

Dokumentacja inżynierii wymagań

Członkowie zespołu:

Zespół	Skład zespołu
MDM	Mateusz Sznurawa
	Dawid Gruszecki
	Michał Chmiel

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 zewnętrznymi usługami?	Tak, powinna być zintegrowana z Google Calendar.	
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	WEBM	Przechwytywanie przez rozszerzenie Chrome z aktywnej karty przeglądarki.
Wejście - Video	Obraz z udostępnianego ekranu.	WEBM	Przechwytywanie przez rozszerzenie Chrome z aktywnej karty przeglądarki.
Wejście - Spotkanie w	Przedział czasowy spotkania, opis,	JSON	Przesyłane przez Google

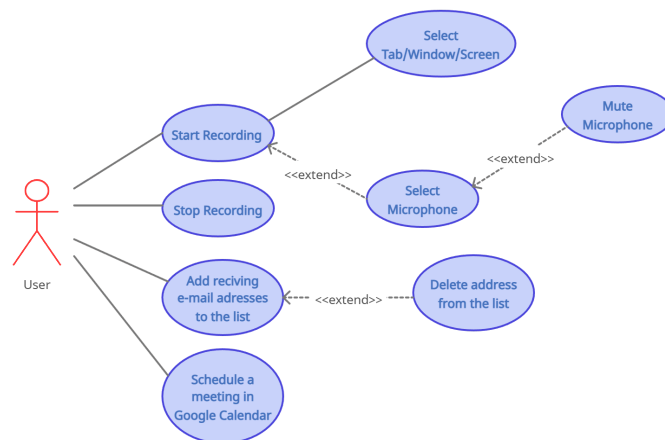
Kalendarzu	szczegóły		Calendar API
Wyjście - Transkrypcja	Tekstowa transkrypcja mowy	TXT	Transkrypcja generowana przez Google Speech-to-Text Api
Wyjście - OCR	Tekst rozpoznany z obrazu	TXT	Generowany przez bibliotekę OCR - Tesseract
Wyjście - Notatki	Połączona transkrypcja i tekst z OCR	HTML, TXT	Generowane automatycznie przez Gemini AI

