# Dokumentacja Inżynierii Wymagań

1. **Macierz kompetencji zespołu**

|  |  |  |  |
| --- | --- | --- | --- |
| Kompetencja | Jan Krzyszkowski | Karol Słowiak | Piotr Kondziołka |
| Testowanie oprogramowania | 2 | 2 | 2 |
| Programowanie w Rust | 0 | 0 | 0 |
| Relacyjne bazy danych | 2 | 2 | 2 |
| Programowanie w Python | 2 | 2 | 2 |
| Nierelacyjne bazy danych | 1 | 1 | 1 |
| Programowanie w ADA | 2 | 0 | 0 |
| Znajomość UMLa | 1 | 1 | 1 |
| Tworzenie dokumentacji | 1 | 2 | 1 |

#### *Legenda:*

* 0 - Nie posiada
* 1 - Podstawy
* 2 - Posiada

1. **Zestaw pytań dla uszczegółowienia projektu**

|  |  |  |
| --- | --- | --- |
| **Pytanie** | **Odpowiedź** | **Uwagi** |
| Jakie aplikacje do wideokonferencji obsługujemy? | Wszystkie, aplikacja działa niezależnie, nie korzysta z konkretnych API aplikacji do wideokonferencji | Aplikację można również używać do nagrywania i streszczania filmów |
| Na jakiej platformie aplikacja ma działać? | Jest to aplikacja desktopowa, działa zarówno na systemach Linux, jak i Windows |  |
| Jakie możliwości interakcji ma użytkownik? | Nagrywanie ekranu z dźwiękiem, tworzenie zrzutów ekranu | Przed uruchomieniem nagrywania należy podać email na który będą wysłane wyniki |
| Jaki problem rozwiązuje aplikacja? | Nagrywanie zdalnych spotkań umożliwia dotarcie do informacji przekazywanych na nich, zarówno w przypadku potrzeby przypomnienia jak i nieobecności użytkownika | Wykorzystywane urządzenie musi być uruchomione, wraz ze spotkaniem, którego użytkownik chce nagrać |
| Czy potrzeba rejestracji oraz logowania? | Nie | Potrzebne jest jedynie podanie maila, na który wysyłana jest wiadomość z podsumowaniem |

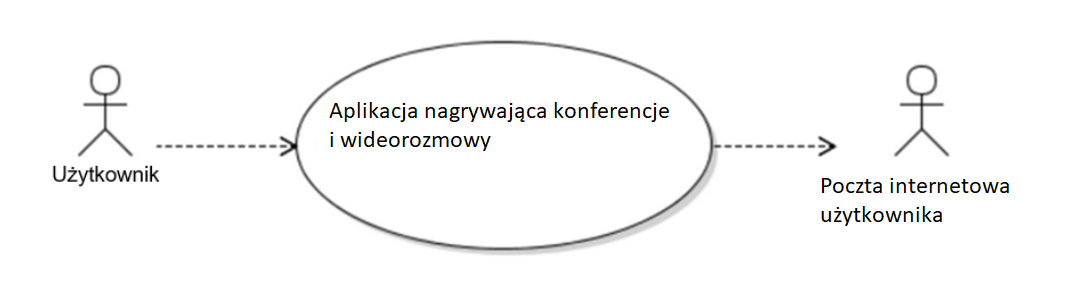
1. **Ustalony format danych**

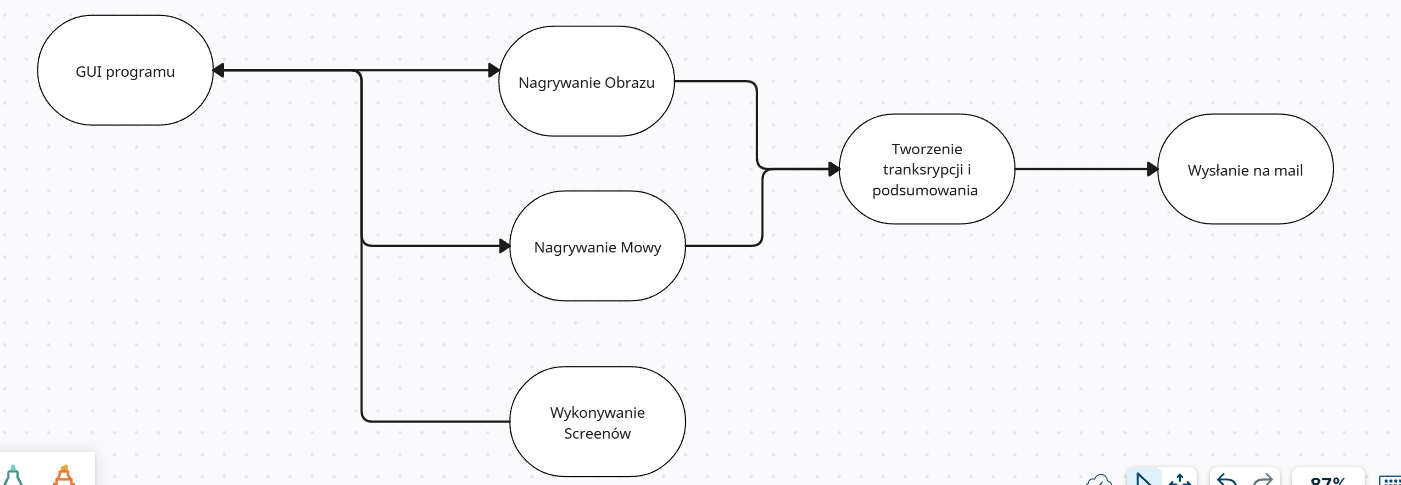
|  |  |  |  |
| --- | --- | --- | --- |
| **Źródło** | **Typ Wejściowy** | **Typ Wyjściowy** | **Opis** |
| video.py | Plik \*.jpg | Plik \*.avi | Tworzenie nagrania z klatek |
| sound.py | Dźwięk z karty | Plik \*.wav | Przechwytywanie dźwięku |
| speechToText.py | Plik \*.wav  Plik \*.avi | Plik \*.txt | Generowanie tekstu z mowy za pomocą usługi Google’a,  Wysyłanie wiadomości email |
| groqPart.py | Plik \*.txt | Plik \*.txt | Utworzenie podsumowania z wykorzystaniem AI |

