

## Dokumentacja Inżynierii Wymagań

### Członkowie zespołu:

#### 1. Macierz kompetencji zespołu.

Kompetencje	Szymon	Maurycy	Mateusz	Patryk
Programowanie Python	+	+	+	+
Obsługa DataBase	+/-	+	+	-
Programowanie React	+	+	-	+/-
Web Development	+	+	+	+
Implementacja modeli AI	+	-	+	-
Testy Oprogramowania	-	-	+	+/-
Projektowanie Grafiki	-	+/-	-	-
Znajomość UML	+/-	-	+	+/-
Znajomość Docker	-	-	+/-	+

#### 2. Zestaw pytań, które zostały sformułowane w celu uszczegółowienia zadanego projektu.

Pytanie	Odpowiedź	Uwagi
Jak powinna wyglądać aplikacja?	Prosta strona internetowa z obsługą wymienionych interakcji.	Nie wymaga się logowania.
Czy powinna być interakcja pomiędzy użytkownikami?	Tylko interakcja z asystentem.	
Jak powinna wyglądać historia w aplikacji?	W historii powinno się znajdować 10 ostatnich rozmów.	Dotyczy aktualnej sesji użytkownika.

Ile języków powinna obsługiwać aplikacja?	Aplikacja powinna obsługiwać język angielski.	Opcjonalnie interfejs po polsku.
Jakich technologii użyć do stworzenia aplikacji?	Dobór technologii jest dowolny.	
Czy aplikacja powinna obsługiwać polecenia głosowe?	Tak – powinna umożliwiać rozmowę z asystentem głosowym.	
Czy aplikacja powinna umożliwiać analizę zdjęć?	Tak - pełna obsługa zdjęć.	
Czy jest jakiś określony wygląd aplikacji?		

Czy wszystkie wymagania klienta są możliwe do spełnienia? - Tak

### 3. Ustalony format danych wejściowych.

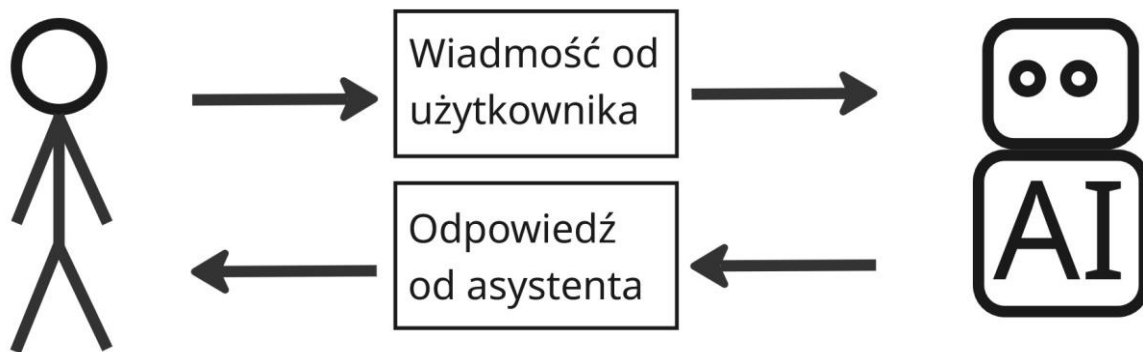
Wiadomość	Obraz	Mowa
ID (int unique const) Treść (string) Godzina(string)	ID (int unique const) Obraz (po konwersji string) Godzina(string)	ID (int unique const) Treść po speech-to-txt (string) Godzina(string)

