Bayes. Cet algorithme est couramment utilisé dans le domaine du traitement automatique du langage naturel (NLP) pour des tâches de classification de texte. La principale raison de son utilisation en NLP est due à ses caractéristiques et hypothèses qui se prêtent bien à la représentation des données textuelles.

Les données d'entraînement sont de la forme :

```
("Quels microservices ai-je déjà inclus dans mon projet ?", "demande_infos"),
("Quelles sont les étapes clés que je dois suivre pour terminer mon
application ?", "demande_aide"),
("J'aimerais changer certains détails concernant la partie web.",
"demande_changement")
```

#### c. Evaluation du modèle :

Pour évaluer la performance de notre modèle, nous avons utilisé plusieurs métriques d'évaluation, chacune fournissant un aperçu différent de la qualité des prédictions du modèle. Ces métriques nous permettent de quantifier la précision et l'efficacité de notre modèle sur un ensemble de données de test distinct, offrant ainsi une compréhension globale de ses performances.

On peut citer parmi ces métriques : précision, rappel et F-mesure

	precision	recall	f1-score	support
demande_aide	1.00	1.00	1.00	8
demande_changement	0.88	1.00	0.93	7
demande_infos	1.00	0.75	0.86	4
fournir_infos	1.00	1.00	1.00	11
accuracy			0.97	30
macro avg	0.97	0.94	0.95	30
weighted avg	0.97	0.97	0.97	30

Figure 5 Métriques d'évaluation du modèle

#### d. Les intentions :

- *Fournir informations :*

L'une des intentions clés que le chatbot devait comprendre et répondre était 'Fournir Informations'. Cette intention permettait aux utilisateurs de fournir des informations pertinentes nécessaires au processus de développement de l'application. Au sein du système de chatbot, cette intention se présentait comme un pilier essentiel pour recueillir des détails spécifiques des utilisateurs concernant divers aspects du projet. Parmi les informations demandées figuraient notamment le nom souhaité pour l'application, les composants serveur et web prévus, les langues devant être supportées par l'application, les spécifications des profils de la partie serveur, ainsi que la définition des micro-services nécessaires pour la mise en place de l'application.

## ❖ Extraction des entités :

### Présentation des entités :

Les entités à extraire sont essentiellement les éléments nécessaires pour la création de l'application comme le nom de l'application, l'espace de travail de la partie serveur et web, les micro-services...

### Processus d'extraction :

On a utilisé la technique NER (Named Entity Recognition) ou REN (Reconnaissance d'Entité Nommées) en français est une tâche de NLP qui a pour but d'étiqueter un mot (ou groupe de mots) d'un texte, à partir un ensemble d'étiquettes prédéfinies.

1. Préparer les données d'entraînement : on doit collecter un ensemble d'exemples annotés où le nom de l'entité est clairement marqué comme une entité nommée.

<https://agateteam.org/spacynerannotate/>

Par exemple, pour le nom de l'application :

```
[
  ("Je veux créer une application nommée MyApp", {"entities": [(23, 28, "nomApplication")] }),
  ("Crée une nouvelle application appelée XYZ", {"entities": [(30, 33, "nomApplication")] }),
]
```

2. Entraîner un modèle personnalisé : Utilisez ces données annotées pour entraîner un modèle personnalisé spaCy en utilisant la méthode `spacy.blank("fr")` pour initialiser un modèle vide, puis ajoutez et entraînez un composant d'extraction d'entités nommées.
3. Utiliser le modèle entraîné pour détecter les entités : Une fois que vous avez entraîné le modèle avec votre entité personnalisée, vous pouvez utiliser ce modèle pour détecter les entités dans le texte de l'utilisateur.

### ❖ Exemple d'utilisation du modèle :

Pour ce texte ci-dessous, on a annoté les entités manuellement à partir du site officiel de SpaCy pour vérifier avec le résultat de notre modèle.

