

# Rapport de projet fin d'année

**Thème :** réalisation d'un chatbot de service client de restaurant avec python et Framework flask.

**Filière :**

*Ingénierie informatique et réseaux.*

Réalisé par : **-LAHBABI AHMED**

**-HENNACH YOUSSEF**

Encadré par : **Pr. Karoum.**

# Table de matières :

<b>Introduction générale :</b>	4
<b>1 Chapitre 1 : besoins client du projet :</b>	5
1.1 Définition d'un chatbot :	5
1.2 Pourquoi utiliser un Chatbot dans le Service Client d'un restaurant?	6
1.3 Description du besoin client :	6
.....	7
1.4 Elaboration du cahier de charge :	7
1.5 Besoins fonctionnels:	8
1.6 Besoins techniques:	8
<b>2 Chapitre 2 : conception et modélisation du projet :</b>	9
2.1 Langage de modélisation:	9
2.2 Identification des acteurs et leurs rôles :	9
2.3 Modélisation:	10
2.3.1 Diagramme de cas d'utilisation :	10
2.3.2 Diagramme de classe :	12
2.3.3 Diagramme de séquence :	15
<b>3 Chapitre 3 : réalisation et développement :</b>	17
3.1 Introduction :	17
3.2 Outils logiciels :	17
3.3 Travail réalisé :	18
3.3.1 Les interfaces graphiques :	19
3.3.1.1 Interface du site web :	19
3.3.1.2 Interface de la demande la listes des restaurants :	19
3.3.1.3 Interface de la demande du menu :	21
3.3.1.4 Interface de la demande de la localisation :	22
3.3.1.5 Interface de la demande de l'horaire :	24
3.3.1.6 Interface de la demande d'une reservation :	25
3.3.1.7 Interface de l'annulation d'une reservation :	26
3.3.1.8 Interface de la demande d'effectuer une commande :	27

<b>Conclusion générale :</b> .....	29
<b>Bibliographie :</b> .....	30

## Table de figures :

Figure 1 : : Logos des restaurants kfc McDonalds et Burger King. ....	7
Figure 2: Diagramme cas d'utilisation representant les acteurs et leurs interactions.....	10
Figure 3: Diagramme de classe modélisant les différentes classes .....	12
Figure 4: Diagramme de séquence.....	16
Figure 5: Interface du site réalisé.....	19
Figure 6: Interface du site réalisé .....	20
Figure 7: Interface des menus. ....	21
Figure 8: Interface des localisations. ....	22
Figure 9: Interface de localisation de burger King. ....	23
Figure 10: Interface de localisation sur Google maps .....	24
Figure 11: Interface de l'horaire.....	25
Figure 12: Interface de reservation .....	25
Figure 13: Interface de reservation bien enregistré.....	26
Figure 14: Interface de l'annulation de la réservation .....	26
Figure 15: Interface si la reservation est annulé .....	27
Figure 16: Interface des plats proposés .....	28
Figure 17: Interface du formulaire de la commande .....	28
Figure 18: Interface de l'enregistrement de la commande.....	29

# Introduction générale :

Le présent rapport présente une étude détaillée sur le développement et l'implémentation d'un chatbot service client de restauration. Dans un monde où les entreprises cherchent constamment à améliorer leur relation avec leurs clients, l'utilisation de chatbots intelligents pour offrir un support client efficace et personnalisé est devenue une tendance majeure. Ce projet vise à concevoir et à mettre en œuvre un chatbot capable de répondre aux besoins et aux demandes des clients de manière automatique et interactive.

Dans un premier lieu , nous examinerons en détail les besoins des clients en matière de service client. Nous réaliserons des enquêtes et des entretiens pour comprendre les attentes des clients en termes de rapidité, d'accessibilité et de pertinence des réponses. Nous identifierons également les problèmes les plus couramment rencontrés par les clients et les types de demandes auxquelles le chatbot devra répondre. Cette analyse des besoins clients nous permettra de définir les fonctionnalités clés et les objectifs du chatbot.

Dans la 2eme section, nous aborderons la conception et la modélisation du chatbot de service client. Nous décrirons l'architecture du système, en mettant en évidence les différents composants tels que le moteur de traitement du langage naturel, la base de connaissances et l'interface utilisateur. Nous discuterons également des choix technologiques, tels que l'utilisation de l'apprentissage automatique et des techniques d'intelligence artificielle pour améliorer les performances du chatbot. Nous élaborerons également des scénarios d'utilisation et des flux de conversation pour assurer une interaction fluide entre le chatbot et les clients.

Enfin nous détaillerons les étapes de réalisation et de mise en œuvre du chatbot de service client. Nous expliquerons comment nous avons développé le moteur de traitement du langage naturel, comment nous avons constitué la base de connaissances et comment nous avons formé le chatbot à partir de données existantes et en utilisant des techniques d'apprentissage automatique. Nous présenterons également les défis techniques rencontrés et les solutions mises en place. Enfin, nous discuterons de l'intégration du chatbot dans les canaux de communication existants, tels que les sites web, les applications mobiles et les plateformes de messagerie.

# 1 Chapitre 1 : besoins client du projet :

## 1.1 Définition d'un chatbot :

Un chatbot, également appelé agent conversationnel, est un programme informatique conçu pour simuler une conversation humaine. Il utilise des techniques d'intelligence artificielle (IA) et de traitement du langage naturel (NLP) pour interagir avec les utilisateurs de manière automatisée. Les chatbots sont souvent utilisés dans le domaine du service client pour répondre aux questions des clients, fournir des informations, effectuer des transactions et résoudre des problèmes courants.

Cependant, il faut savoir qu'il existe plusieurs types de Chatbots :

- **Ceux qui sont très simples**, avec une capacité de compréhension faible, mais qui permettent de gérer des interactions simples.

Ces Chatbots fonctionnent sur le principe de mots clés, et de scénarios de conversations très scriptés (le but est de guider les utilisateurs).

Par exemple, si vous utilisez Chatfuel.com et qu'un client a plusieurs manières de poser une même question, vous devrez apprendre au bot la plupart (heureusement pas 100% !) des moyens de faire une demande.

- **Ceux qui permettent des interactions plus riches**, via un meilleur moteur d'analyse sémantique (compréhension des demandes des clients via le contexte et les synonymes, et non pas une analyse par mots clés), et la possibilité de lancer des actions (ex: passer une commande, gérer un bon de retour...).

## **1.2 Pourquoi utiliser un Chatbot dans le Service Client d'un restaurant?**

L'utilisation d'un chatbot dans le service client d'un restaurant présente de nombreux avantages significatifs. Voici quelques raisons pour lesquelles un chatbot peut être bénéfique dans ce contexte :

**Disponibilité 24/7 :** Un chatbot peut être disponible pour les clients à tout moment, que ce soit en dehors des heures d'ouverture du restaurant ou pendant les périodes de forte affluence. Cela permet aux clients d'obtenir rapidement des réponses à leurs questions ou de recevoir une assistance, améliorant ainsi leur satisfaction et leur expérience globale.

**Gestion efficace des réservations :** Un chatbot peut être intégré au système de réservation du restaurant, permettant aux clients de vérifier la disponibilité des tables, de réserver une table et même d'annuler ou de modifier une réservation existante. Cela facilite le processus de réservation, réduit les erreurs et libère le personnel du restaurant pour d'autres tâches.

**Réponses instantanées aux questions fréquentes :** Les chatbots sont capables de répondre rapidement aux questions fréquemment posées par les clients, comme les horaires d'ouverture, les options végétariennes, les informations sur les menus, les prix, etc. Cela permet d'économiser du temps et de l'effort à la fois pour les clients et le personnel du restaurant.

**Personnalisation de l'expérience client :** Un chatbot peut recueillir des informations sur les préférences des clients, tels que les allergies alimentaires, les préférences diététiques ou les plats préférés. En utilisant ces informations, le chatbot peut recommander des plats ou des boissons adaptées aux préférences spécifiques de chaque client, offrant ainsi une expérience personnalisée.

## **1.3 Description du besoin client :**

Nous disposons de 3 restaurants : KFC, McDonalds et Burger King. Chaque restaurant a son propre site qui montre des informations comme le menu, l'horaire, localisation du restaurant et des formulaires pour effectuer des réservations et des commandes.



Figure 1 : : Logos des restaurants kfc McDonalds et Burger King.