Przykładowy format: 1,Hello,13:43:13

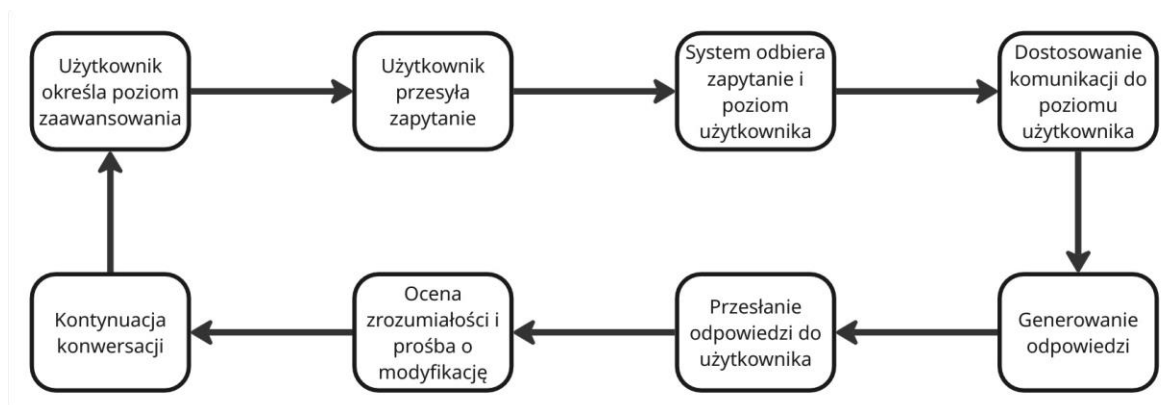
### 4.1 Modelowanie systemu za pomocą tabeli.

Aktorzy	Użytkownik, Asystent®
Opis	Celem działania asystenta będzie nauczanie użytkownika rozmowy w języku angielskim na wyższym poziomie
Dane	Prompt podany przez użytkownika (tekst, obraz, audio).
Wyzwalacz	Od razu po wystąpieniu wiadomości, chatbot rozpoczyna proces “myślenia” nad odpowiedzią.
Odpowiedź	Dostosowana do kontekstu, spersonalizowana wiadomość tekstowa.
Uwagi	System jest ograniczony inteligencją użytego modelu językowego

