**Dokumentacja inżynierii wymagań**

**Członkowie zespołu:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Kompetencje** | **Jakub Kogut** | **Bruno Banaszczyk** | **Antek Lenart** |
| Python | Posiada | Posiada | Posiada |
| Design UI | Posiada | Posiada | Nie posiada |
| UML | Posiada(podstawy) | Posiada(podstawy) | Posiada(podstawy) |
| Testowanie oprogramowania | Nie posiada | Posiada | Nie posiada |

**Pytania do klienta:**

|  |  |  |
| --- | --- | --- |
| **Pytanie** | **Odpowiedź** | **Uwagi** |
| **Jakie mają być obsługiwane formaty plików ?** | Obsługiwane formatu: JPEG, PNG, GIF, MP3, WAV, FLAC, MP4, MOV | Brak |
| **Jak ma wyglądać interfejs ?** | Prosty, intuicyjny, Ładny |  |
| **Jaki ma być maksymalny obsługiwany rozmiar pliku ?** | Rozmiar pliku do 200MB | Czas przetwarzania (200MB) max. 5min |
| **Jakie są wymagania dotyczące platformy ?** | Desktop/Web |  |
| **Jaka jest dozwolona maksymalna utrata jakości pliku ?** | Bez znaczącej utraty jakości ( < 20% ) |  |
| **Czy potrzebny jest backup plików ?** | Nie |  |
| **Czy nowy plik ma nadpisywać stary ?** | Nie, nowy plik ma być generowany jako kopia przesłanego pliku. |  |
| **Jaki jest preferowany język interfejsu użytkownika ?** | Polski |  |

Wszystkie wymagania klienta są możliwe do spełnienia.

**Ustalony format danych wejściowych**

|  |  |
| --- | --- |
| **Plik** | **Format po konwersji** |
| **Format:** JPEG, PNG, GIF, MP3, WAV, FLAC, MP4, MOV  **Rozmiar**: max. 200MB | Wybór formatu z listy |

**Modele prezentujące działanie systemu**

4.1. Modelowanego systemu za pomocą tabeli

|  |  |
| --- | --- |
| **Aktorzy** | Klient, aplikacja do konwersji plików |
| **Opis** | Celem działa tworzonej aplikacji będzie przekonwertowania podanego przez klienta pliku. W razie powodzenia aplikacja zwraca przekonwertowaną kopię przesłanego pliku. W razie niepowodzenia zostanie wyświetlony odpowiedni komunikat. |
| **Dane** | Plik |
| **Wyzwalacz** | Przycisk |
| **Odpowiedź** | Przekonwertowana kopia przesłanego przez użytkownika pliku lub komunikat o błędzie. |
| **Uwagi** | Maksymalny wymiar pliku nie może przekroczyć 200MB. Maksymalny czas obsługi pliku 200MB nie przekroczy 5 minut. |

Rys. 4.2 Diagram przypadków użycia

Obraz zawierający tekst, zrzut ekranu, diagram, krąg

Opis wygenerowany automatycznie

Rys. 4.3 Diagram przepływu danych

Obraz zawierający tekst, zrzut ekranu, czarne, design

Opis wygenerowany automatycznie

Rys. 4.4 Diagram sekwencyjny UML

Obraz zawierający tekst, zrzut ekranu, diagram, linia

Opis wygenerowany automatycznie

**Projekt architektury opracowanej aplikacji ( Architektura potokowa )**

Rys. 4.5 Projekt architektury opracowywanego systemu

s

Obraz zawierający tekst, zrzut ekranu, Czcionka, diagram

Opis wygenerowany automatycznie

Architektura aplikacji do konwersji typów plików składa się z trzech kluczowych komponentów, które współdziałają, aby zapewnić użytkownikom efektywne narzędzie do przekształcania plików w różne formaty.

Pierwszym elementem systemu jest intuicyjny interfejs użytkownika, który umożliwia łatwe przesyłanie plików oraz zrozumienie dostępnych opcji.

Drugim kluczowym komponentem jest proces weryfikacji danych, który sprawdza, czy typ pliku jest obsługiwany i czy jego rozmiar nie przekracza ustalonych limitów. W przypadku błędów użytkownik otrzymuje jasne komunikaty, co zapobiega dalszym problemom.

Trzecim komponentem jest algorytm konwersji, odpowiedzialny za elastyczne i szybkie przekształcanie plików do wybranego formatu. Musi on obsługiwać różnorodne formaty wejściowe, co zapewnia elastyczność całego systemu.

1. Sugerowany język implementacji: Python

Uzasadnienie: Język z rozbudowaną społecznością i z gotowymi, sprawdzonymi rozwiązaniami w temacie konwersji plików. Pomaga on także przyspieszyć proces wprowadzania oprogramowania do użytku przez użytkownika końcowego.