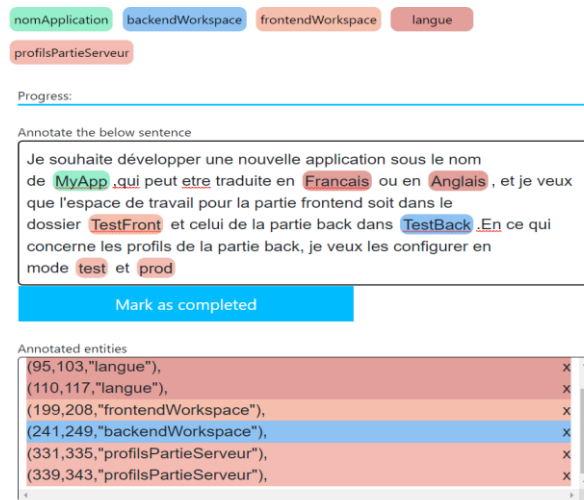


Figure 6 Annotation manuelle des entités avec spaCy

```
[(63, 68, 'nomApplication'), (95, 103, 'langue'), (110, 117, 'langue'), (199, 208, 'frontendWorkspace'), (241, 249, 'backendWorkspace'), (331, 335, 'profilsPartieServeur'), (339, 343, 'profilsPartieServeur')]
PS C:\Users\USER\Desktop\ST21>
```

Figure 7 Résultat d'annotation avec le modèle

```
{
  "backendWorkspace": [
    "TestBack"
  ],
  "frontendWorkspace": [
    "TestFront"
  ],
  "langue": [
    "Francais",
    "Anglais"
  ],
  "nomApplication": [
    "MyApp"
  ],
  "profilsPartieServeur": [
    "test",
    "prod"
  ]
}
```

Figure 8 Résultat d'extraction des entités sur l'exemple

- *Demande aide :*

Une autre facette cruciale de ma contribution au chatbot était l'intention 'Demande d'Aide'. Cette intention visait à fournir aux utilisateurs une assistance précise et ciblée tout au long de leur processus de développement d'application.

L'intention 'Demande d'Aide' a été conçue pour reconnaître les signaux indiquant que l'utilisateur pourrait rencontrer des difficultés ou nécessiter des clarifications supplémentaires. Le chatbot identifiait les mots clés, les phrases ou les contextes spécifiques qui indiquaient une demande d'assistance. Par exemple, si un utilisateur mentionnait des termes comme 'je ne sais pas comment', 'j'aurais besoin d'aide avec', ou 'pouvez-vous m'aider à comprendre', le chatbot était capable de détecter ces signaux et de réagir en conséquence.

❖ *Processus du travail :*

Le processus de demande d'aide au sein du chatbot repose sur la compréhension fine des entités, des types d'étapes et des types d'aide. Ces composants clés ont été soigneusement conçus pour offrir une assistance précise et adaptée aux utilisateurs lors de leurs interactions avec le chatbot.

a) *Entités :*

Comme détaillé auparavant, les entités présentent les différents champs du formulaire tels que le 'nom de l'application', les 'parties serveur et web', les 'langues supportées', les 'profils de la partie serveur' et les 'micro-services'. Ceci permet au chatbot de répondre à la demande d'aide d'une façon plus spécifique tout en se basant sur la base de connaissance. On va détailler cette dernière dans le chapitre suivant.

b) *Type Étape :*

Un autre aspect essentiel était la catégorisation des étapes du processus de développement. Nous avons défini deux types d'étapes : les 'étapes principales' et les 'sous-étapes'. Les étapes principales étaient des jalons clés du processus, tels que la spécification du nom de l'application, la configuration de la partie serveur et la configuration de la partie web. Les sous-étapes, quant à elles, décomposaient davantage ces étapes principales en actions spécifiques à suivre. Cette distinction a permis au chatbot de guider les utilisateurs à travers chaque phase de développement de manière structurée et progressive.