**La problématique à résoudre est la suivante :**

Les restaurants KFC, McDonald's et Burger King sont confrontés à des défis dans la gestion des réservations et des commandes, ainsi que dans la fourniture d'informations précises à leurs clients. Les processus traditionnels peuvent être lents, sujets aux erreurs et ne sont pas toujours disponibles en dehors des heures d'ouverture. Il est donc essentiel de trouver une solution efficace pour optimiser ces processus, offrir une expérience client améliorée, réduire les délais d'attente, automatiser les tâches répétitives et fournir des informations à jour, tout en libérant le personnel pour des tâches à plus forte valeur ajoutée.

C'est dans ce contexte qu'un chatbot de gestion des réservations et des commandes apparaît comme une solution prometteuse, permettant une disponibilité 24/7, une interaction conviviale, une personnalisation des recommandations et une fourniture d'informations précises et actualisées, contribuant ainsi à renforcer la satisfaction des clients et à optimiser les opérations des restaurants.

## **1.4 *Elaboration du cahier de charge :***

Le cahier des charges du projet de chatbot de service client de restauration vise à définir les objectifs, les fonctionnalités et les exigences techniques du chatbot. L'objectif principal est d'améliorer l'expérience client en offrant un service client rapide et efficace, facilitant ainsi la gestion des réservations et des demandes d'information sur les restaurants. Le chatbot doit être capable de fournir des réponses personnalisées aux questions des clients, de gérer les

commentaires et les plaintes, et d'optimiser les coûts opérationnels en automatisant certaines tâches du service client.

## **1.5 Besoins fonctionnels:**

- Le chatbot doit commencer avec un message lors de l'ouverture de l'application de la part du client.
- Le chatbot doit fournir les informations de chaque restaurant (menu, horaire, localisation) quand le client lui y demande.
- Chatbot va enregistrer les informations des réservations et des commandes dans la base de donnée.
- Chaque réservation est définie par : nom, prénom, email et numéro de téléphone du client, le jour de la réservation, heure de la réservation, nombre de personnes et le nom de restaurant.
- Chaque commande est définie aussi par les informations du client , la quantité à commander, et le type de livraison ( sur place ou emporté).
- Si un message du client est mal formulé , chatbot doit envoyer un message d'erreur et demander a nouveau de saisir le message.

## **1.6 Besoins techniques:**

- Le projet va être développé avec python et le Framework flask qui va comporter un réseau neuronal utilisant des couches denses pour l'apprentissage profond.
- Le model est construit en utilisant la bibliothèque Keras qui est une interface conviviale pour créer et entraîner des réseaux de neurones.
- L'utilisation d'un fichier JSON afin de fournir les données d'entraînement au modèle de réseau neuronal.
- Utilisation d'une base de donnée relationnel avec MySQL afin de stocker les informations des réservations et des commandes.
- Création d'une interface utilisateur qui va contenir un site web de restauration avec l'application chatbot situé en bas à droite convenable afin d'afficher les données envoyées par le back end.



## 2 Chapitre 2 : conception et modélisation du projet :

### 2.1 Langage de modélisation:

Dans le cadre de notre projet nous avons choisi UML pour une modélisation Objet UML : C'est un langage de modélisation objet convient pour toutes les méthodes objets et il est dans le domaine public. Ses objectifs :

- Fournir un langage visuel et expressif prêt à être utiliser pour le développement et l'échange de spécifications.
- Supporter des spécifications indépendantes d'un langage de programmation ou d'une méthode.
- Encourager le développement du marché des outils orientés objet.
- Supporte les concepts de haut niveau tels que les composants, les collaborations, les Framework...

### 2.2 Identification des acteurs et leurs rôles :

**Client :** le client présente l'acteur principale du projet , c'est lui qui va interagir avec l'application et pouvoir recevoir des réponses de la part du chabot.

**Chatbot :** un acteur principal qui présente l'interface entre le client et le système.

**Système de gestion des restaurant :** un acteur secondaire qui va récupérer les informations des réservations et des commandes afin de les traiter.

## 2.3 Modélisation:

### 2.3.1 Diagramme de cas d'utilisation :

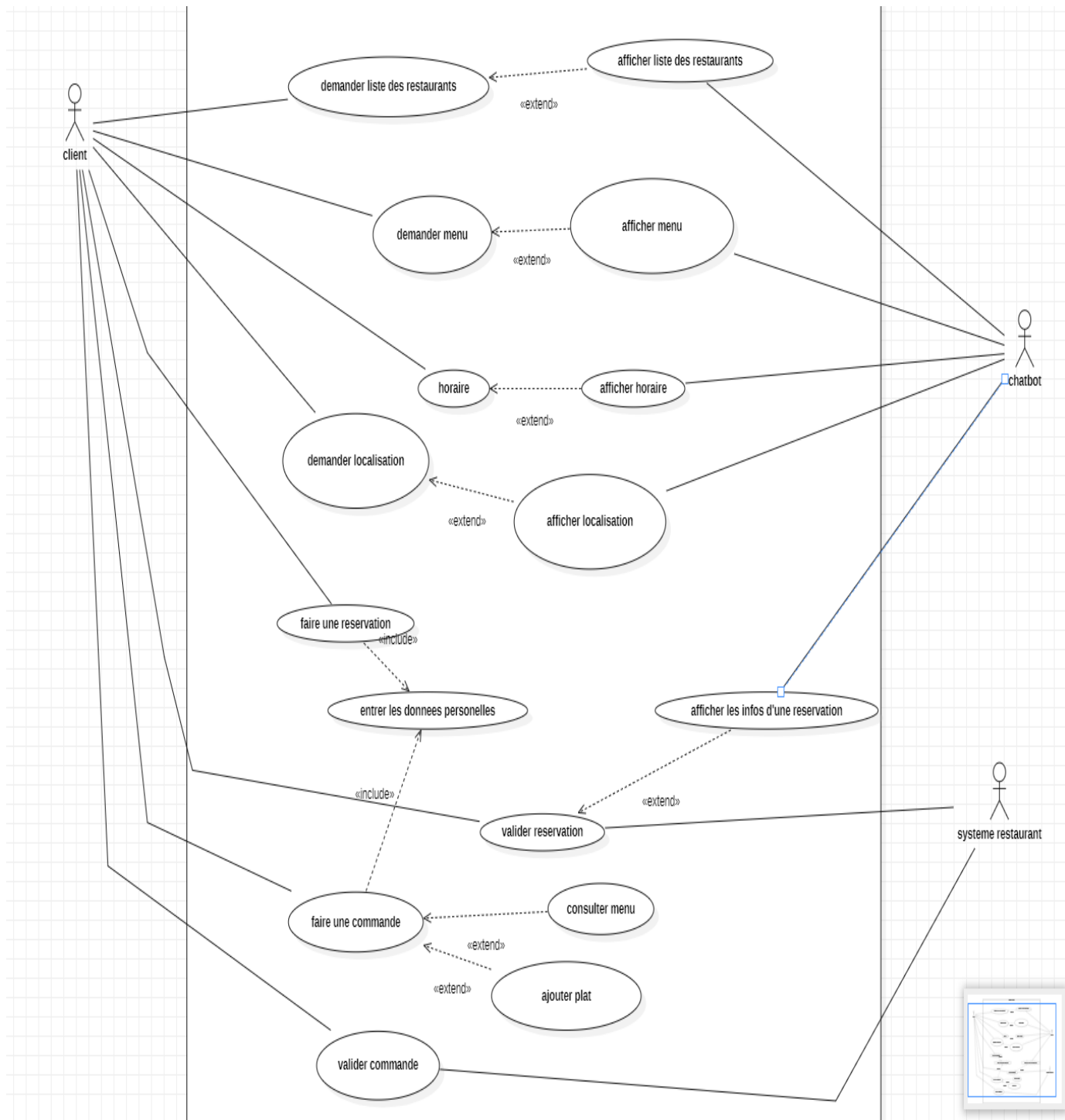


Figure 2: Diagramme cas d'utilisation representant les acteurs et leurs interactions.

### **Actions de l'acteur client :**

- **Donner la liste des restaurant** : le client va demander la liste des restaurants qui existent.
- **Demander menu** : le client va demander le menu du restaurant mentionné.
- **Horaire** : le client va demander l'horaire du restaurant mentionné.
- **Demander localisation** : le client va demander la localisation du restaurant mentionné.
- **Faire une réservation** : le client demande d'effectuer une réservation et cela nécessite obligatoirement de saisir ses informations personnelles c'est pour cela on a fait <<include>>
- **Valider réservation** : après avoir saisir les informations le client valide la réservation et ses informations seront envoyées vers l'acteur secondaire 'système restaurant' qui va traiter sa réservation.
- **Faire une commande** : le client va effectuer une commande et c'est obligatoire encore une fois de faire saisir ses informations personnelles grâce à la section : <<include>> mais il n'est pas nécessaire de consulter le menu et d'ajouter un plat.
- **valider commande** : après avoir saisi les informations de la commande le client la valide et les informations seront envoyées vers le système de gestion des restaurants qui va les traiter.

