Dokumentacja inżynierii wymagań

**Cel dokumentu**

Celem niniejszego dokumentu jest szczegółowe określenie wymagań funkcjonalnych i niefunkcjonalnych dla systemu transkrypcji mowy na tekst oraz dodatkowych funkcji związanych z zapisem i przetwarzaniem spotkań online. Dokument ten ma na celu zapewnienie spójnego i precyzyjnego opisu oczekiwań, które będą stanowiły podstawę do projektowania, implementacji oraz testowania rozwiązania.

**Słownik pojęć**

* Transkrypcja mowy – proces automatycznego przekształcania mowy ( audio ) na tekst
* Zoom, Google Meet, MS Teams – narzędzia służące do prowadzenia telekonferencji, które powinny być wspierane przez oprogramowanie
* UI ( User Interface ) – interfejs użytkownika aplikacji
* DOCS, TXT – formaty plików, w których będą przechowywane dane związane z nagranymi spotkaniami
* Notes - Zbiór plików, służących jako notatka ze spotkania
* App\_Calendar\_Actions - Utworzenie eventu w Google Calendar
* App\_User\_Actions - Rozpoczęcie, zakończenie procesu nagrywania. Zmiana ustawień aplikacji (jakość nagrania, język transkrypcji, użyte narzędzie telekonferencyjne, maksymalna ilość miejsca na dysku, którą może wykorzystać aplikacja na przechowywanie Notes). Umożliwienie przeglądania notatek (lokalnie - pliki, online - na stronie internetowej). Szukanie słów kluczowych w notatkach (lokalnie). Logowanie do Google Calendar (w celu integracji)
* User\_Calendar\_Actions - Umożliwienie przeglądania notatek przez URL (online - na stronie internetowej)

**Członkowie zespołu**

Tabela Macierz kompetencji zespołu

|  |  |  |  |
| --- | --- | --- | --- |
| **Kompetencje** | **Jakub Kogut** | **Jakub Kowalski** | **Filip Kaczor** |
| Python | Posiada | Posiada | Posiada |
| Eclipse JS | Posiada ( podstawy ) | Posiada ( podstawy ) | Posiada |
| Design UI | Posiada | Posiada | Posiada |
| UML | Posiada(podstawy) | Posiada(podstawy) | Posiada(podstawy) |
| Testowanie oprogramowania | Nie posiada | Nie posiada | Posiada |

**Zestaw pytań sformułowanych w trakcie spotkania**

|  |  |  |
| --- | --- | --- |
| **Pytanie** | **Odpowiedź** | **Uwagi** |
| **Jakie narzędzia do telekonferencji ma obsługiwać program?** | Działa na Zoom, MS Teams, Google Meet |  |
| **Czy użytkownik może wykonywać inne czynności na urządzeniu w trakcie nagrywania spotkania?** | Nie będzie zakłócał nagrywania (audio i video): nie będzie zasłaniał okna nagrywanego spotkania i wprowadzał dodatkowego dźwięku (np. muzyka w tle) |  |
| **W jakich formatach notatki ze spotkania mają być zapisane?** | Notatki .docx i .txt Nagranie video .mp4 Zrzuty ekranu .png | Powinno być stworzone podsumowanie spotkania |
| **Czy spotkanie ma mieć transkrypcję?** | Tak | Z podziałem na role - opcjonalne |
| **Czy spotkanie ma mieć timestampy?** | Tak | Chronologicznie |
| **Jaki język transkrypcji powinien być obsługiwany?** | Polski, Angielski | Możliwe błędy w transkrypcji |
| **Czy program powinien mieć dodatkowe integracje?** | Tak, z kalendarzem Google |  |
| **Jak umożliwić podgląd notatek?** | Za pomocą plików .docx i .txt (lokalnie na urządzeniu użytkownika).  Przez stronę internetową |  |

Czy wszystkie wymagania klienta są możliwe do spełnienia? Prawdopodobnie.