c) *Type Aide :*

En ce qui concerne les types d'aide, nous avons identifié deux catégories principales : 'explication' et 'spécification'. L'aide de type 'explication' fournissait des informations détaillées sur une étape ou une sous-étape, expliquant en profondeur son importance, son rôle et les concepts associés. L'aide de type 'spécification', en revanche, se concentrait sur les instructions pratiques, les conseils et les exemples concrets pour accomplir une étape ou une sous-étape donnée. Cette classification a permis au chatbot de fournir une assistance adaptée à la demande spécifique de l'utilisateur, qu'il s'agisse de comprendre en profondeur ou de mettre en œuvre de manière pratique.

- Spécification : le chatbot va fournir les instructions, les conseils et des exemples.

- Explication : le chatbot va fournir une explication détaillée sur l'étape.



En combinant ces éléments, le chatbot est en mesure de reconnaître les intentions des utilisateurs, d'extraire les informations pertinentes et de fournir des réponses ciblées et utiles.

### Exemple d'utilisation du modèle

ci-dessous un échantillon de la liste des données nécessaire pour l'entraînement du modèle

```
("Comment spécifier le nom de mon application ?", {"entities": [(21,43, ["nomApplication","etape","specification"])}]),  
("En quoi consiste l'étape de spécification du nom de l'application ?", {"entities": [(45,65, ["nomApplication", "etape", "explication"])}]),
```

Figure 9 Nature des exemple pour l'entrainement du modèle

Dans la première phrase, la demande d'aide est axée sur l'entité Nom d'application. Il s'agit d'une étape, et le cœur de l'aide est spécification.

On peut tester notre modèle NER sur cet exemple :

```
"Pouvez-vous m'aider à comprendre les étapes restantes pour finaliser mon application ?"
```

Figure 10 Exemple pour le test du modèle

Le chatbot va générer cette réponse en se basant sur la base de connaissances et l'historique du processus :

```
Pour finaliser le processus de génération de votre application, voici les étapes qui restent à accomplir :  
1) Spécifiez le nom de votre application.  
Sentez-vous libre de me demander si vous avez des questions supplémentaires ou si vous avez besoin d'assistance pour des étapes spécifiques de votre développement.
```

Figure 11 Réponse du Chatbot sur l'exemple

Dans cette réponse, l'utilisateur a oublié de spécifier le nom de l'application. En effet le chatbot vérifie le fichier de sortie « output.json », qui contient tous les informations requises et écrites par l'utilisateur, et s'il trouve un champ vide il a tendance à recourir à la base de connaissance pour extraire les informations relatives à ces étapes.

- *Demande de changement :*

L'intention "demande de changement" est au cœur de la capacité de l'application chatbot à gérer les demandes de modifications spécifiques dans le processus de développement d'une application. Cette fonctionnalité permet aux utilisateurs d'exprimer leurs besoins en matière de modifications, que ce soit pour le nom de l'application, les parties serveur et web, les langues supportées, les profils de la partie serveur, les micro-services, etc.

- ❖ **Processus du travail :**

Lorsque l'utilisateur exprime une demande de changement, l'application chatbot entreprend un processus méticuleux pour garantir la précision et la cohérence des modifications apportées.

1. **Analyse Contextuelle :** L'application chatbot analyse le contexte de la demande pour déterminer la portée de la modification souhaitée. Cela inclut la détection de l'étape de

développement spécifique concernée ainsi que l'action requise (ajouter, supprimer ou modifier).