### **Actions de l'acteur Chatbot :**

- **afficher liste des restaurants** : le Chatbot va afficher les restaurants qui existent une lorsqu'il reçoit une demande de la part du client.
- **Afficher menu** : Chatbot va afficher le menu une fois qu'il reçoit une demande de la part du client.
- **Afficher horaire** : Chatbot va afficher l'horaire du restaurant une fois qu'il reçoit un message d'afficher l'horaire.
- **Afficher infos d'une réservation** : Chatbot va afficher les informations saisies par le client après avoir validé la réservation.

### **Actions de l'acteur 'système restaurant' :**

- **Valider réservation** : récupérer les informations de la réservation validé par le client de la base de données du système afin de les traiter.
- **Valider commande** : récupérer les informations de la commande validé par le client de la base de données du système afin de les traiter.

### 2.3.2 Diagramme de classe :

Dans le diagramme de classe on va modéliser le schéma de la base de donnée de notre système et identifier les méthodes qu'on va utiliser dans la partie de réalisation :

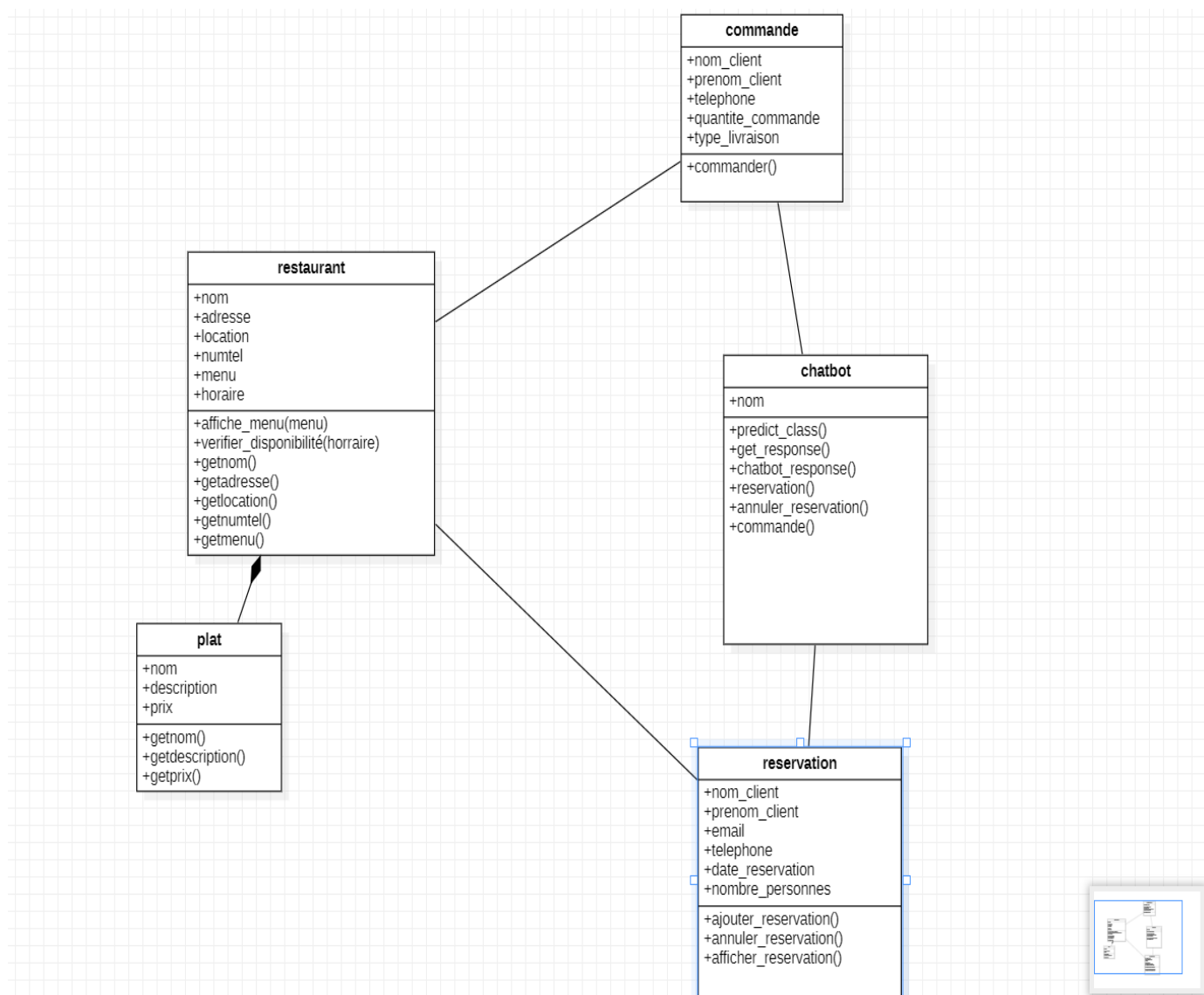


Figure 3: Diagramme de classe modélisant les différentes classes

### La classe commande :

- ❖ **attributs :** nom\_client : le nom du client (varchar) .  
Prenom\_client : le prénom du client (varchar).  
Téléphone : numéro de telephone du client (integer) .  
Quantité\_commande : la quantité du plat a commander (varchar).  
Type\_livraison : type de livraison soit sûr place ou emporté (varchar).
- ❖ **Méthodes :** commander() : ajouter une commande d'un client

### la classe réservation :

- **attributs :** nom\_client : le nom du client (varchar).  
Prenom\_client : le prénom du client (varchar) .  
Email : email du client (vatchar) .  
Téléphone : numéro de téléphone du client (integer).  
Date\_reservation : le jour de la réservation (date).
- **Méthodes :** ajouter\_reservation() : faire une réservation.  
Afficher\_reservation() : afficher les informations d'une réservation après  
La valider de la part du client.  
Annuler\_reservation() : annuler une reservation.

### La classe restaurant :

- **Attributs :** nom (varchar) , adresse (varchar) , location (varchar), numtel (int) , menu (varchar) , horaire (varchar) .

La classe plat est lié à la classe restaurant par composition car on possède plusieurs pour un seul restaurant

### La classe chatbot :

- **attributs :**  
nom : on donne un nom au chatbot.
- **Méthodes :**  
predict\_class() : cette méthode identifie les mots clés du message du client afin de détecter la réponse qu'on souhaite obtenir.

`get_response()` : la méthode qui permet d'obtenir la réponse du chatbot en fonction de l'entrée du client.

`Chatbot_response()` : la méthode qui permet d'envoyer la réponse approprié au client.

`Réservation()` : la méthode qui permet d'envoyer le formulaire de reservation.

`Annuler_reservation()` : méthode qui permet d'envoyer le formulaire d'annuler la reservation.

`Commande()` : méthode qui permet d'envoyer le formulaire de la commande.

### ***A noter :***

les méthodes `reservation()` , `annuler_réservation()` et `commande()` de la classe `chatbot` sont différentes de celle des classes `reservation` et `commande` puisque les premiers interagissent avec l'interface client et les autres appellent à la base de donnée afin de stocker les informations du client.

On peut déclarer une classe client comme une classe abstraite mais c'est facultatif parce qu'on va gérer des réservations et des commandes pas des clients comme on a noter dans le cahier de charge.

### ***Les règles de gestion :***

- Un restaurant peut recevoir plusieurs réservations.
- Un restaurant peut recevoir plusieurs commandes.
- Une reservation est associé à un restaurant.
- Une commande est associée à un seul restaurant.
- Un restaurant peut contenir plusieurs plats
- Un plat appartient à un seul restaurant.

### 2.3.3 Diagramme de séquence :

Le diagramme de séquence en UML est utilisé pour représenter l'interaction entre les objets d'un système dans un ordre chronologique. Il illustre comment les messages sont échangés entre les objets pendant l'exécution d'un scénario ou d'une fonctionnalité spécifique.

Voici les éléments clés d'un diagramme de séquence en UML :

**Objets/participants :** Les objets ou les acteurs qui interagissent dans le système sont représentés par des boîtes horizontales avec leur nom en haut.

**Ligne de vie :** Une ligne verticale associée à chaque objet/participant, qui représente la durée de vie de cet objet dans le système.

**Messages :** Les messages échangés entre les objets sont représentés par des flèches dirigées avec des étiquettes indiquant la nature du message. Ils peuvent être synchrones (avec une attente de réponse) ou asynchrones (sans attente de réponse).