#### 4.2 Przedstawienie systemu za pomocą diagramu UML.

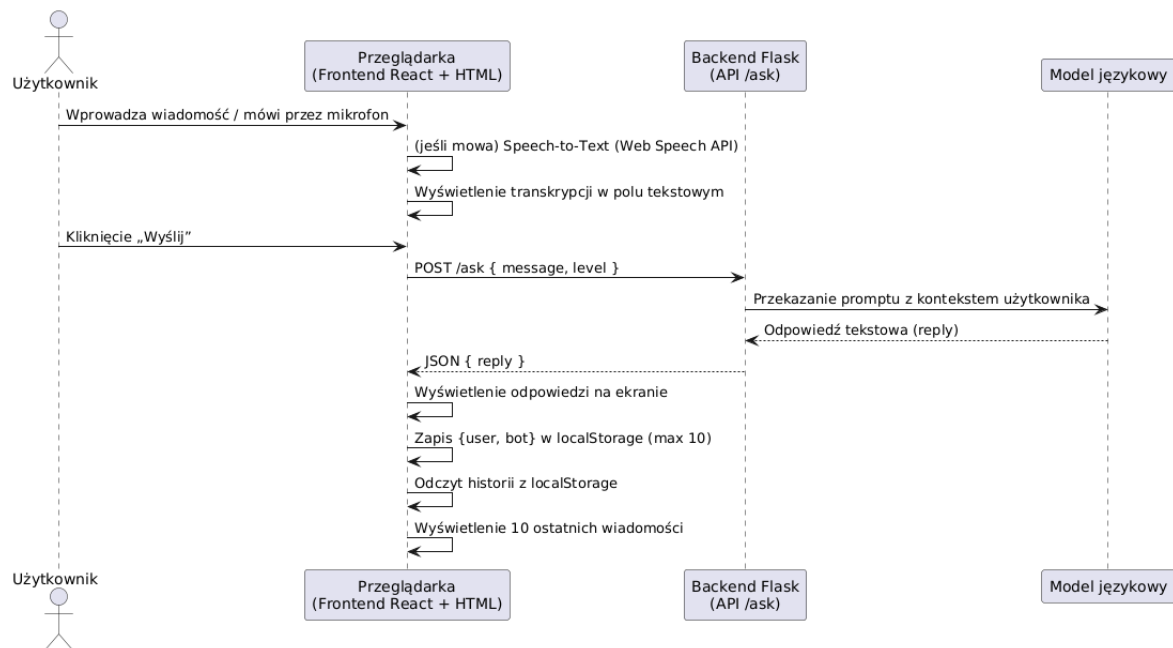


Rys. 1 Diagram użycia

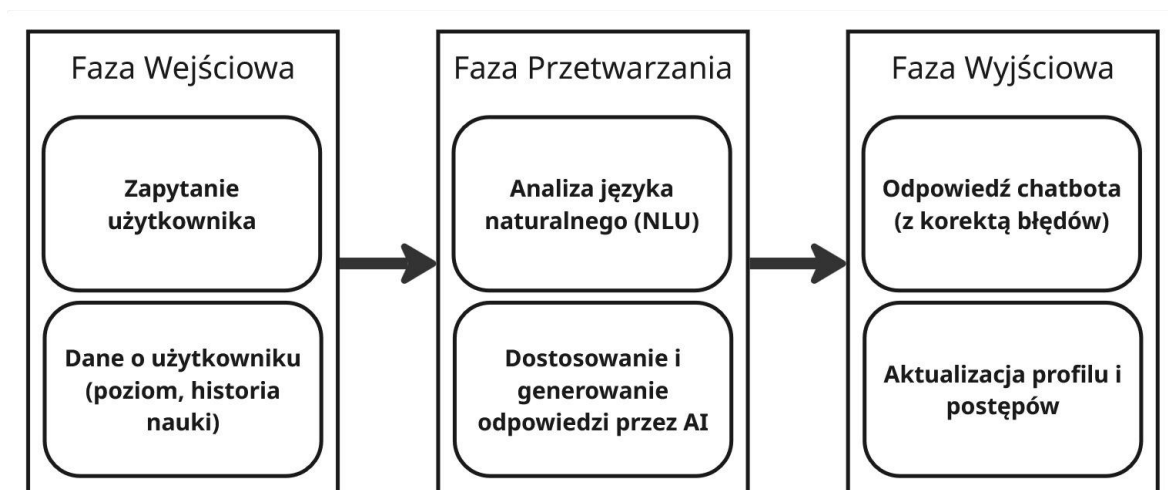


Rys. 2 Diagram przepływu danych

## 5. Diagram sekwencyjny UML.



## 6. Projekt architektury opracowanego systemu.



Każdy z trzech dużych prostokątów reprezentuje poszczególne komponenty systemu. Zadaniem systemu jest najpierw przyjąć dane wejściowe od użytkownika, czyli jego wiadomość oraz poziom zaawansowania, aby następnie przetworzyć je przez algorytmy sztucznej inteligencji. Na końcu system generuje spersonalizowaną odpowiedź i aktualizuje profil postępów użytkownika.

Pierwszym kluczowym i krytycznym elementem systemu jest prawidłowa interpretacja danych wejściowych w celu zapewnienia adekwatnej i efektywnej interakcji. Drugim ważnym komponentem systemu są algorytmy odpowiedzialne za analizę języka i generowanie odpowiedzi. Ich dobór oraz metoda działania muszą być precyzyjne i elastyczne ze względu na różnorodność zapytań i poziomów zaawansowania

użytkowników. Wszystko to stanowi podstawę do wygenerowania odpowiedzi, która jest końcową, kluczową częścią interakcji, oraz do aktualizacji profilu, która zapewnia ciągłość i personalizację procesu nauki.

## 7. Sugerowany język implementacji: Python

### Uzasadnienie:

W projekcie zdecydowano się na wykorzystanie języka Python jako głównego języka programowania po stronie serwera. Głównym powodem tego wyboru jest fakt, że członkowie zespołu najlepiej znają właśnie ten język, co umożliwia efektywną współpracę, szybkie rozwiązywanie problemów oraz sprawną realizację zadań programistycznych.

Python, w połączeniu z frameworkiem Flask, stanowi lekkie i elastyczne rozwiązanie do tworzenia warstwy backendowej aplikacji webowej; przy użyciu REST API pozwala komunikować się z frontendem zbudowanym we frameworku React, a także z zewnętrznym API modelu językowego (LLM) wykorzystywanym przez naszego chatbota.

## 8. Przykładowa Wizualizacja:

