

Inżynieria Wymagań - Projekt IO

1. Macierz kompetencji zespołu

Kompetencje	Zuzanna Rotarska	Szczepan Dworak	Grzegorz Bąk
Python (desktop apps)			
Biblioteki do przetwarzania dźwięku			
Przetwarzanie obrazu			
Przetwarzanie mowy			
Tworzenie GUI			
Narzędzia developerskie i debugging			
Analiza ruchu sieciowego (HTTP)			
WebSockets / komunikacja w czasie rzeczywistym			
Testowanie oprogramowania			

2. Pytania i odpowiedzi z klientem

Pytanie	Odpowiedź	Uwagi
W jaki sposób powinny być zapisywane dane z obrazu udostępnionego przez uczestników spotkania?	Treść udostępnionego ekranu powinna być zapisana w postaci tekstowej za pomocą OCR lub w postaci zrzutów ekranu.	Klient pozostawił zespołowi deweloperów dowolność co do wybranego sposobu przedstawienia danych (OCR lub zrzutu ekranu). Zespół wybiera opcję ze zrzutami ekranu.
W jakim formacie powinien zostać zapisany plik danych zawierający notatki ze spotkania.	Plik powinien zostać zapisany w formacie PDF, TXT.	

Jakie elementy interfejsu aplikacja powinna zawierać?	Przycisk start, stop zaczynające proces rejestrowania danych do notatek, jak i kończący ten proces. Interfejs do przeglądania danych notatek. Menu ustawień z maksymalnym miejscem jakie może zajmować notatka, jakością wideo i językiem.	Obsługiwane języki na tą chwilę to jedynie język polski.
Czy aplikacja powinna mieć jakieś funkcjonalności związane z datowaniem spotkań i notatek na ich podstawie sporządzonych?	Aplikacja powinna mieć funkcję, pozwalającą na przeglądanie notatek ze względu na datę, jak i możliwość rozpoczęcia rejestrowania danych dla spotkania rozpoczętego.	
Czy aplikacja powinna rozróżniać źródło danych/ poszczególnych uczestników spotkania?	Notatki powinny być otagowane w oparciu o nazwę rozmówcy.	Funkcjonalność jest opcjonalna i nie wymagana przez klienta w pierwszej wersji produkcyjnej.
Jakie usługi telekonferencji aplikacja powinna wspierać?	Aplikacja powinna wspierać Zoom, MS Teams, Google Meet	
Czy aplikacja powinna być desktopowa czy przeglądarkowa?	Desktopowa	

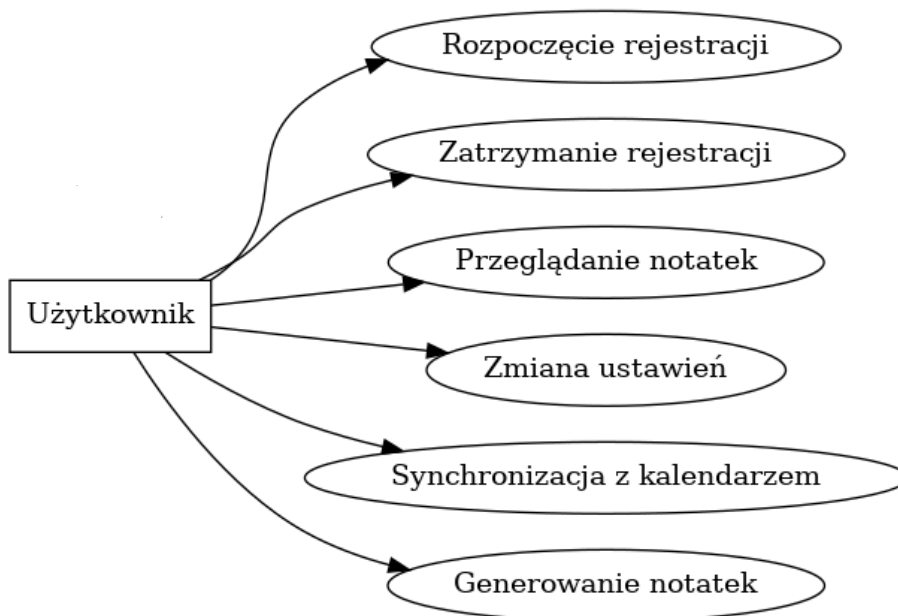
3. Ustalony format danych

Dane są pobierane z widoku ekranu użytkownika oraz danych audio pochodzących z aplikacji telekonferencyjnej. Aplikacja nagrywa audio oraz okresowo wykonuje zrzuty ekranu w celu pozyskania danych do notatek.