**Activation :** Une barre rectangulaire sur la ligne de vie d'un objet indique la période d'activité de cet objet pendant le traitement d'un message.

**Fragments :** Des fragments peuvent être utilisés pour représenter des conditions, des boucles ou des alternatives dans le flux d'exécution.

**Retours :** Les retours de messages ou les réponses sont indiqués par des flèches pointillées ou des flèches de retour.

Le diagramme de séquence permet de visualiser et de comprendre l'ordre chronologique des interactions entre les objets et de modéliser le comportement dynamique d'un système. Il est souvent utilisé pour représenter des scénarios complexes, des fonctionnalités ou des processus métier.

Dans le cas de notre projet le diagramme de sequence serait comme le suivant :

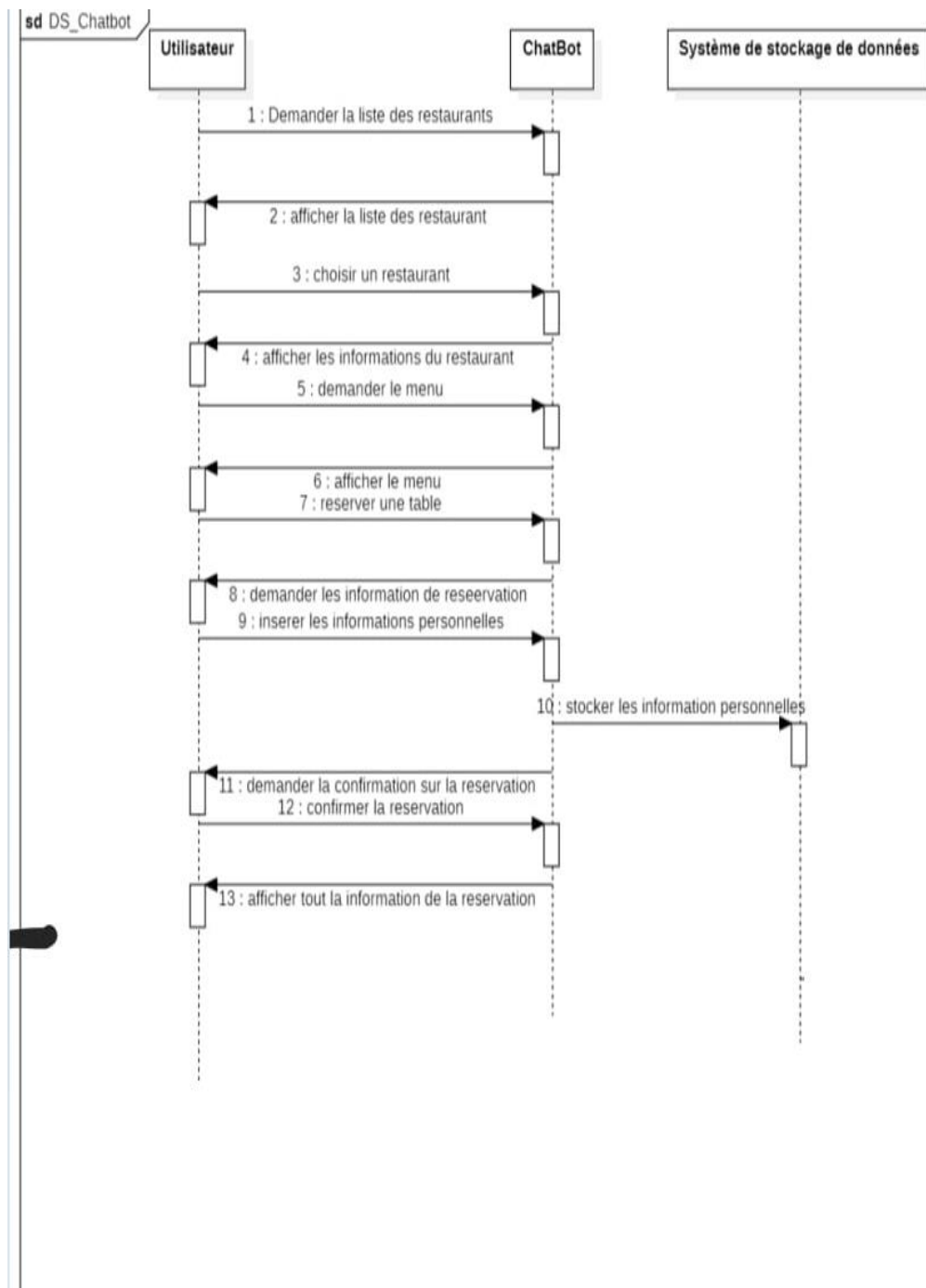


Figure 4: Diagramme de séquence.



## 3 Chapitre 3 : réalisation et développement :

### 3.1 Introduction :

Dans les chapitres précédents, nous avons essayé de suivre un enchaînement logique qui nous a permis de développer notre application. Nous arrivons maintenant à la phase de développement qui constitue la phase d'achèvement et d'aboutissement du projet. Pour cela, nous présentons, en premier lieu, l'environnement du travail, les outils et les langages du développement utilisés. En second lieu, nous proposons quelques imprimés écrans, et nous finissons par élaborer les tests fonctionnels de notre application.

### 3.2 Outils logiciels :

- **Python :** Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens.
- **Framework flask :** Flask est un Framework web minimaliste et léger en Python. Il est conçu pour faciliter la création rapide d'applications web avec une approche simple et flexible. Flask est souvent qualifié de "micro-Framework" car il ne nécessite pas de dépendances ou de composants externes importants. Cela le rend facile à utiliser et à comprendre, ce qui en fait un excellent choix pour les débutants en développement web.

L'architecture d'un projet flask est comme le suivant :

1. **Dossier racine :** Le dossier racine du projet Flask contient tous les fichiers et dossiers liés à l'application. Vous pouvez lui donner le nom de votre choix.
2. **Fichier principal de l'application :** Le fichier principal de l'application est généralement nommé app.py ou main.py. C'est là que vous initialisez l'application Flask, configurez les routes, les extensions et d'autres configurations spécifiques à l'application.

3. **Dossiers statiques** : Le dossier "static" contient les fichiers statiques tels que les fichiers CSS, JavaScript, images, etc., utilisés dans l'application.
4. **Dossiers templates** : Le dossier "templates" contient les templates Jinja2, qui sont les fichiers HTML avec leurs styles css et JS utilisés pour générer les pages web de l'application. Ces templates peuvent inclure des variables et des blocs de code pour rendre les pages dynamiques.

on note qu'on doit ajouter un autre fichier python : 'training.py' qui présente le réseau neuronal de notre application.

- **Phpmyadmin** : phpMyAdmin est une application web open source écrite en PHP qui permet de gérer et d'administrer des bases de données MySQL. Elle fournit une interface conviviale basée sur le web qui permet aux utilisateurs de créer, modifier, supprimer et gérer les tables, les requêtes SQL, les utilisateurs, les privilèges et autres objets de base de données.

### 3.3 Travail réalisé :

#### Etapes de construction de l'application :

- **Création de l'application chatbot.**
- **Création du site web ou on va intégrer notre chatbot.**
- **Lier chatbot au site web grâce à une API ou bien inclure le code html dans le site et modifier le css afin d'éviter les conflits entre les styles ccs.**

