

# Dokumentacja inżynierii wymagań

Członkowie zespołu:

Zespół	Skład zespołu
<b>MDM</b>	<b>Mateusz Sznurawa</b>
	<b>Dawid Gruszecki</b>
	<b>Michał Chmiel</b>

## 1. Macierz kompetencji zespołu

Kompetencje	Dawid Gruszecki	Michał Chmiel	Mateusz Sznurawa
Znajomość języków skryptowych	"+"	"+"	"+"
Umiejętność tworzenia UI	"+"	"+"	"+"
Programowanie backendowe	"+"	"+"	"+"
Umiejętność korzystania z API	"-"	"-"	"+"
Doświadczenie z bibliotekami przetwarzania audio	"-"	"-"	"-"
Znajomość frameworków uczenia maszynowego	"-"	"+"	"-"
Testowanie oprogramowania	"+"	"+"	"+"

## 2. Dodatkowe pytania precyzujące

Pytanie	Odpowiedź	Uwagi
Czy po spotkaniu składany jest raport?	Tak, do uczestników spotkania wysyłana jest wiadomość email z raportem.	
Kto może nagrywać ekran?	Każdy użytkownik obecny na spotkaniu.	
Dla jakich platform powinno być wsparcie?	Zoom, MS-Teams, Google Meet.	
Czy identyfikacja mówców jest potrzebna ?	Traktujemy jako <i>nice to have</i> .	

Czy na podstawie spotkania określane są statystyki?	Statystykę mowy oraz udziału mówcy w spotkaniu traktujemy jako <i>nice to have</i> .	Implementacja zależna od postępu prac.
Jak prezentowane będą elementy prezentowane na spotkaniu (wykresy, grafy, wizualizacje)?	Ich zrzuty ekranu prezentowane będą w podsumowaniu/notatce.	
Czy aplikacja ma być zintegrowana z kalendarzem?	Tak, polegać ma na ustaleniu czasu automatycznego rozpoczęcia nagrywania.	
Jak przedstawione zostaną tekstowe elementy prezentowane w czasie spotkania?	Tekst uzyskany z pomocą optycznego rozpoznawania znaków będzie dodawany do notatki.	
Czy wymagane jest podsumowanie notatek?	Tak, notatki wymagają podsumowania.	Podsumowanie z notatek generowane automatycznie.
W jakim formacie dostępne będą notatki?	W formatach PDF, HTML, .mo	
W jaki sposób będzie można zapoznać się z wypowiedziami uczestników?	Za pomocą transkrypcji mowy.	Kwestia tego, czy odbędzie się to na żywo, czy z odtworzenia zostanie ustalona.

### 3. Ustalony format danych wejściowych

Element	Opis	Format	Źródło
Wejście - Audio	Dźwięk z mikrofonów uczestników spotkania	WAV, MP3	Przechwytywane przez rozszerzenie Chrome z aktywnej karty przeglądarki
Wejście - Wideo	Obraz z udostępnianego ekranu	MKV, MP4	Przechwytywane przez rozszerzenie Chrome z aktywnej karty przeglądarki
Wejście - Metadane	Informacje o spotkaniu i uczestnikach	JSON	Pobierane z platformy telekonferencyjnej lub wprowadzane przez użytkownika
Wyjście - Transkrypcja	Tekstowa transkrypcja mowy	TXT, SRT, JSON	Generowana przez Web Speech API lub zewnętrzną usługę transkrypcji
Wyjście - OCR	Tekst rozpoznany z obrazu	TXT	Generowany przez bibliotekę OCR lub zewnętrzną usługę OCR
Wyjście - Notatki	Połączona transkrypcja i tekst z OCR	TXT, PDF, HTML	Generowane przez rozszerzenie Chrome, opcjonalnie z wykorzystaniem serwera backendowego

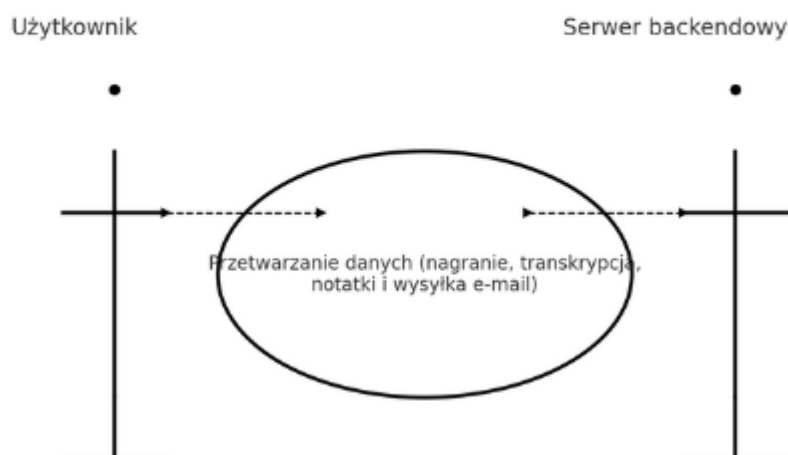
#### 4.1 Przedstawienie modelowanego systemu za pomocą tabeli

Użytkownicy	Uczestnicy spotkania online.
Opis	Wtyczka rejestruje dźwięk i/lub wideo ze spotkania, przetwarza nagranie na tekstową transkrypcję i notatki, a następnie wysyła je na podany e-mail.
Dane	Nagranie audio/wideo, metadane o spotkaniu (uczestnicy, czas trwania), ustawienia użytkownika (np. adres e-mail, format notatek).
Wyzwalacz	Rozpoczęcie nagrania po kliknięciu przycisku w wtyczce Chrome.
Odpowiedź	Transkrypcja tekstowa spotkania, wygenerowane notatki (w wybranym formacie: TXT, PDF), wysyłka na adres e-mail użytkownika.
Uwagi	<ul style="list-style-type: none"><li>- Wtyczka powinna działać w tle podczas spotkania.</li><li>- Nagranie nie może być większe niż maksymalny limit rozmiaru pliku obsługiwanej przez API przetwarzania.</li></ul>