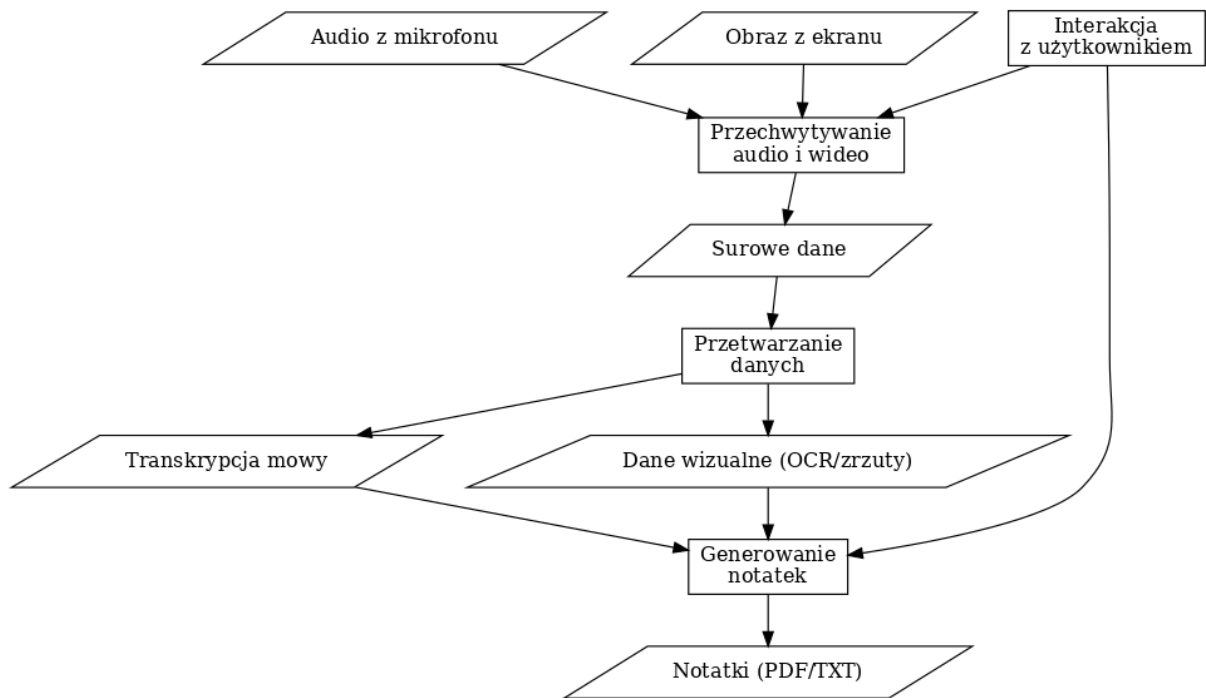
4.1. Modelowanie systemu za pomocą tabeli

Aktorzy	Użytkownik
Opis	Celem programu jest stworzenie notatki ze spotkania zawierającej transkrybowane audio oraz przedstawione na udostępnionym ekranie w spotkaniu dane/informacje przekazane uczestnikom spotkania.
Dane	Audio z aplikacji telekonferencyjnej, Dane wizualne z udostępnionego ekranu
Wyzwalacz	Przycisk "Start"
Odpowiedź	Plik PDF, TXT
Uwagi	Dane powinny zajmować jak najmniej miejsca tak aby użytkownik mógł zarezerwować tylko potrzebne miejsce.