### 3.3.1 Les interfaces graphiques :

#### 3.3.1.1 Interface du site web :

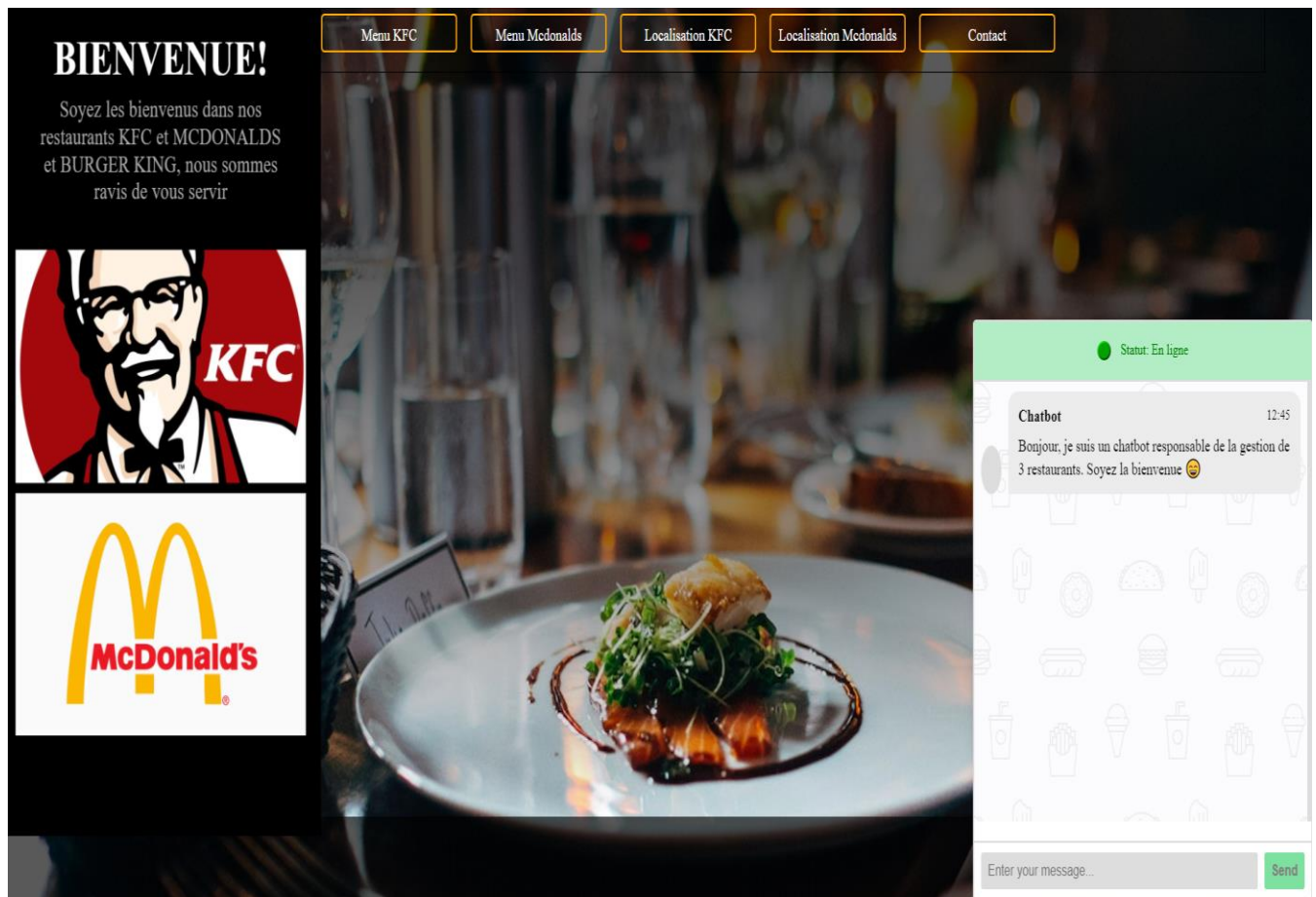


Figure 5: Interface du site réalisé

La figure ci-dessus présente la totalité du site web avec le chatbot situé en bas à droite. À gauche, nous avons les logos des restaurants et dans la barre, nous avons les informations à propos de chaque restaurant.

#### 3.3.1.2 Interface de la demande la listes des restaurants :

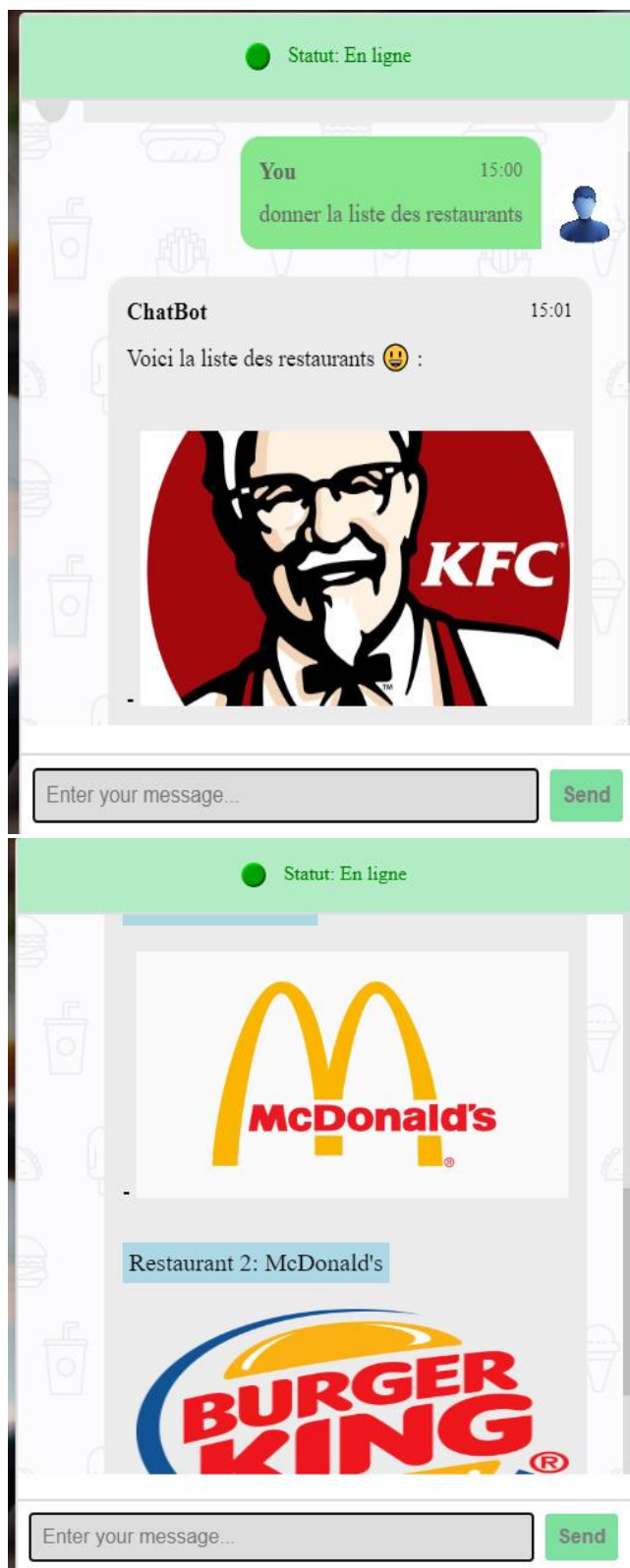


Figure 6: Interface du site réalisé

Dans ces 2 figures le client demande la liste des restaurants qui existent puis le chatbot lui répond avec les images sous forme des boutons qui incluent le lien URL de chaqu'un.

