

# Dokumentacja UML

---

## Aplikacja do ekstrakcji tekstu

Jakub Sośnicki, Bartłomiej Stachurski

---

### Spis treści:

1. Wymagania funkcjonalne i нефunkcjonalne
  2. Diagram przypadków użycia wraz z ich szczegółowym opisem
  3. Opis klas i metod
  4. Tutorial
-

## **Cel programu**

Głównym celem programu jest umożliwienie łatwej digitalizacji tekstu pisanego odręcznie. Program ma na celu zamianę notatek i ogólnie pojętego tekstu pisanego pismem odręcznym na formę cyfrową za pomocą kamery w urządzeniu mobilnym i specjalnie dobranych narzędzi OCR i LLM zapewniając możliwość łatwej obróbki i archiwizacji w wersji cyfrowej.

## **Wytyczne dotyczące działania programu**

Aplikacja jako system mobilny jest uruchamiana na systemie Android w wersji 9 i nowszych. Po uruchomieniu aplikacji po raz pierwszy użytkownik jest proszony o możliwość dostępu aplikacji do pamięci urządzenia w celu zapisu i odczytu danych w formacie zdjęć. Następnie użytkownik ma możliwość zrobienie zdjęć swoich notatek w celu ekstrakcji i obróbki tekstu zawartego na zdjęciu. Po zakończeniu skanowania i obróbki użytkownik zobaczy na ekranie aplikacji tekst w formacie .md z wynikiem działania aplikacji.

## **Opis architektury aplikacji**

Program podczas ekstrakcji tekstu wykonuje trzy kluczowe etapy.

- Program wykorzystuje bibliotekę OpenCV
- Wykonanie algorytmu OCR
- Obróbka tekstu za pomocą algorytmu LLM

## **1. Wymagania funkcjonalne i нефункционалне**

### **a. Wymagania funkcjonalne:**

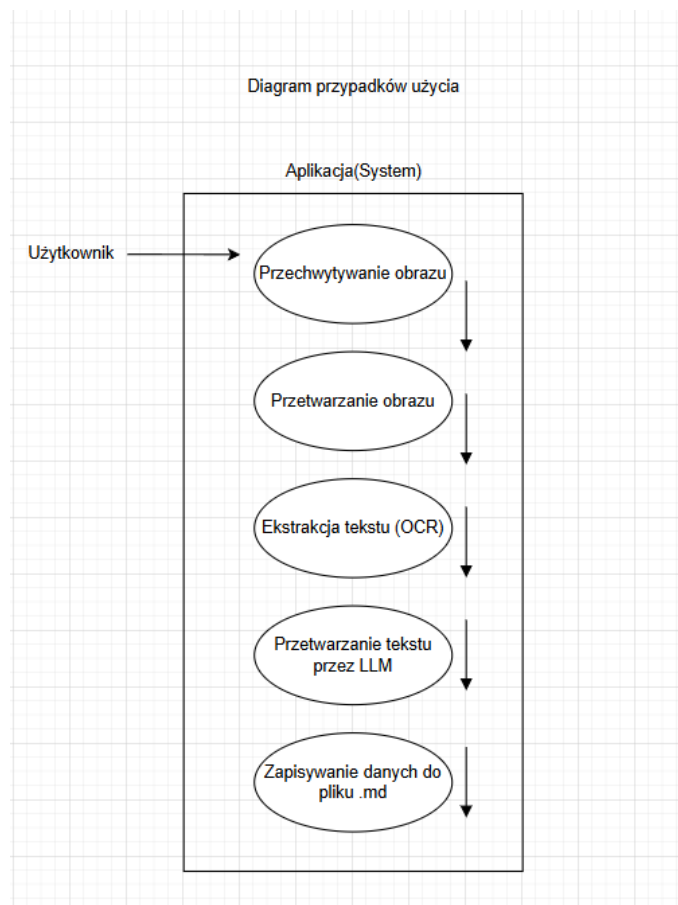
- Przechwytywanie obrazu:
  - Aplikacja mobilna umożliwia użytkownikowi zrobienie zdjęcia dokumentu lub załadowanie istniejącego zdjęcia.
- Przetwarzanie obrazu:
  - Wstępne przetwarzanie obrazu, w tym zastosowanie filtru czarno-białego OpenCV
- Ekstrakcja tekstu:
  - Rozpoznawanie tekstu z obrazu za pomocą narzędzia OCR Google ML Kit API
  - Zapis wyekstrahowanego tekstu w formacie .txt.
- Przetwarzanie tekstu przez LLM:
  - Wystanie wyekstrahowanego tekstu do modelu językowego LLM OpenAI
- Wypis danych:
  - Możliwość skopiowania lub pobrania tekstu

### **b. Wymagania нефункционалне:**

- Wydajność:
  - Czas przetwarzania obrazu i tekstu nie powinien przekraczać 5 sekund na urządzeniach średniej klasy.
- Interfejs użytkownika:
  - Intuicyjny interfejs dla użytkowników bez potrzeby zaawansowanej wiedzy technicznej.
- Przenośność:
  - Kompatybilność z urządzeniami z systemem Android 9.0 w górę