2. Clarification des Détails : Si l'utilisateur n'a pas spécifié l'étape exacte ou la nouvelle valeur à modifier, l'application chatbot engage une interaction proactive. Il demande des précisions à l'utilisateur, l'invitant à fournir les informations manquantes.
3. Validation de la Modification : Lorsque l'utilisateur spécifie à la fois l'étape et la nouvelle valeur, l'application chatbot valide la demande. Il vérifie la conformité de la nouvelle valeur avec les règles et les formats requis. Si nécessaire, il propose des suggestions pour garantir l'exactitude des modifications.
4. Mise à Jour Structurée : Après validation, l'application chatbot met à jour le fichier "output.json". Selon l'action spécifiée par l'utilisateur, il ajoute la nouvelle valeur à la liste correspondante ou la supprime de manière cohérente. Cette mise à jour garantit que les étapes de développement sont toujours à jour et reflètent les demandes de l'utilisateur.

#### ❖ Exemple de Gestion de Demande de Changement

Supposons qu'un utilisateur demande : "Je souhaite changer le nom de l'application."

1. Le chatbot détecte que l'utilisateur exprime une demande de changement.
2. Étant donné que l'étape n'est pas spécifiée, le chatbot demande : "Pouvez-vous préciser l'étape que vous souhaitez modifier ?"
3. L'utilisateur répond : "Le nom de l'application."
4. Le chatbot demande ensuite : "Veuillez indiquer la nouvelle valeur que vous souhaitez appliquer."
5. Si l'utilisateur fournit la nouvelle valeur, le chatbot met à jour le fichier "output.json" en conséquence et envoie une confirmation à l'utilisateur.

Pour cet exemple, on essaie de tester notre modèle :

```
new_text = "Je veux remplacer les profils de test et pre-prod par dev et prod."
doc = repl(new_text)
```

Figure 12 Exemple pour le test du modèle

Le fichier "output.json" avant la modification est :

```
{
  "backendWorkspace": [
  ],
  "frontendWorkspace": [
  ],
  "langue": [
    "Francais",
    "Anglais"
  ],
  "microservices": [],
  "nomApplication": [
  ],
  "profilsPartieServeur": [
    "test",
    "pre-prod"
  ]
}
```

Figure 13 Contenu du fichier "output.json"

Après l'exécution du modèle, le fichier "output.json" est devenu comme suit :

```
{
  "backendWorkspace": [],
  "frontendWorkspace": [],
  "langue": [
    "Francais",
    "Anglais"
  ],
  "microservices": [],
  "nomApplication": [],
  "profilsPartieServeur": [
    "dev",
    "prod"
  ]
}
```

Figure 14 Contenu du fichier "output.json" après test

La réponse du chatbot est :

```
La valeur 'test' a été supprimée avec succès.
La valeur 'pre-prod' a été supprimée avec succès.
La valeur 'dev' a été ajoutée avec succès.
La valeur 'prod' a été ajoutée avec succès.
```

Figure 15 Réponse du chatbot

- *Demander des Informations :*

L'intention "Demander des Informations" est intrinsèque à la nature de notre chatbot, dont l'objectif principal est de guider les utilisateurs à travers le processus de génération d'applications.

Cette intention permet aux utilisateurs de solliciter des informations sur divers aspects de leur projet, qu'il s'agisse de détails sur les étapes du processus, les entités personnalisées, les langues prises en charge, les micro-services, ou d'autres éléments pertinents.

❖ Processus du travail

Lorsqu'un utilisateur exprime une demande d'informations, notre chatbot suit un processus bien défini :

1. Analyse de la Demande : Le chatbot analyse la demande de l'utilisateur pour identifier les entités pertinentes. Par exemple, si l'utilisateur demande des informations sur une étape spécifique, l'entité "Étape" est extraite.
2. Extraction des Données : Le chatbot extrait les données requises à partir de sa base de connaissances ou d'autres sources fiables.
3. Réponse Personnalisée : Le chatbot génère une réponse précise en utilisant les données extraites. Si l'entité ou les détails demandés sont déjà connus, il les fournit directement. Dans le cas contraire, il peut demander à l'utilisateur de fournir des informations complémentaires.
4. Dialogue Itératif : Si nécessaire, le chatbot peut initier un dialogue itératif pour obtenir davantage de précisions de la part de l'utilisateur afin de répondre à sa demande de manière plus approfondie.