### 3.3.1.3 Interface de la demande du menu :

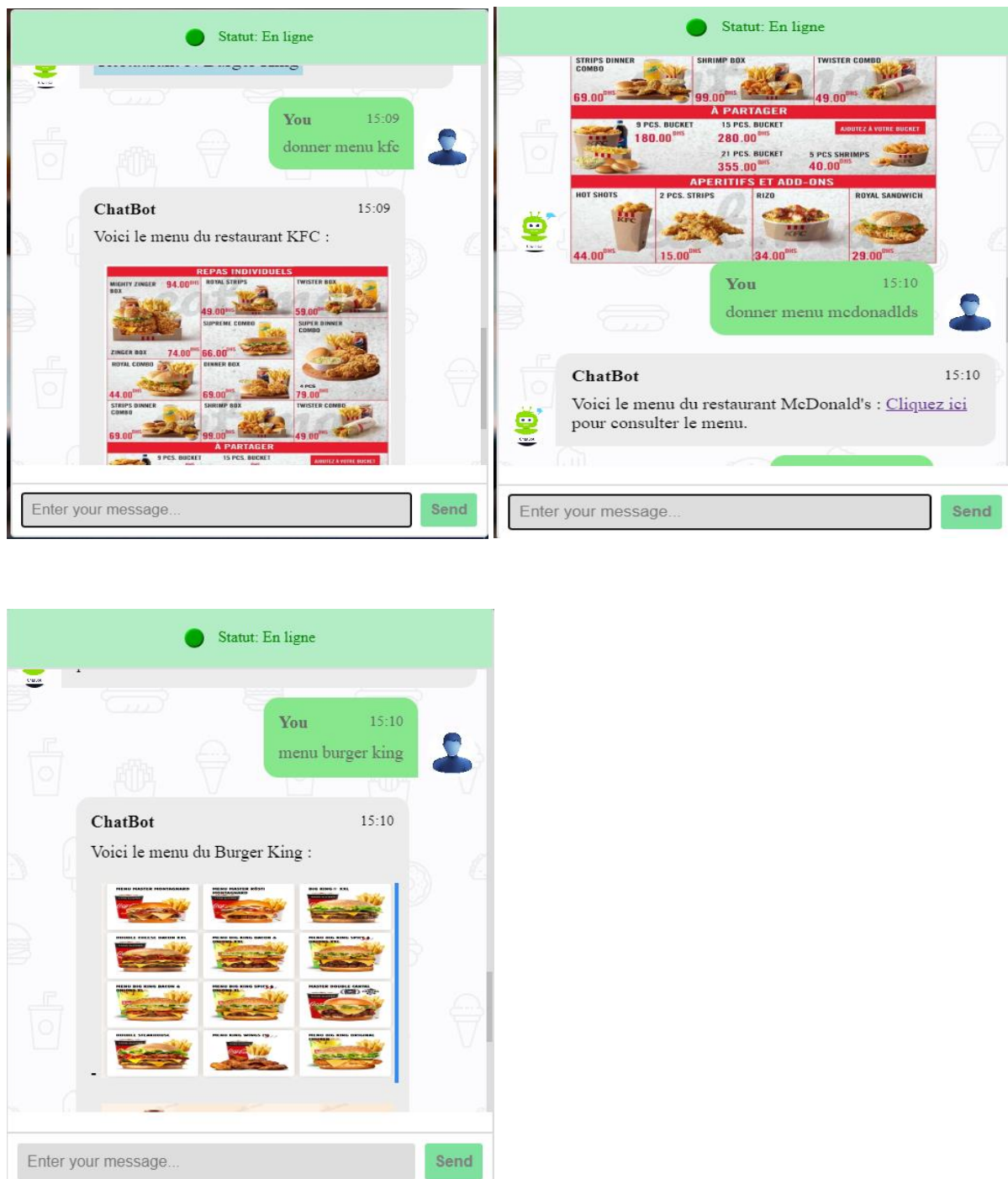
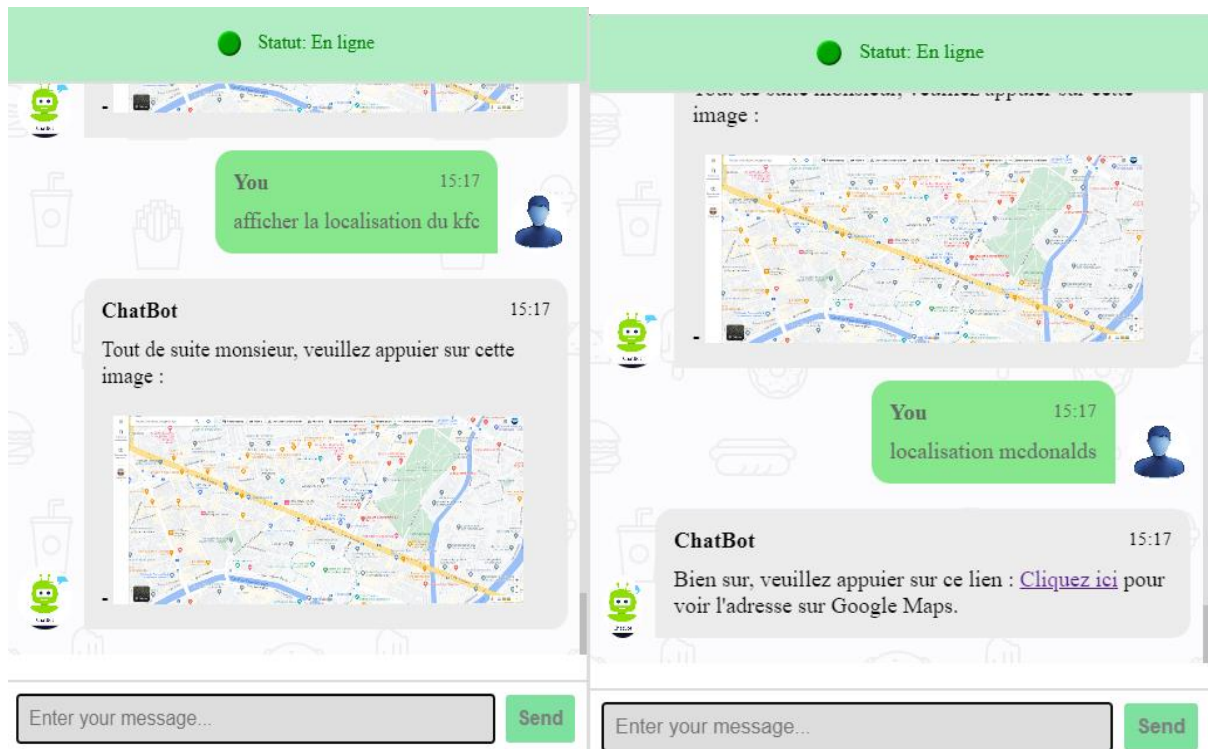


Figure 7: Interface des menus.

Dans ces 3 figures le client demande le menu de chaque restaurant puis chatbot lui répond avec le lien image de chaqu'un qui va envoyer vers un autre lien URL pour visualiser le menu.

### **3.3.1.4 Interface de la demande de la localisation :**



*Figure 8: Interface des localisations.*



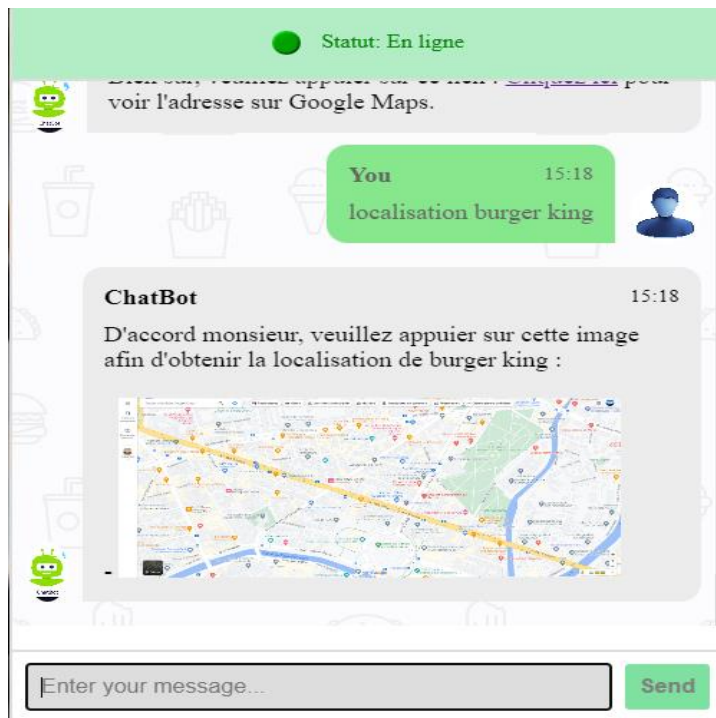


Figure 9: Interface de localisation de burger King.

Dans ces 3 figures chatbot affiche la localisation de chaque restaurant puis envoie le client vers un lien URL de google maps afin de savoir où se situe chaque restaurant.