**Ustalony format danych wejściowych**

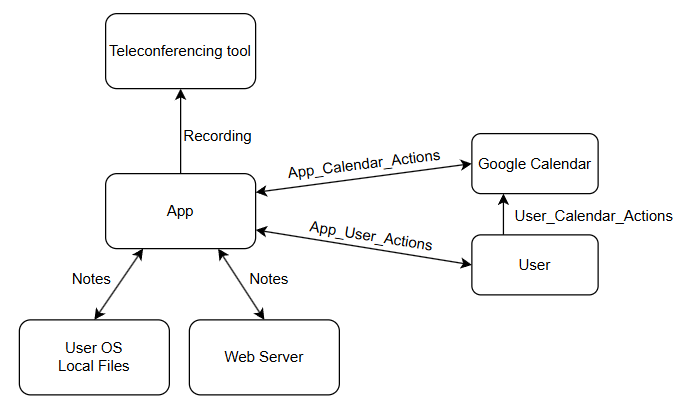
Nagranie w formacie .mp4.

**Modelowanie systemu**

Tabela Modelowanie systemu za pomocą tabeli

|  |  |
| --- | --- |
| **Aktorzy** | Użytkownik |
| **Opis** | Celem działania modelowanego systemu jest nagrywanie spotkań i tworzenie na ich podstawie transkrypcji. Dodatkowo powinien mieć integrację z kalendarzem, tworzyć w nim wydarzenia i dodawać opisy z danymi ze spotkania. |
| **Dane** | Nagranie, jakość nagrania, narzędzie telekonferencyjne |
| **Wyzwalacz** | Przycisk do rozpoczęcia nagrywania |
| **Odpowiedź** | Transkrypcja spotkania, zrzuty ekranu z prezentacji dostępne lokalnie i przez stronę internetową w sieci enterprise |
| **Uwagi** | Brak |

Rysunek Diagram zależności pomiędzy komponentami systemu



Rysunek Diagram przetwarzania danych komponentu - Data Processing

Obraz zawierający tekst, diagram, Plan, linia

Opis wygenerowany automatycznie

**Format danych wyjściowych**

Notatka po spotkaniu będzie zawierać:

1. Nazwę spotkania
2. Transkrypcję spotkania (z podziałem na role - opcjonalne)
3. Zrzuty ekranu zawierające potencjalne slajdy prezentacji pojawiającej się na ekranie
4. Timestampy do wypowiedzi i zrzutów ekranu

Format plików wyjściowych:

1. Plik .docx zawierający powyższą notatkę
2. Plik .txt zawierający tylko transkrypcję
3. Nagranie w pliku .mp4 dostępne lokalnie
4. Strona internetowa zawierająca powyższą notatkę

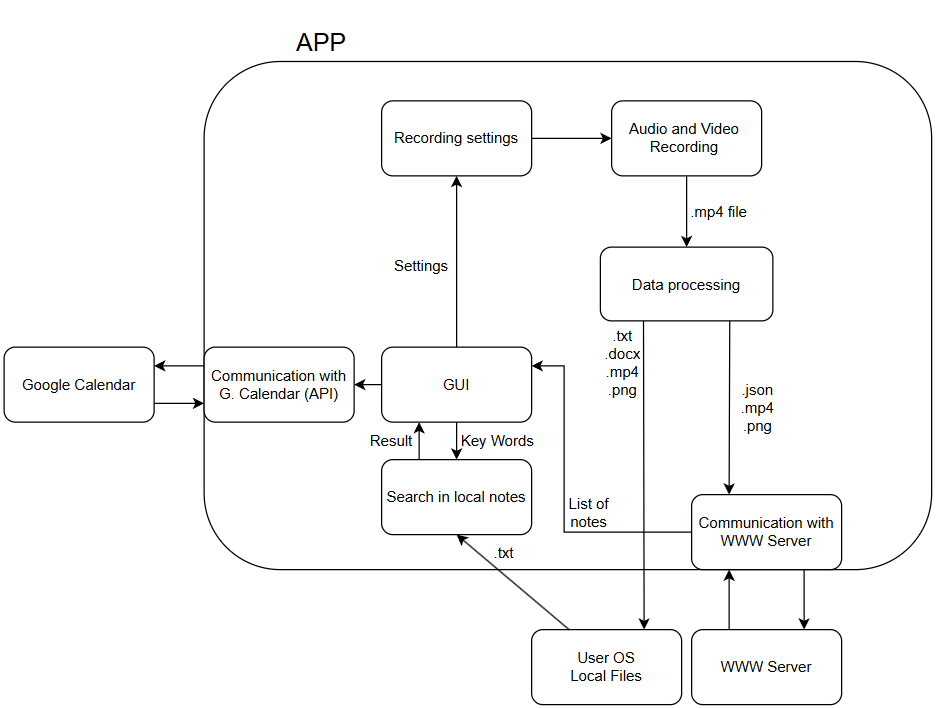
**Implementacja**

Narzędzia

1. Aplikacja desktop w języku Python  
    Dodatkowo użyte biblioteki:
   1. nagrywanie obrazu
   2. nagrywanie dźwięku
   3. speech to text
   4. komunikacja z kalendarzem Google
2. Serwer web w języku PHP  
    Dodatkowo użyte biblioteki:
   1. obsługa plików JSON
3. System operacyjny: Microsoft Windows 10 i 11