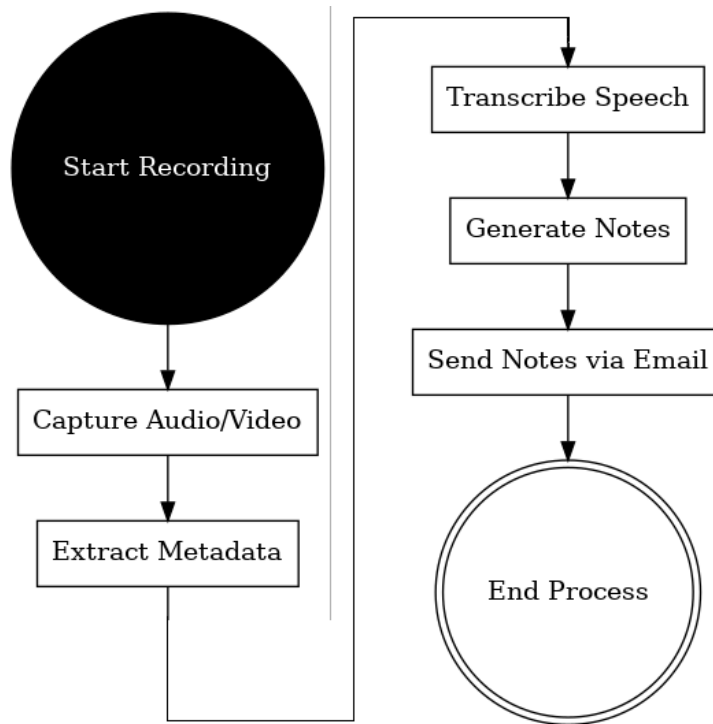
#### 4.2 Przedstawienie modelowanego systemu za pomocą diagramów UML

Rys. 1 Diagram przypadków użycia

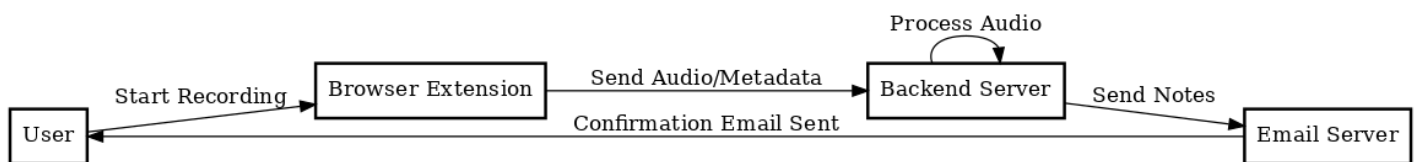
Diagram UML - Wtyczka Chrome do transkrypcji i wysyłki notatek



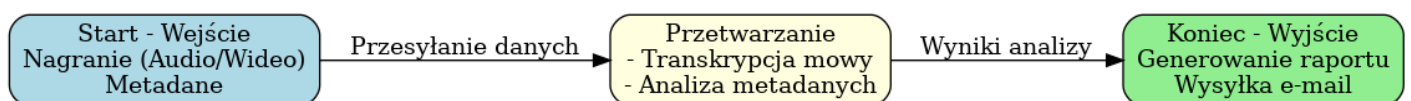
Rys. 2 Diagram przepływu danych



## 5. Diagram sekwencyjny UML



## 6. Projekt architektury opracowywanego systemu



Każdy z trzech dużych prostokątów reprezentuje poszczególne komponenty systemu. Zadaniem systemu jest najpierw przyjąć dane w postaci nagrania audio/wideo oraz metadanych, aby następnie przetworzyć je w celu uzyskania transkrypcji oraz analizy. Na końcu system generuje raport zawierający zrozumiałe notatki oraz przesyła je na wskazany adres e-mail.

Pierwszym kluczowym i krytycznym elementem systemu jest zapewnienie, że dane wejściowe (nagrania oraz metadane) są poprawne i odpowiednio sformatowane, aby system mógł je skutecznie przetworzyć. Bez tej walidacji przetwarzanie mogłoby zakończyć się niepowodzeniem lub błędnym raportem.

Drugim istotnym komponentem systemu jest proces przetwarzania danych, który obejmuje transkrypcję mowy, analizę metadanych oraz integrację tych informacji w celu wygenerowania czytelnych notatek. Proces ten musi być szybki i efektywny, aby użytkownicy mogli otrzymać raport w możliwie krótkim czasie.

Na zakończenie, system generuje raport w wybranym formacie (np. PDF, TXT) oraz wysyła go na podany adres e-mail. Ostatni krok jest niekrytyczny, lecz kluczowy dla zapewnienia wartości użytkownikowi końcowemu. Dzięki temu system może być używany w praktycznych zastosowaniach, takich jak dokumentacja spotkań czy analiza rozmów.

## 7. Tech Stack

- **JavaScript** - podstawowy język tworzenia rozszerzeń Chrome, zostanie wykorzystany do budowy interfejsu użytkownika, interakcji z API przeglądarki, przetwarzania danych i komunikacji z serwerem backendowym.
- **Python** - jeśli zdecydujemy się na implementację serwera backendowego, będzie on dobrym wyborem.
- **HTML, CSS** - do tworzenia interfejsu użytkownika.
- **Biblioteki przetwarzania audio** - do analizy i manipulacji dźwiękiem.
- **Biblioteki OCR, np. Tesseract** - do rozpoznawania tekstu z udostępnianych obrazów.