❖ Exemple d'Interaction :

*Utilisateur : "Quelles sont les langues prises en charge pour mon application ?"*

*Chatbot : "Pour votre application, nous prenons en charge le Français et l'Anglais. "*

Cet exemple illustre comment le chatbot répond à une demande d'informations concernant les langues prises en charge. Il fournit une réponse précise et, dans ce cas, propose également une demande de précision pour répondre de manière plus complète aux besoins de l'utilisateur. Cette interaction reflète l'objectif de l'intention "Demander des Informations" d'offrir des informations pertinentes et personnalisées.

Pour l'exemple ci-dessous on va tester notre modèle :

```
new_text="Pourriez-vous me rappeler les langues que j'ai sélectionnées ?"
```

*Figure 16 Exemple pour tester le modèle*

Le fichier "output.json" est :

```
{
  "backendWorkspace": [
  ],
  "frontendWorkspace": [
  ],
  "langue": [
    "Francais",
    "Anglais"
  ],
  "microservices": [],
  "nomApplication": [
  ],
  "profilsPartieServeur": [
    "test",
    "pre-prod"
  ]
}
```

Figure 17 Contenu du fichier "output.json"

Le modèle est supposé détecter l'entité Langues et accéder au fichier sortie pour extraire les informations de cette entité si elle existe.

```
langue : Francais,Anglais,;
```

Figure 18 Réponse du Chatbot

## CHAPITRE 3 : La Base de Connaissances

### 1. Introduction

Dans ce chapitre, nous allons explorer la base de connaissances qui alimente notre chatbot de développement d'application. Cette base de connaissances est essentielle pour comprendre comment notre chatbot interagit avec les utilisateurs et fournit des informations et de l'assistance lors du processus de développement d'une application.

### 2. Structure de la Base de Connaissances

Notre base de connaissances est organisée en règles, chacune étant associée à une étape spécifique du processus de développement de l'application. Chaque règle comprend les éléments suivants :

- ✓ **Pattern** : Le motif ou la clé qui déclenche cette règle. Par exemple, "nomApplication" déclenche la première étape du processus.
- ✓ **Nom de l'Étape** : Un nom convivial qui décrit l'étape en question. Par exemple, "Nom de l'application" pour l'étape 1.
- ✓ **Étape** : Chaque règle est associée à un numéro d'ordre correspondant à son emplacement dans le processus de développement.
- ✓ **Instructions** : Les étapes spécifiques que l'utilisateur doit suivre pour accomplir cette étape.
- ✓ **Assistance** : Cette section fournit des conseils et des suggestions pour aider l'utilisateur à franchir l'étape. Elle peut également contenir des réponses aux questions fréquemment posées.
- ✓ **Valeur par Défaut** : Des valeurs préremplies que l'utilisateur peut utiliser s'il le souhaite, simplifiant ainsi le processus.
- ✓ **Conseils** : Des conseils supplémentaires pour accomplir l'étape avec succès.
- ✓ **Explication Détaillée** : Une explication plus approfondie de l'étape, de son importance et de ce à quoi l'utilisateur doit faire attention.
- ✓ **Sous-Étapes** : Le cas échéant, des sous-étapes qui peuvent découler de cette étape principale.

### 3. Exemple :

#### ❖ Étape 1 - Nom de l'Application

Dans cette section, nous détaillerons l'étape initiale du processus, qui consiste à spécifier le nom de l'application. Vous trouverez ici un aperçu des instructions précises à suivre, des astuces pour choisir un nom approprié, des valeurs par défaut disponibles et une explication détaillée de l'importance de cette étape.

```

{
  "pattern": "nomApplication",
  "nom_etape": "nom de l'application",
  "etape": 1,
  "precedence": null,
  "instructions": [
    "Spécifiez le nom de votre application.",
    "Choisissez un nom significatif et unique pour votre application.",
    "Assurez-vous que le nom de votre application respecte les conventions de dénomination appropriées."
  ],
  "assistance": [
    "Si vous avez besoin d'aide pour choisir un nom pour votre application, je peux vous donner des suggestions personnalisées.",
    "N'hésitez pas à me demander si vous avez des questions sur la dénomination des applications.",
    "Si vous avez besoin d'aide pour choisir un nom approprié pour votre application, n'hésitez pas à demander !"
  ],
  "valeur_par_defaut": ["MyApp", "QuickApp", "EasyApp", "FirstApp", "AppX", "SmartApp", "MyApplication"],
  "conseils": [
    "Choisissez un nom significatif qui décrit bien le but de votre application.",
    "Vérifiez que le nom de l'application ne contient pas de caractères spéciaux ou d'espaces inutiles."
  ],
  "explication_detaillée": "Le nom de votre application est une étape cruciale dans le processus de développement. Il s'agit du nom qui sera utilisé pour identifier votre applic.",
  "sous_etapes": null
},

```

Figure 19 Contenu de la base de connaissances

#### 4. Assistance Supplémentaire

Pour compléter notre base de connaissances, nous avons également inclus une section d'assistance supplémentaire qui offre un soutien général et des réponses aux questions fréquentes des utilisateurs.

```

"assistance_supplementaire": [
  "Si vous avez des questions supplémentaires ou si vous avez besoin d'aide pour des étapes spécifiques du développement, n'hésitez pas à me demander !",
  "Si vous souhaitez obtenir plus d'informations ou si vous avez besoin d'assistance concernant certaines étapes spécifiques du développement, je suis là pour vous aider !",
  "Si vous avez des interrogations supplémentaires ou si vous nécessitez une aide particulière pour des étapes spécifiques de votre projet, je suis à votre disposition !",
  "N'hésitez pas à me solliciter si vous avez des questions supplémentaires ou si vous avez besoin d'assistance dans des étapes spécifiques du développement.",
  "Si vous désirez plus de précisions ou si vous avez besoin d'aide pour des étapes spécifiques de votre projet, je suis là pour répondre à vos questions !",
  "Sentez-vous libre de me demander si vous avez des questions supplémentaires ou si vous avez besoin d'assistance pour des étapes spécifiques de votre développement."
]

```

Figure 20 exemples pour l'assistance supplémentaire

#### 5. Conclusion

Ce chapitre met en lumière le rôle central de notre base de connaissances dans l'interaction utilisateur. Il illustre comment cette base de connaissances rend le processus de développement plus accessible et plus efficace pour nos utilisateurs. Comprendre la structure et le fonctionnement de la base de connaissances est essentiel pour apprécier pleinement le potentiel de notre chatbot de développement d'application.

## CHAPITRE 4 : Développement de l'Interface Graphique et Intégration avec le Chatbot

### 1. Introduction

Dans ce chapitre, nous explorerons le processus de développement de l'interface graphique de notre chatbot, ainsi que la manière dont elle communique avec le chatbot Python sous-jacent. L'interface graphique en Angular joue un rôle clé dans l'interaction entre l'utilisateur et le chatbot, en offrant une expérience utilisateur intuitive et conviviale.

### 2. Conception de l'Interface Graphique en Angular

#### a. Choix de la Technologie Angular

Pour le développement de l'interface graphique, nous avons choisi d'utiliser Angular en raison de sa robustesse, de sa modularité et de sa grande communauté de développeurs. Angular offre également des fonctionnalités avancées pour la création d'applications web interactives.

#### b. Architecture de l'Application Angular

Notre application Angular suit une architecture modulaire, comprenant des composants pour la fenêtre de chat, les bulles de chat, les champs de texte, et plus encore. L'application est organisée en modules pour une meilleure gestion des dépendances.

