

Noé Libon

Marina Quadu

Nathan Sancke-Squelin

Patson Tiako Ngouadje

29-10-2021

Cahier des charges

Module : Chatbot Externe

1. Présentation du client

L'EPHEC est une école supérieure située à Louvain-La-Neuve. Elle accueille à peu près 4000 étudiants regroupés dans 4 sections différentes. Elle dispose de plusieurs moyens de communications : téléphone, mail, Moodle, Teams, ...

2. Contexte

Compte tenu des nombreux moyens de communications mis à disposition de l'étudiant, l'EPHEC souhaiterait développer son propre canal de communication basé sur le modèle de l'application Discord. Une application personnelle à l'école lui permettrait d'être indépendante vis-à-vis des nombreuses autres structures existantes et d'avoir ainsi un contrôle total.

Pour ce faire, un projet noyau a été créé et plusieurs autres modules doivent être implémentés afin de faciliter la gestion et le confort du projet final.

Cela sera bénéfique tant aux administrateurs qu'aux élèves.

Le module Chatbot externe permettra à tous ses utilisateurs d'avoir une réponse rapide à des questions élémentaires telles que la recherche d'un resto ou la météo du lendemain.

3. Objectifs

Le module Chatbot externe doit pouvoir communiquer avec l'ordinateur dans un premier temps via une commande en console qui sera interprétée par l'ordinateur. L'utilisateur recevra une réponse adéquate. Les demandes concerneront divers domaines tels que la météo, une recherche d'itinéraire, les actualités, une recherche de centre d'intérêt...

4. Cible/périmètre

Ce projet s'adresse à toute personne évoluant au sein du site de l'Ephec à Louvain-La-Neuve.

5. Description fonctionnelle des besoins

- Définir une liste de requêtes pour l'utilisateur telles que la météo, les actualités... afin de pouvoir communiquer avec le bot :
 - Help : renvoie une liste de toutes les commandes disponibles
 - Lorsque l'on tape **/help** dans l'interface, la réponse nous renvoie la liste de toutes les commandes en précisant leur utilisation.
 - Météo : renvoie la température, la probabilité de pluie et les prévisions sur la journée de l'endroit précisé ;
 - Lorsque l'on tape **/weather** dans l'interface, la réponse renvoie les points cités ci-dessus de manière géolocalisée.
 - Un deuxième paramètre **ville** permettra d'affiner en choisissant la localité.
 - Un troisième paramètre **chiffre** (compris entre 1 et 7) renverra une prévision sur un ou plusieurs jours.
 - ⚠ Les paramètres ne doivent pas contenir d'espace.
 - Actualités : renvoie les actualités du jour d'un endroit prédéfini ;
 - Lorsque l'on tape **/news** dans l'interface, la réponse renvoie, au hasard, une proposition d'article de l'actualité nationale du jour.

- Un deuxième paramètre **pays** permettra d'avoir accès à un article de l'actualité du pays cité.
- Itinéraire : renvoie un lien Google Map du trajet demandé lorsque l'on tape **/itinerary** ;
 - Si un seul paramètre **adresse** est précisé, renvoie le trajet en fonction de la localisation de l'utilisateur.
 - Si deux paramètres **adresse** sont précisés, la réponse renvoie un itinéraire calculé à partir de la première adresse.

⚠ Les adresses doivent être sous forme de chaînes de caractères.
- Restaurant : renvoie une liste de restaurants les plus proches en fonction de la géolocalisation de l'utilisateur ;
 - Lorsque l'on tape **/resto** avec un paramètre optionnel **rayon**, la réponse renvoie tous les restaurants répertoriés dans le périmètre défini. Par défaut, le rayon est de 10km.
- Cinéma : renvoie une liste des cinémas les plus proches en fonction de la géolocalisation de l'utilisateur ;
 - Lorsque l'on tape **/cine** avec un paramètre optionnel **rayon**, la réponse renvoie tous les cinémas répertoriés dans le périmètre défini. Par défaut, le rayon est de 10km.
- Mesure de la satisfaction de l'utilisateur par retour d'expériences :
 - Une commande **/opinion** suivi du paramètre **positif** ou **négatif** permet d'avoir un retour sur l'utilisation du chatbot.
 - Un deuxième paramètre **commentaire**, optionnel, permet à l'utilisateur de commenter son avis sous forme d'une chaîne de caractères.

⚠ Les commentaires doivent être sous forme de chaînes de caractères.

6. Spécificités techniques/contraintes

Le client souhaite que le code soit écrit en python 3.8+ puisque c'est le langage avec lequel le code noyau a été écrit. Le code devra respecter les règles de la PEP 8.

Le module doit aussi s'intégrer le plus facilement au code noyau grâce aux fonctionnalités de git afin de fonctionner sur l'interface graphique.

L'organisation du travail se fera à l'aide d'un GitHub Project et la documentation du module sera reprise dans un wiki sur GitHub.

Les fonctionnalités **/itinerary**, **/resto** et **/cine** utiliseront l'API de google maps.

La fonctionnalité **/weather** utilisera l'API de weatherstack.

La fonctionnalité **/news** utilisera l'API de mediastack.

La fonctionnalité **/opinion** communiquera avec la base de données afin de stocker les commentaires des utilisateurs.

Les commandes ainsi que leurs paramètres seront stockées dans la base de données gérée avec MongoDB.

7. Charte graphique

Dans un premier temps, le module sera développé en console.

8. Budget

5 millions de dollars versés sur un compte aux îles Caïmans !

9. Consultations

Les consultations se feront de manière hebdomadaire. Il a été convenu avec le client que les entrevues auraient lieu le lundi.

10. Planning

Le module doit être fonctionnel pour la fin de cette année 2021.