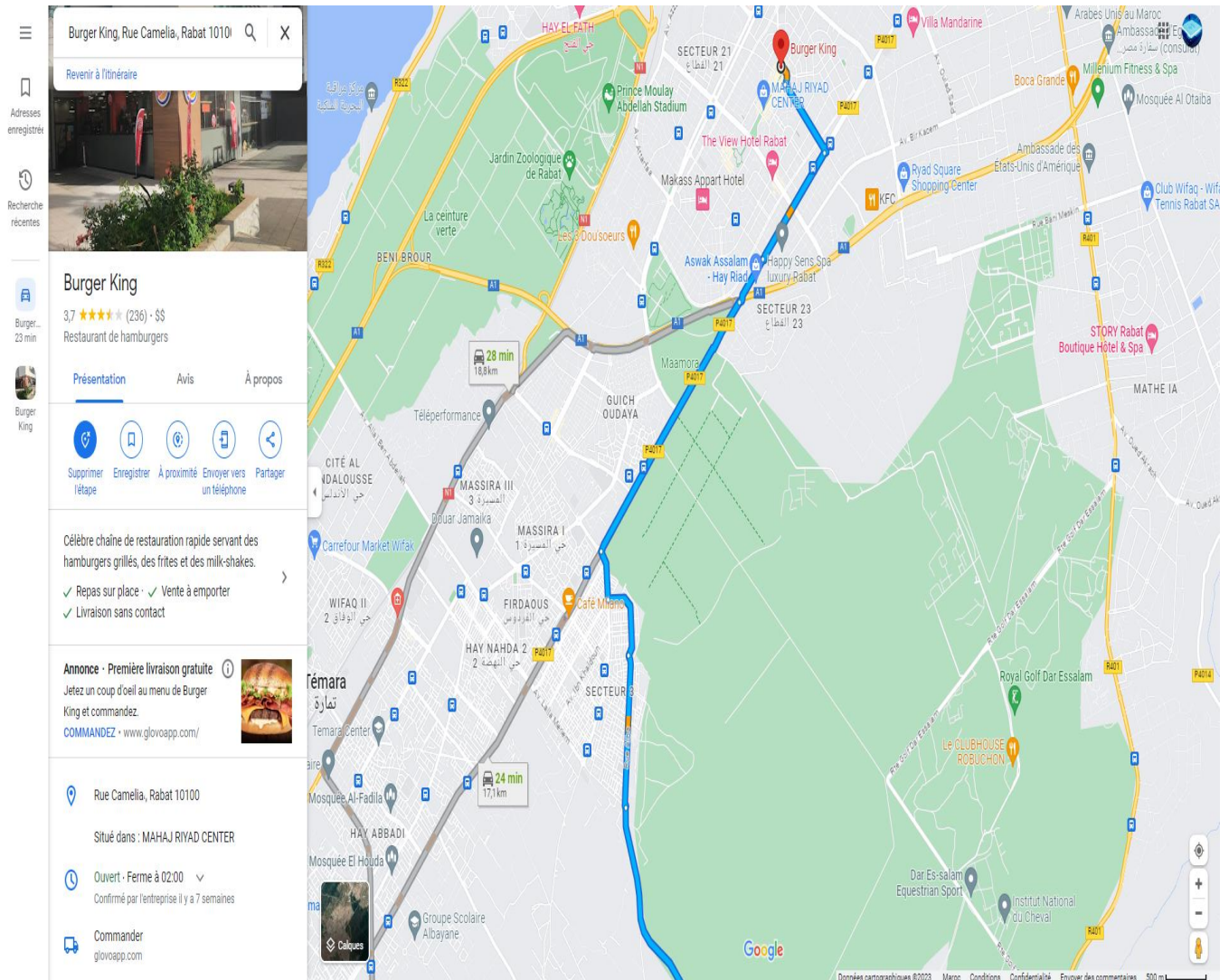


Figure 10: Interface de localisation sur Google maps

### 3.3.1.5 Interface de la demande de l'horaire :

**Le client demande l'horaire de chaque restaurant puis chatbot lui affiche la réponse.**



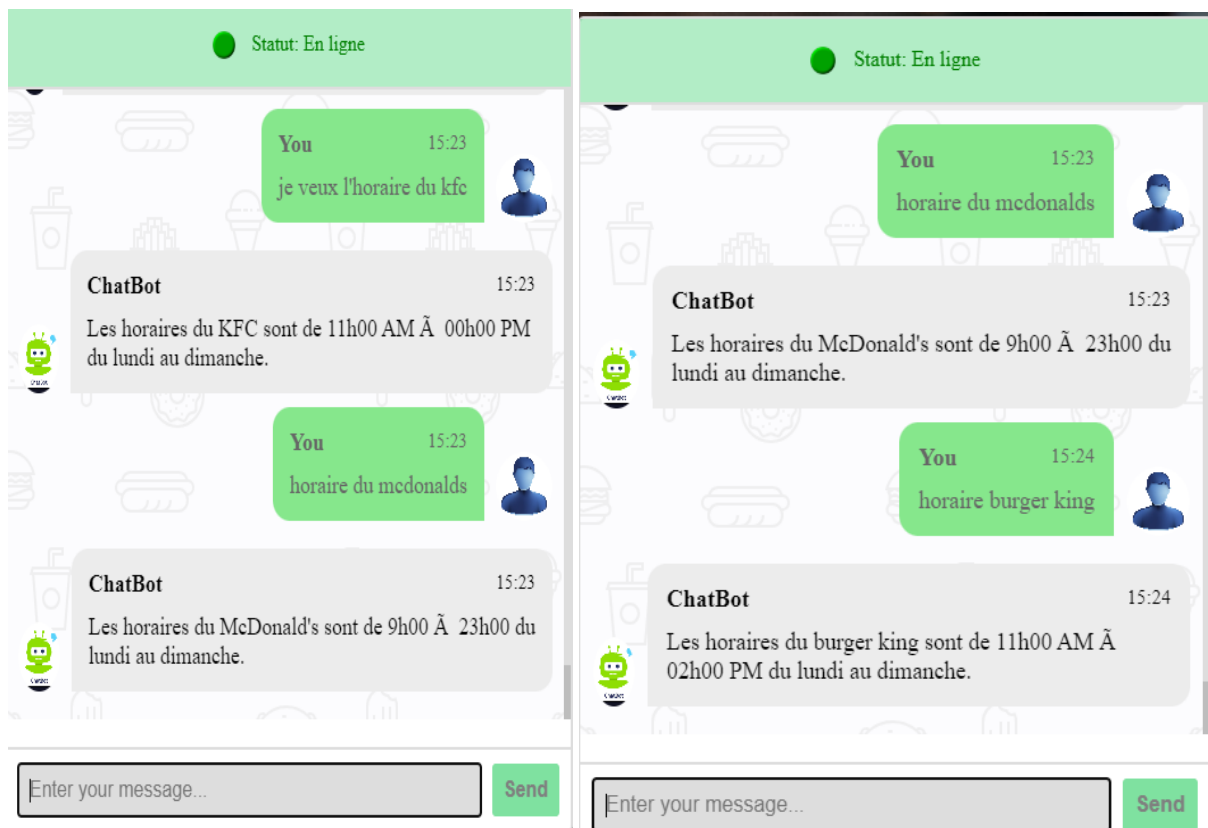


Figure 11: Interface de l'horaire

### 3.3.1.6 Interface de la demande d'une reservation :

The image displays two side-by-side screenshots of a chatbot interface. Both screenshots have a green header bar with a status indicator (a green dot) and the text "Statut: En ligne". The background of the chat area is decorated with faint icons of food items like burgers, fries, and drinks.

**Left Screenshot:**

- User:** "je veux reserver" (15:26)
- ChatBot:** "Réservation" (15:26)

The chatbot's response is a form titled "Réservation" with the following fields:

- Nom:
- Prénom:
- Email:
- Numéro de téléphone:

**Right Screenshot:**

- ChatBot:** "Numéro de téléphone:
- ChatBot:** "Jour de réservation:
- ChatBot:** "Heure:
- ChatBot:** "Nombre de personnes:
- ChatBot:** "choisis le restaurant (noter 1 pour kfc ,2pour mcdonalds et 3 pour burger king):
- ChatBot:** "Réserver" (button)

Both screenshots feature a text input field at the bottom labeled "Enter your message..." and a green "Send" button.

Figure 12: Interface de reservation

ici le client demande de réserver mais pas précisément dans un restaurant , le chatbot lui envoie le formulaire de reservation puis a la fin il demande au client de saisir le restaurant dans lequel il veut réserver. A la fin le client valide sa reservation puis il s'affiche un formulaire que la reservation a été enregistrée.

**Réservation enregistrée ✓**

Votre réservation a été enregistrée avec succès. Merci de nous avoir choisis !

**Informations de réservation :**

Nom : ahmed  
 Prénom : lahbabl  
 Email : lahmedx@hotmail.ca  
 Numéro de téléphone : 2124341431  
 Date de réservation : 13/23/2021  
 Heure de réservation : 12 PM  
 Nombre de personnes : 2  
 ID du restaurant : 1

[Retour au chatbot](#)

Figure 13: Interface de reservation bien enregistré.

### 3.3.1.7 Interface de l'annulation d'une reservation :

Avec la meme logique de la reservation chatbot envoie un autre formulaire d'annulation de la reservation, le client saisie quelque information puis si la reservation est trouvé chatbot envoie un formulaire 'Votre réservation a été annulée avec succès.' Sinon el envoie un formulaire d'erreur.

Statut: En ligne

You 15:36  
annuler reservation

ChatBot 15:36

**Annuler Réservation**

Nom:

Prénom:

Email:

Numéro de téléphone:

[Annuler](#)

Enter your message... [Send](#)

Figure 14: Interface de l'annulation de la réservation



Figure 15: Interface si la reservation est annulé

#### 3.3.1.8 Interface de la demande d'effectuer une commande :

Chatbot va envoyer une liste des plats quand il va recevoir une demande de la part du client de faire une commande puis le client fait son choix du plat a commander.

