**Analiza Wymagań**

**Cel Projektu**

**Celem projektu "Face Comparison App" jest opracowanie narzędzia do porównywania twarzy, które umożliwia użytkownikom efektywne porównywanie różnych obrazów twarzy.**

**Funkcjonalności**

1. **Importowanie obrazów:**
   * **Umożliwienie użytkownikom wybór dwóch obrazów do porównania za pomocą przycisków w interfejsie użytkownika.**
2. **Algorytm porównywania obrazów:**
   * **Wykrywanie twarzy na obu obrazach.**
   * **Ekstrakcja cech twarzy z wykrytych twarzy.**
   * **Obliczanie podobieństwa między cechami twarzy.**
   * **Interpretacja wyniku podobieństwa w celu określenia, czy twarze są podobne czy różne.**
3. **Interfejs użytkownika:**
   * **Prosty i intuicyjny interfejs użytkownika do wyboru obrazów i wyświetlania wyników.**
   * **Wyświetlanie postępu przetwarzania za pomocą paska postępu.**
   * **Wyświetlanie wyników porównania w czytelny sposób.**

**Struktura Danych**

* **Listy i Tuples:**
  + **Używane do przechowywania zestawów danych, takich jak lokalizacje twarzy.**
* **Zbiory:**
  + **Używane do przechowywania unikalnych kodowań twarzy w celu uniknięcia duplikatów.**
* **Słowniki:**
  + **Używane do logowania błędów i informacji o przetwarzaniu obrazów.**

**Interakcje Pomiędzy Komponentami**

1. **Interakcje użytkownika z interfejsem:**
   * **Użytkownik wybiera obrazy za pomocą przycisków "Select Image 1" i "Select Image 2".**
   * **Użytkownik inicjuje porównanie twarzy za pomocą przycisku "Compare Faces".**
2. **Przetwarzanie obrazów:**
   * **Klasa FaceComparison ładuje i przetwarza obrazy.**
   * **Klasa FaceComparisonThread zarządza przetwarzaniem w osobnym wątku, aby uniknąć zawieszenia interfejsu użytkownika.**
3. **Wykrywanie i porównywanie twarzy:**
   * **Metody detect\_face i extract\_features wykrywają twarze i ekstraktują cechy.**
   * **Metoda calculate\_similarity oblicza podobieństwo między cechami twarzy.**
   * **Metoda compare\_faces zarządza całym procesem porównania.**

**Podstawy Python**

* **Importowanie bibliotek:**
  + **Używanie import do załadowania niezbędnych modułów, takich jak face\_recognition, cv2, numpy, logging, PyQt5.**
* **Klasy i Metaklasy:**
  + **Definiowanie klas FaceComparison, FaceComparisonThread, FaceComparisonUI.**
  + **Używanie konstruktorów (\_\_init\_\_) do inicjalizacji instancji klasy.**
  + **Dziedziczenie z QThread i QWidget dla integracji z PyQt5.**
* **Deskryptory i Dekoratory:**
  + **Używanie dekoratorów @pyqtSlot do definiowania slotów w PyQt5.**
  + **Używanie sygnałów (pyqtSignal) do komunikacji między wątkami a interfejsem użytkownika.**