4.1 Przedstawienie modelowanego systemu za pomocą tabeli

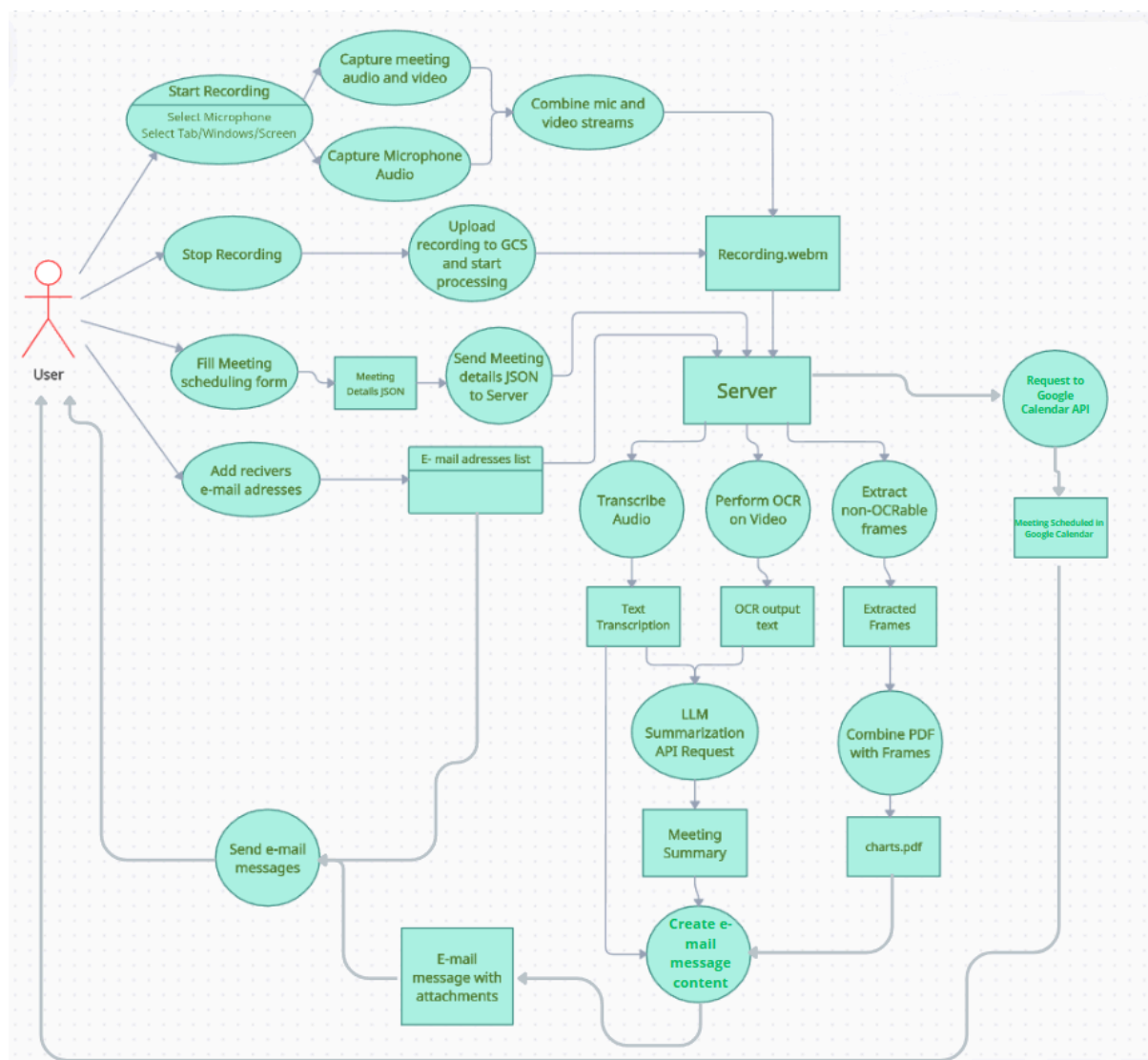
Użytkownicy	Uczestnicy spotkania online.
Opis	Wtyczka rejestruje dźwięk i wideo ze spotkania lub odtwarzanego nagrania, przetwarza nagranie na tekstową transkrypcję i automatycznie generuje notatki dodając zrzuty ekranu i treść wynikającą z optycznego rozpoznawania znaków, a następnie wysyła je na podane adresy e-mail.
Dane	Nagranie audio/wideo, informacje o spotkaniu zawarte w Google Calendar, adresy e-mail odbiorców.
Wyzwalacz	Rozpoczęcie nagrania po kliknięciu przycisku w wtyczce Chrome.
Odpowiedź	Transkrypcja tekstowa spotkania i automatycznie wygenerowane notatki przesłane w wiadomości email i pliku .PDF, wysyłka na adresy e-mail użytkowników.
Uwagi	<ul style="list-style-type: none"> - Wtyczka powinna działać w tle podczas spotkania. - Nagranie nie może być większe niż maksymalny limit rozmiaru pliku obsługiwanego przez API przetwarzania. - Liczba wyciętych klatek przesłanych w podsumowaniu może być ograniczona limitem wielkości przesyłanych załączników wiadomości e-mail.

4.2 Przedstawienie modelowanego systemu za pomocą diagramów UML

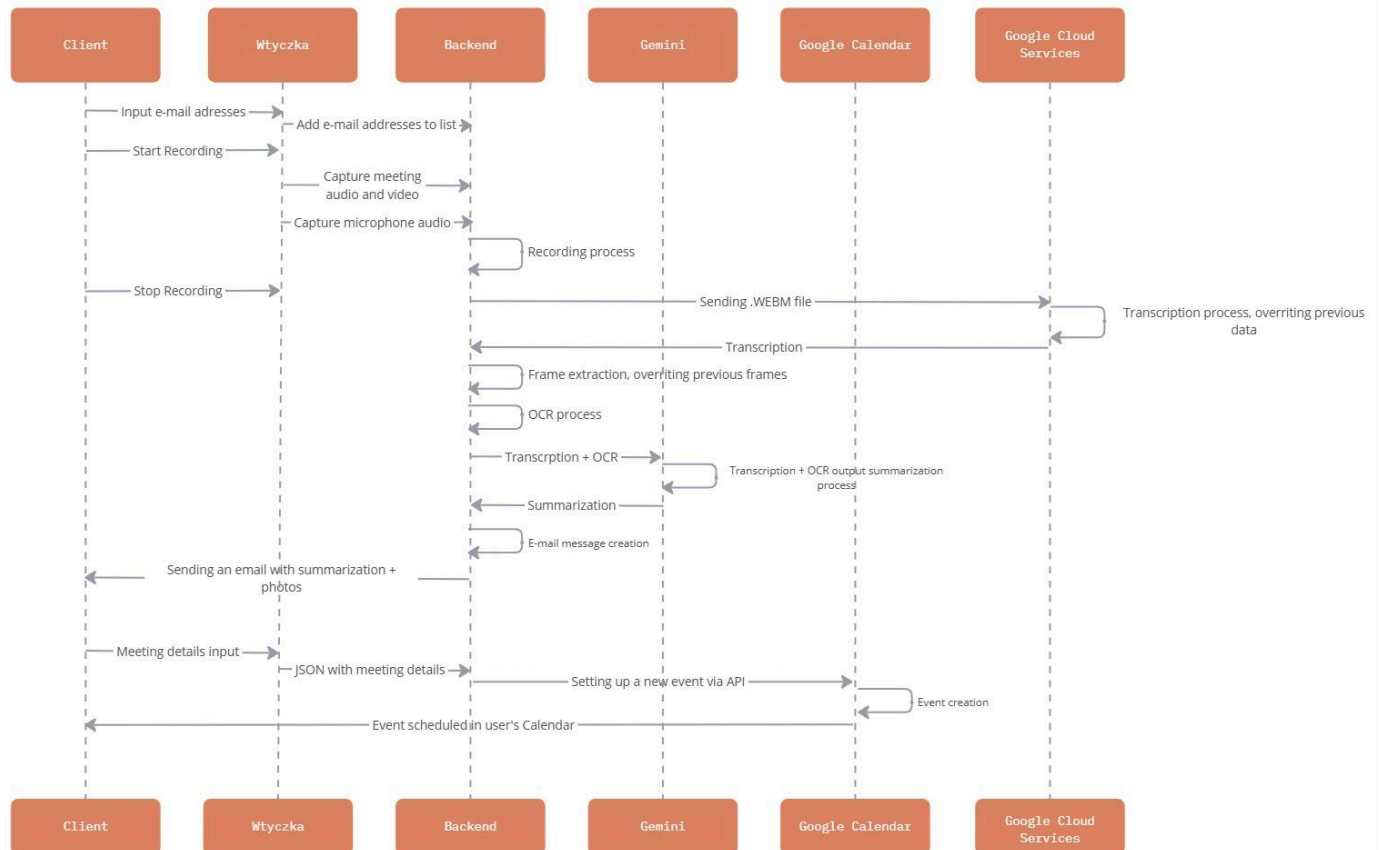
Rys. 1 Diagram UML przypadków użycia



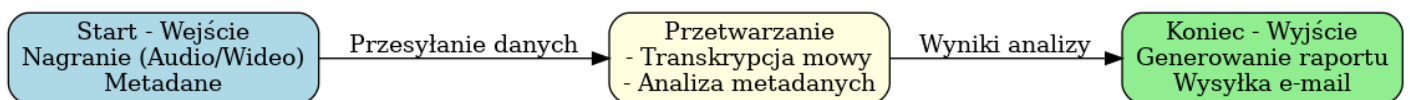
Rys. 2 Diagram przepływu danych



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 podane adresy 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, wykorzystany do budowy interfejsu użytkownika, interakcji z API przeglądarki, przetwarzania danych i komunikacji z serwerem backendowym.
- **Node.js** - implementacja serwera backendowego.
- **HTML, CSS** - tworzenie interfejsu użytkownika.
- **FFMpeg** - przetwarzanie wideo.
- **Google Cloud Storage** - transkrypcja spotkania.
- **Tesseract** - rozpoznawanie tekstu z udostępnianych obrazów.
- **Gemini 1.5** - tworzenie podsumowań spotkań.
- **ImageHash** - przetwarzanie obrazka na bity celem porównywania zmian między klatkami.
- **PDFKit** - generowanie pliku pdf z slajdami z spotkania.
- **Archiver** - pakowanie i kompresowanie wygenerowanych plików do zip.
- **NodeMailer** - wysyłanie maili z transkrypcją i paczką zip z plikami.