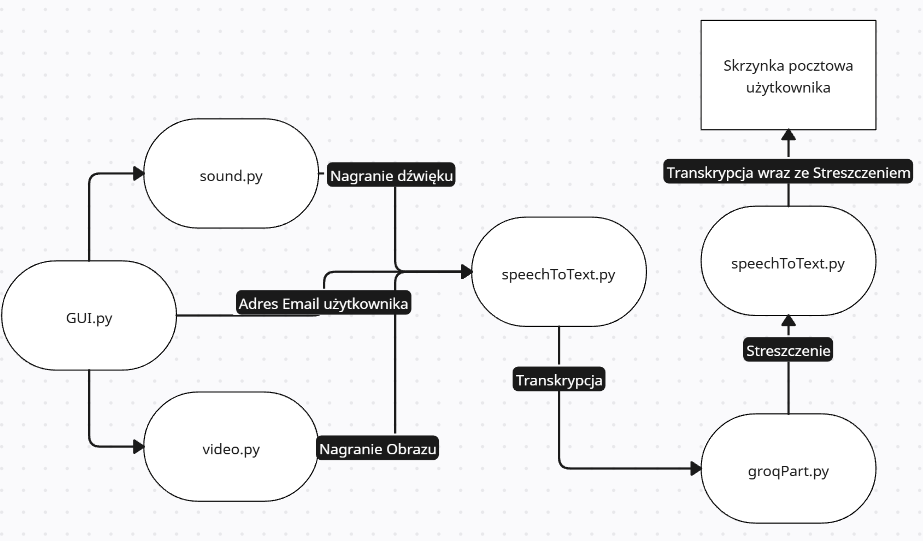
1. **Przedstawienie modelowanego systemu za pomocą tabeli**

|  |  |
| --- | --- |
| Aktorzy | Zespół deweloperski |
| Opis | Tworzenie aplikacji desktopowej do nagrywania wideokonferencji, zapisywania zrzutów ekranu, transkrypcji oraz podsumowania z wykorzystaniem AI, a następnie wysyłania ich na podany przez użytkownika email |
| Dane | Obraz i dźwięk z komputera, email |
| Wyzwalacz | Ręczne inicjowanie nagrywania oraz zrzutów ekranu poprzez naciśnięcie dedykowanego przycisku w aplikacji |
| Odpowiedź | Nagranie, Zrzuty ekranu, transkrypcja oraz podsumowanie wykonywana po zakończeniu nagrywania |
| Uwagi | Aplikacja obsługuje język angielski |

1. **Przedstawienie systemu za pomocą diagramów UML**  
   *Rys. 1 Diagram przypadków użycia*

****

*Rys. 2 Diagram przepływu danych*

1. **Diagram sekwencyjny UML**
2. **Projekt Architektury opracowanego systemu**

|  |  |  |
| --- | --- | --- |
| **Start – wejście** | **Przetwarzanie** | **Koniec – wyjście** |
| Uruchomienie aplikacji desktopowej do wideokonferencji,  Nagrywanie ekranu i dźwięku z urządzenia | Tworzenie transkrypcji na podstawie pliku \*.wav,  Tworzenie podsumowania z wykorzystaniem API modelu Groq, | Email z plikami \*.txt zawierającymi transkrypcję oraz streszczenie. Nagranie wraz z zrzutami ekranu zapisane lokalnie |

1. **Sugerowany język implementacji: Python**

Uzasadnienie:

Zdecydowaliśmy się na wykorzystanie języka Python ze względu na nasze kompetencje oraz jego wszechstronność w realizacji projektów związanych z przetwarzaniem mowy. Python jest językiem, który znamy ogólnikowo najlepiej, co pozwoli nam na sprawną i efektywną implementację zaawansowanych funkcji.