4.2. Przedstawienie modelowego systemu za pomocą diagramów UML

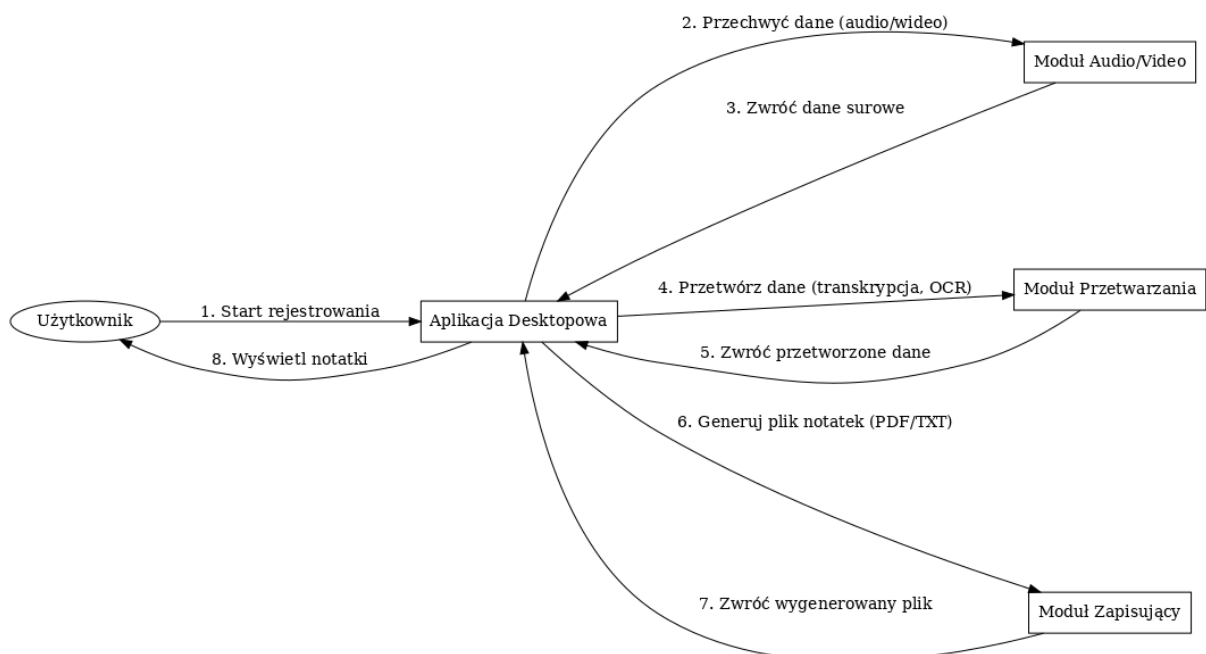


Rys. 1 Diagram przypadków użycia

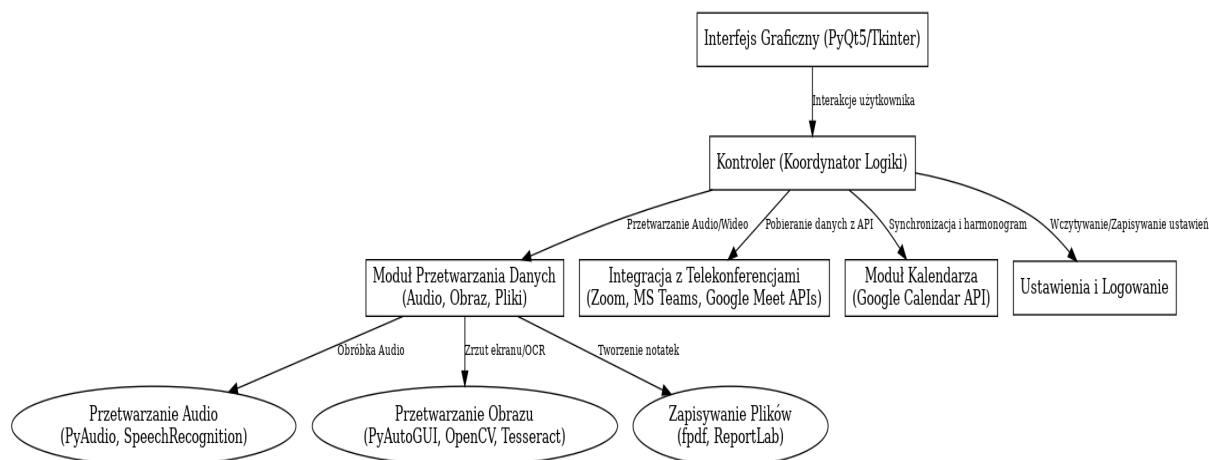


Rys. 2 Diagram przepływu danych

5. Diagram Sekwencyjny UML



6. Projekt architektury opracowywanego systemu



Prostokąty i elipsy reprezentują komponenty systemu. System w pierwszej kolejności wyświetla użytkownikowi, główne okno interfejsu graficznego aplikacji w którym użytkownik dokonuje wyboru ustawień dotyczących rejestracji notatek, ustala/pobiera z kalendarza dane o planowanych spotkaniach oraz wybiera rodzaj narzędzia do telekonferencji. W następnym kroku po rozpoczęciu spotkania i jego rejestracji dzięki zastosowaniu szeregu funkcji program gromadzi dane ze spotkania, a następnie je przetwarza. W ostatnim kroku system zgromadzone i przetworzone dane zapisuje w formacie TXT lub PDF i przekazuje użytkownikowi.

7. Sugerowany język implementacji: Python

Python został wybrany jako główny język programowania ze względu na swoją prostotę, wszechstronność i bogaty ekosystem bibliotek. Oferuje gotowe narzędzia do obsługi dźwięku (PyAudio), obrazu i OCR (OpenCV, Tesseract), a także przetwarzania mowy (SpeechRecognition, Google Speech API), co znacznie przyspiesza rozwój projektu. Jego frameworki, takie jak PyQt czy Tkinter, umożliwiają łatwe tworzenie aplikacji desktopowych, a wsparcie społeczności oraz dostępność dokumentacji sprawiają, że rozwiązywanie problemów jest proste. Python działa na różnych systemach operacyjnych, co czyni go idealnym wyborem do budowy uniwersalnej i zaawansowanej aplikacji.