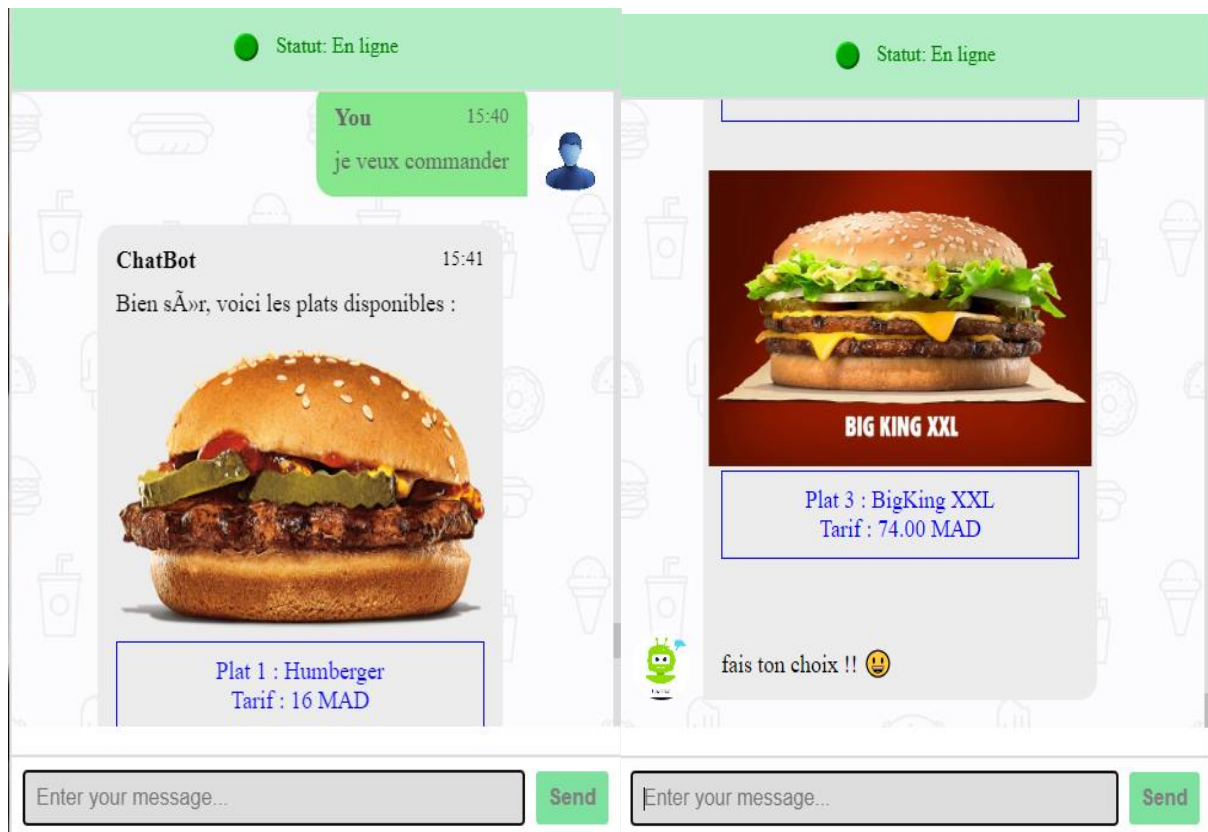


Figure 16: Interface des plats proposés

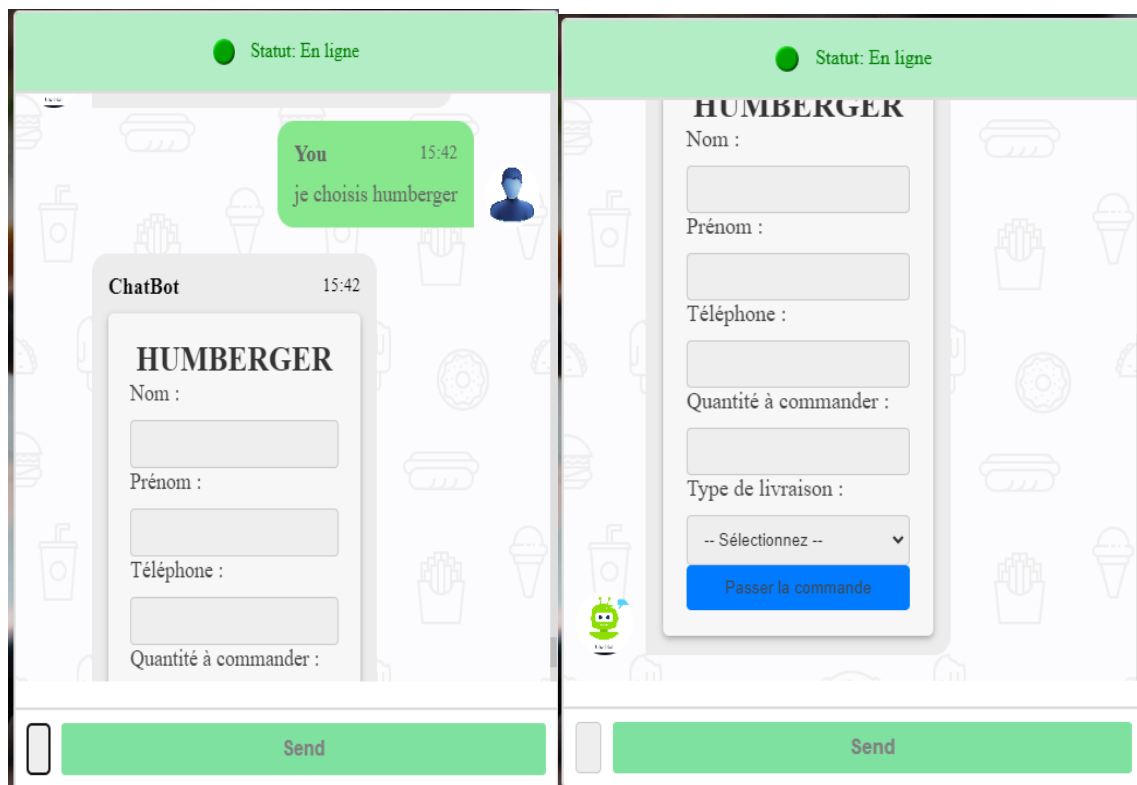


Figure 17: Interface du formulaire de la commande

Après avoir rempli les informations le client valide son choix puis chatbot envoie un formulaire de confirmation.



*Figure 18: Interface de l'enregistrement de la commande.*

## *Conclusion générale :*

Le projet de chatbot de restaurant est une solution novatrice pour offrir un service client personnalisé et efficace aux clients dans l'industrie de la restauration. Ce chatbot, spécialement conçu pour répondre aux besoins des clients dans ce domaine, permet de faciliter la prise de commandes, de répondre aux questions fréquentes, de fournir des informations sur les menus et les offres spéciales, ainsi que de recueillir les commentaires des clients.

Le chatbot de restaurant est un outil polyvalent qui peut être intégré sur différentes plateformes de communication, telles que les sites web, les applications mobiles et les réseaux sociaux. Il est capable de comprendre les requêtes des clients en langage naturel et de leur fournir des réponses précises et pertinentes en temps réel.

L'un des avantages clés du chatbot de restaurant est sa disponibilité 24 heures sur 24 et 7 jours sur 7. Les clients peuvent interagir avec le chatbot à tout moment, que ce soit pour passer une commande tard le soir, obtenir des recommandations de menu ou simplement poser des questions sur les horaires d'ouverture. Cela permet d'améliorer l'accessibilité du service client et de renforcer la satisfaction des clients.

En conclusion, le chatbot de restaurant est un outil puissant qui révolutionne le service client dans l'industrie de la restauration. En combinant l'intelligence artificielle, la convivialité et la personnalisation, il offre une expérience client améliorée, renforce la fidélisation de la clientèle et contribue à la croissance des entreprises de restauration. Grâce à son accessibilité 24/7 et à sa capacité à répondre rapidement aux demandes des clients, le chatbot de restaurant joue un rôle essentiel dans la transformation numérique de ce secteur.

## *Bibliographie :*

1. Buff ML. "Web-Based Chatbot Using Flask API." [En ligne]. Disponible sur : <https://buffml.com/web-based-chatbot-using-flask-api/>
2. Maina Dennis. "An AI Chatbot in Python and Flask." [En ligne]. Disponible sur : <https://github.com/mainadennis/An-AI-Chatbot-in-Python-and-Flask>
3. Artificial Intelligence Hub. "Chaîne YouTube." [En ligne]. Disponible sur : [\[https://www.youtube.com/watch?v=VvOZjw0ceNw&t=4s&pp=ygUSY2hhdGJvdCB3aXRoIGZsYXNr\]](https://www.youtube.com/watch?v=VvOZjw0ceNw&t=4s&pp=ygUSY2hhdGJvdCB3aXRoIGZsYXNr)