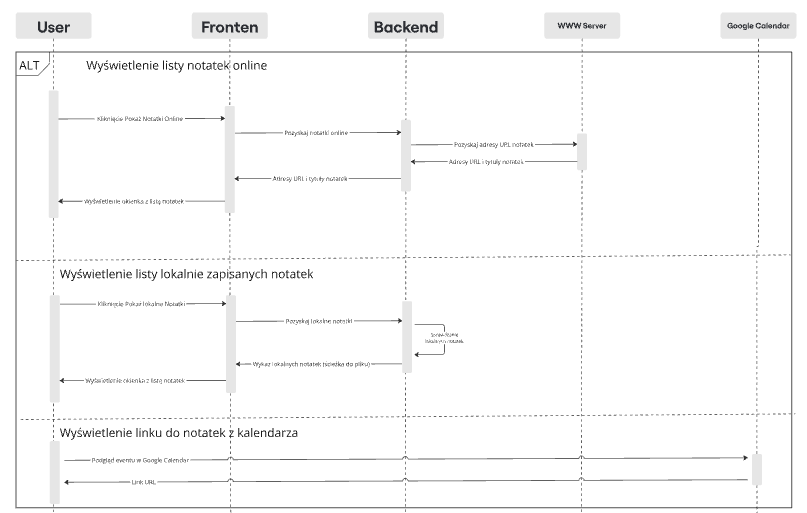
**Struktura aplikacji**

Rysunek 3 Struktura aplikacji przy użyciu ww.narzędzi

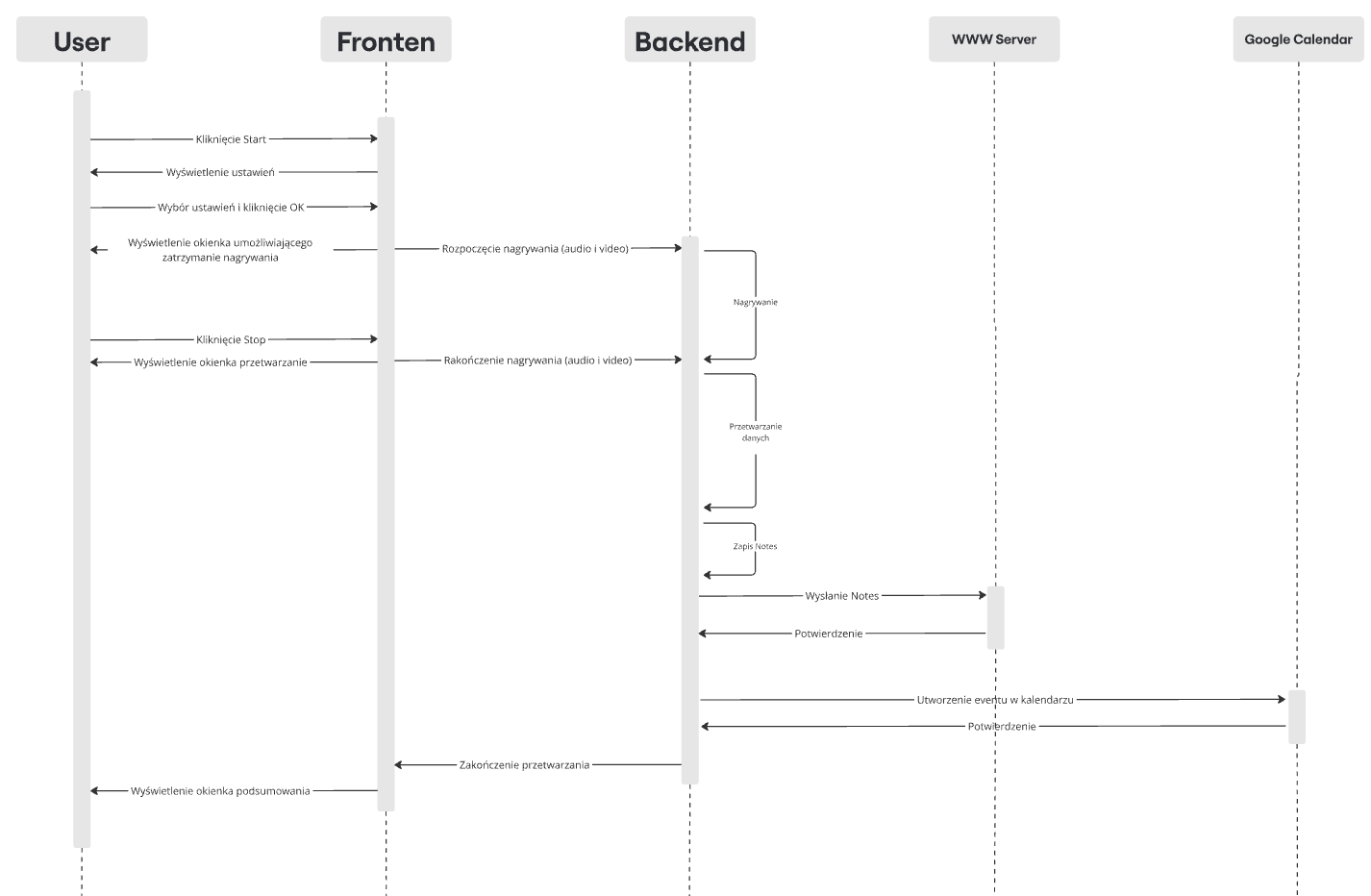


**Diagramy sekwencji**

Rysunek 4 Diagram sekwencji - wyświetlenie listy notatek

****

Rysunek 5 Diagram sekwencji - nagrywanie i przetwarzanie danych

****

Stos technologiczny

**1. Frontend (GUI)**

**Technologia**: Tkinter (Python)  
**Opis**:  
Tkinter to biblioteka GUI dla Pythona, wykorzystywana do tworzenia aplikacji z interfejsem użytkownika. W tym przypadku, GUI aplikacji będzie zaprojektowane za pomocą Tkinter, co umożliwi tworzenie okienek, przycisków, pól tekstowych, menu, a także innych interaktywnych elementów, takich jak mechanizmy start/stop, przeglądanie danych (notatek, wideo), czy ustawienia aplikacji.