#### c. Conception de l'Interface Utilisateur

L'interface utilisateur a été conçue pour être conviviale, avec une fenêtre de chat au centre, des bulles de chat qui affichent les messages de l'utilisateur et du chatbot, et des champs de texte pour l'entrée utilisateur. Les couleurs ont été choisies pour une meilleure lisibilité et l'expérience utilisateur.

**Générateur d'application**

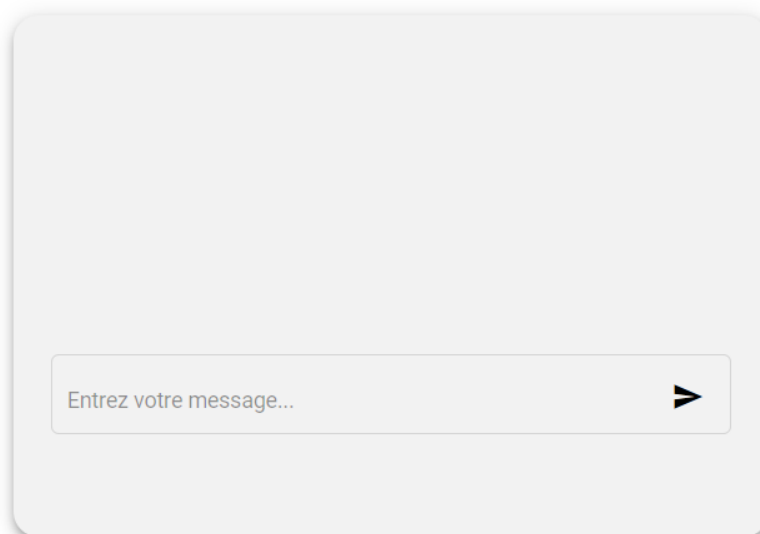


Figure 21 Interface graphique du chatbot



### 3. Liaison avec le Chatbot Python

#### a. Communication entre l'Interface Graphique et le Chatbot

La communication bidirectionnelle entre l'interface graphique Angular et le chatbot Python repose sur une API REST. L'interface graphique envoie des requêtes au serveur Flask Python pour obtenir des réponses du chatbot.

#### b. Intégration de l'API REST

Nous avons mis en place une API REST côté Python en utilisant Flask. Cette API gère les requêtes de l'interface graphique, traite les demandes du chatbot et renvoie les réponses appropriées.

