**Documentație Arhitecturală Preliminară: Chatbot cu Procesare a Limbajului Natural**

**1. Descriere generală a proiectului**

Acest proiect își propune să dezvolte un chatbot capabil să răspundă la întrebări comune și să mențină conversații simple. Chatbotul va folosi tehnici de Procesare a Limbajului Natural (NLP) pentru a înțelege intențiile utilizatorului și pentru a genera răspunsuri relevante. Proiectul este implementat de două persoane, fiecare cu nivel de cunoștințe începător spre mediu în Python, utilizând framework-uri și biblioteci accesibile.

**2. Obiectivele proiectului**

* **Procesarea Limbajului Natural:** Implementarea unei componente NLP care să identifice intențiile utilizatorilor și să ofere răspunsuri adecvate, folosind biblioteci precum spaCy și NLTK.
* **API de chat și interfață de utilizator:** Crearea unui API prin Flask pentru gestionarea interacțiunilor și implementarea unei interfețe de chat pentru interacțiunea utilizatorului.

**3. Puncte principale de vizat**

* **NLP și Generarea Răspunsurilor**
  + Configurarea și procesarea datelor de intrare folosind tehnici NLP pentru a extrage intenția.
  + Crearea unui motor de generare a răspunsurilor, adaptat să răspundă la întrebări comune, folosind spaCy și NLTK.
  + Integrarea componentei NLP în API-ul de chat dezvoltat de Persoana 2.

**API-ul și Interfața de Chat**

* + Implementarea unui API folosind Flask pentru a facilita interacțiunea între utilizator și componenta NLP.
  + Dezvoltarea unei interfețe de chat simplificată, utilizând HTML/CSS și JavaScript pentru comunicarea cu API-ul.
  + Asigurarea gestionării sesiunilor de chat și colectarea intrărilor de la utilizator.

**4. Arhitectura Sistemului**

**4.1 Componente principale:**

1. **Frontend (Interfața de Chat)**
   * **Descriere:** Interfață cu utilizatorul, care oferă un câmp de text pentru input-ul utilizatorului și o zonă de afişare a răspunsurilor.
   * **Tehnologii:** HTML, CSS, JavaScript (AJAX pentru comunicare asincronă).
   * **Funcționalitate:** Trimite cererea utilizatorului către API și afișează răspunsurile primite.
2. **Backend (API de Chat)**
   * **Descriere:** API RESTful creat cu Flask care primește mesajele de la utilizator și returnează răspunsurile generate de componenta NLP.
   * **Endpoint principal:** /chat - primește mesaje printr-o cerere POST și returnează un răspuns.
   * **Tehnologii:** Flask pentru gestionarea cererilor și rutarea lor către modulul NLP.
3. **Componenta NLP (Procesare a Limbajului Natural)**
   * **Descriere:** Sistem de înțelegere a intențiilor utilizatorului și de generare a răspunsurilor folosind algoritmi NLP.
   * **Tehnologii:** spaCy pentru identificarea entităților și intențiilor, NLTK pentru tokenizare și eventuale reguli de generare a răspunsurilor.
   * **Funcționalitate:** Identifică intenția utilizatorului (de exemplu, "Ce faci?" -> întrebare despre starea chatbot-ului) și formulează un răspuns.

**4.2 Fluxul de date**

1. **Utilizatorul** introduce un mesaj în interfața de chat.
2. **Frontend-ul** trimite mesajul către API-ul Flask printr-o cerere POST la endpoint-ul /chat.
3. **API-ul Flask** trimite mesajul la componenta NLP pentru pre-procesare și înțelegerea intenției.
4. **Componenta NLP** procesează mesajul, identifică intenția și generează un răspuns.
5. **API-ul Flask** trimite răspunsul înapoi la interfața de chat.
6. **Frontend-ul** afișează răspunsul în interfața de chat.

**5. Descrierea Tehnică a Fiecărei Componente**

**5.1 Interfața de Chat (Frontend)**

* **Cerințe tehnice:** HTML simplu pentru structură, CSS pentru stilizare și JavaScript pentru comunicare AJAX.
* **Funcționalități de bază:**
  + Un câmp de input pentru mesajul utilizatorului.
  + O secțiune pentru afișarea dialogului între utilizator și chatbot.
  + Funcție de trimitere a mesajului către API și de afișare a răspunsului.

**5.2 API-ul de Chat (Backend)**

* **Cerințe tehnice:** Flask ca framework pentru API.
* **Funcționalități de bază:**
  + Endpoint-ul /chat primește mesajele prin POST și le trimite componentelor NLP.
  + Returnează răspunsul generat de componenta NLP către frontend.

**5.3 Componenta NLP**

* **Cerințe tehnice:** spaCy pentru recunoașterea entităților și clasificarea intențiilor, NLTK pentru tehnici de preprocesare a textului.
* **Funcționalități de bază:**
  + **Tokenizare:** Segmentarea textului în cuvinte folosind NLTK.
  + **Identificarea intenției:** Algoritm de clasificare a intențiilor pentru a înțelege mesajele utilizatorului (e.g., întrebări, răspunsuri, comenzi).
  + **Generarea răspunsurilor:** Răspunsuri bazate pe intențiile identificate, stocate într-o bază de date simplă sau hardcodată.

**6. Considerații privind Dezvoltarea și Testarea**

* **Dezvoltare incrementală:** Se recomandă dezvoltarea și testarea fiecărei componente separat înainte de integrare (de exemplu, testarea API-ului Flask cu mesaje de test, fără componenta NLP).
* **Testare unităților (Unit Testing):** Se vor implementa teste de unitate pentru funcțiile NLP și API.
* **Validarea răspunsurilor:** Se vor crea cazuri de test pentru a verifica dacă răspunsurile sunt relevante și corecte.

**7. Concluzii și Recomandări**

Acest proiect oferă o introducere practică în dezvoltarea unui chatbot simplu, cu o arhitectură modulară, utilizând NLP și un API RESTful. Recomandăm utilizarea tehnicilor de dezvoltare incrementală și teste unitare pentru a asigura funcționarea corectă a sistemului și a îmbunătăți acuratețea răspunsurilor.