**Funkcjonalności GUI**:

* **Mechanizm Start/Stop** – przycisk umożliwiający rozpoczęcie i zatrzymanie transkrypcji mowy.
* **Przeglądanie danych** – umożliwiające użytkownikowi dostęp do zapisów wideo, notatek, czy innych materiałów spotkań.
* **Ustawienia aplikacji** – ustawienia dla użytkownika, takie jak maksymalna ilość miejsca na dysku, jakość nagrań, język spotkania itp.
* **Interaktywność** – prosty interfejs, który umożliwia użytkownikowi łatwe korzystanie z aplikacji, przechodzenie między sekcjami i zarządzanie spotkaniami.

**2. Backend (Serwer WWW)**

**Technologia**: PHP  
**Opis**:  
PHP będzie używane do obsługi serwera WWW aplikacji, zapewniając dostęp do plików i notatek zapisanych w formacie DOCS, TXT. Serwer będzie odpowiedzialny za przechowywanie i udostępnianie plików oraz zarządzanie użytkownikami. Umożliwi również integrację z Google Calendar oraz zapewni interfejs do przeglądania notatek na odpowiednich podstronach.

**Funkcjonalności Backend**:

* **Obsługa plików** – Serwer PHP będzie zarządzał przechowywaniem i udostępnianiem plików spotkań, takich jak transkrypcje w formatach DOCS, TXT, a także zrzuty ekranu.
* **Integracja z Google Calendar** – Serwer będzie współpracował z API Google Calendar, aby umożliwić synchronizację z kalendarzem użytkownika i generowanie linków do odpowiednich notatek.
* **Zarządzanie użytkownikami i uprawnieniami** – Obsługa logowania i autoryzacji użytkowników w aplikacji oraz zarządzanie dostępem do konkretnych notatek.

**3. Integracja z Google Calendar**

**Technologia**: Google Calendar API  
**Opis**:  
Aplikacja będzie zintegrowana z Google Calendar za pomocą API, co umożliwi synchronizację spotkań z kalendarzem użytkownika. Dzięki tej integracji, użytkownik będzie mógł łatwo przeglądać spotkania z poszczególnych dni, a każde spotkanie będzie miało link do odpowiednich podstron z notatkami.

**Funkcjonalności**:

* **Synchronizacja z kalendarzem** – Aplikacja będzie automatycznie dodawała spotkania do Google Calendar oraz generowała linki do odpowiednich notatek w aplikacji.
* **Podstrony notatek** – Po kliknięciu w wydarzenie w kalendarzu, użytkownik będzie przekierowywany do podstrony, na której będą dostępne notatki, transkrypcje i inne materiały związane z danym spotkaniem.

**4. Transkrypcja mowy na tekst + nagrywanie telekonferencji**

**Technologie**: Python (Biblioteki do transkrypcji mowy)  
**Opis**:  
Transkrypcja mowy na tekst będzie wykonywana przy użyciu bibliotek Python, takich jak **Whisper oraz Pyannote.audio**. Te biblioteki pozwolą na przekształcanie nagrań audio z rozmów w tekst, który będzie następnie zapisywany w formacie DOCS lub TXT. Do nagrywania ekranu będzie ( wstępnie ) wykorzystywana biblioteka **pyautogui.**

**Funkcjonalności**:

* **Transkrypcja mowy** – Po zakończeniu spotkania, nagranie audio będzie analizowane i transkrybowane na tekst, który będzie następnie zapisany w odpowiednich plikach.
* **Identyfikacja mówcy** – Opcjonalnie, algorytmy rozpoznawania mówcy (np. **Speaker Diarization**) mogą być używane do identyfikowania mówców w transkrypcji.
* **Jakość transkrypcji** – Użytkownik może ustawić jakość transkrypcji (np. lepsza jakość dla wyraźniejszych nagrań, słabsza dla nagrań o niższej jakości).