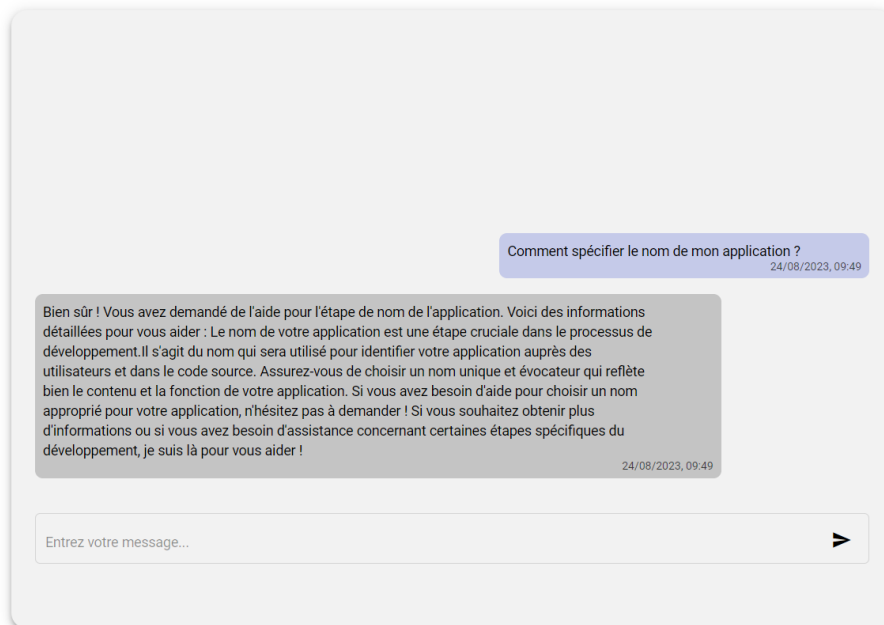


Figure 22 Exemple de génération de la réponse

## CHAPITRE 5 : Limites du Chatbot

### 1. Introduction

Bien que notre chatbot de développement d'application soit un outil puissant pour simplifier le processus de création d'applications, il est essentiel de comprendre ses limites. Ce chapitre explore en détail les différentes limites de notre chatbot, en se concentrant notamment sur les contraintes liées à la base de données.

### 2. Limites de la Base de Données

#### a. Volume de Données :

Notre base de données contient une quantité considérable d'informations, mais elle peut ne pas être exhaustive. Elle ne peut pas couvrir chaque scénario, technologie ou problème de développement d'application possible.

#### b. Mises à Jour :

Les informations dans la base de données sont susceptibles de devenir obsolètes au fil du temps. Les technologies évoluent rapidement, et notre base de données peut ne pas toujours refléter les dernières tendances et les meilleures pratiques.

#### c. Personnalisation Limitée :

Bien que notre chatbot soit conçu pour être personnalisable, il peut ne pas répondre à des besoins de personnalisation très spécifiques. Les utilisateurs ayant des exigences uniques peuvent se retrouver avec des réponses génériques.

#### d. Complexité des Questions :

Notre chatbot est capable de gérer des questions de complexité modérée, mais il peut ne pas toujours être adapté aux questions extrêmement techniques ou spécifiques.

#### e. Méconnaissance des Scénarios Rares :

Des scénarios de développement d'application extrêmement rares ou inhabituels peuvent échapper à la base de données. Dans de tels cas, les utilisateurs peuvent ne pas recevoir de réponse adéquate.

### 3. Autres Limites du Chatbot

Outre les limites de la base de données, notre chatbot présente également d'autres limites :

#### a. Apprentissage Continu :

Bien que notre chatbot soit formé pour être compétent, il ne peut pas apprendre en temps réel à partir des interactions utilisateur. Les mises à jour de la base de données sont nécessaires pour l'améliorer.

#### b. Pas d'Intelligence Artificielle Générale :

Notre chatbot est spécialisé dans le développement d'applications. Il ne possède pas une intelligence artificielle générale et ne peut pas répondre à des questions en dehors de ce domaine.

### 4. Conclusion

Comprendre les limites de notre chatbot est essentiel pour des utilisations efficaces. Les utilisateurs doivent être conscients de ces limites et comprendre que notre chatbot est un outil d'assistance, mais qu'il peut ne pas couvrir tous les scénarios possibles. La prise en compte de ces limites permettra aux utilisateurs de tirer le meilleur parti de notre chatbot tout en maintenant des attentes réalistes.

## CHAPITRE 6 : Conclusion Générale

En résumé, mon stage chez ST2I a été une expérience exceptionnelle. J'ai travaillé sur un projet innovant de développement d'un chatbot pour l'assistance au processus de création d'applications. J'ai acquis des compétences essentielles en traitement du langage naturel (NLP) et en gestion de bases de connaissances.

J'ai également eu l'opportunité de collaborer avec une équipe multidisciplinaire et d'améliorer la précision des réponses du chatbot grâce à des modèles NER (Named Entity Recognition).

Ce stage a renforcé ma passion pour l'intelligence artificielle et le développement d'applications. Je suis reconnaissant envers toute l'équipe de ST2I pour son soutien et son mentorat.

En conclusion, ce stage a été une étape cruciale de ma formation et de ma préparation à une carrière prometteuse dans le domaine de l'IA et du développement d'applications. Je remercie sincèrement tous ceux qui ont contribué à cette expérience.