## 2. Diagram przypadków użycia wraz z ich szczegółowym opisem

### a. Diagram przypadków użycia



### b. Szczegółowy opis przypadków użycia

#### Przypadek użycia: Przechwytywanie obrazu

- Opis: Użytkownik robi zdjęcie lub wybiera istniejące zdjęcie w aplikacji.
- Aktorzy: Użytkownik
- Kroki scenariusza podstawowego:
  1. Użytkownik otwiera aplikację mobilną.
  2. Użytkownik wybiera opcję „Zrób zdjęcie” lub „Załaduj zdjęcie z galerii”.
  3. System zapisuje wybrane zdjęcie do dalszego przetwarzania.
- Warunek początkowy: Użytkownik ma zainstalowaną aplikację.
- Warunek końcowy: Zdjęcie jest gotowe do przetwarzania.

**Przypadek użycia: Przetwarzanie obrazu**

- Opis: System stosuje filtr czarno-biały, aby poprawić jakość przetwarzania OCR.
- Aktorzy: System.
- Kroki scenariusza podstawowego:
  1. System odbiera zdjęcie.
  2. System przetwarza zdjęcie, stosując filtr czarno-biały.
  3. Przetworzony obraz jest przekazywany do procesu OCR.
- Warunek początkowy: Zdjęcie zostało wybrane lub zrobione przez użytkownika.
- Warunek końcowy: Zdjęcie jest przetworzone.

**Przypadek użycia: Ekstrakcja tekstu (OCR)**

- Opis: System wyodrębnia tekst z przetworzonego obrazu za pomocą narzędzia OCR.
- Aktorzy: System.
- Kroki scenariusza podstawowego:
  1. System odbiera przetworzone zdjęcie.
  2. System wykonuje operację OCR, wyodrębniając tekst.
  3. Wyekstrahowany tekst jest zapisany w pliku tymczasowym.
- Warunek początkowy: Obraz został przetworzony.
- Warunek końcowy: Tekst został wyodrębniony.

**Przypadek użycia: Przetwarzanie tekstu przez LLM**

- Opis: System poprawia jakość wyekstrahowanego tekstu za pomocą modelu językowego.
- Aktorzy: System.
- Kroki scenariusza podstawowego:
  1. System wysyła wyekstrahowany tekst do modelu LLM.
  2. Model LLM poprawia tekst i generuje sformatowaną wersję.
  3. Poprawiony tekst zostaje zapisany w pliku.
- Warunek początkowy: Tekst został wyekstrahowany.
- Warunek końcowy: Tekst został poprawiony i sformatowany.

**Przypadek użycia:** Zapis danych do pliku .md

- Opis: System zapisuje poprawiony tekst w pliku .md.
- Aktorzy: System.
- Kroki scenariusza podstawowego:
  1. System odbiera poprawiony tekst od modelu LLM.
  2. System zapisuje tekst w formacie .md do lokalnego katalogu użytkownika.
- Warunek początkowy: Tekst został poprawiony i sformatowany.
- Warunek końcowy: Plik .md jest gotowy do użytku.

### 3. Opis klas i metod

Mamy 3 klasy:

#### **MainActivity:**

Odpowiada za tworzenie activity\_main.xml

- protected void onCreate
- private void openGallery -> otwiera galerię do wyboru zdjęcia
- public void onRequestPermissionsResult -> obsługa dla żądań uprawnień
- private void openCameraActivity -> przejście do CameraActivity z bitmapą z aparatu
- private void openCameraActivity2 -> przejście do CameraActivity z stringiem z url do imagePath
- protected void onActivityResult -> obsługa wyniku wybrania plików z galerii
- private boolean isNetworkAvailable -> sprawdza aktualne połączenie z internetem

#### **CameraActivity:**

Odpowiada za tworzenie activity\_camera.xml

- Protected void onCreate
- Protected void onActivityResult -> obsługa zrobionego zdjęcia
- Public void recognizeTextFromImage -> metoda do rozpoznawania tekstu ze zdjęcia
- Public void recognizeTextFromImageWithLLM -> metoda do rozpoznawania tekstu ze zdjęcia + korekta modelem językowym
- Private void saveTextToFile -> metoda zapisuje skopiowany tekst do pamięci urządzenia

## OpenAIRequestTask:

Odpowiada za obsługę zapytań do API OpenAI

- Public OpenAIRequestTask -> konstruktor
- Protected String doInBackground -> obsługa zapytania
- Protected void onPostExecute -> metoda dla zapytania POST

## 4. Tutorial

### NoteScan: Digitalizacja tekstu pisanego

NoteScan to prosta i intuicyjna aplikacja mobilna, która pozwala przekształcić odręczne notatki w format cyfrowy. Oto jak z niej korzystać:

#### Krok 1: Uruchomienie aplikacji

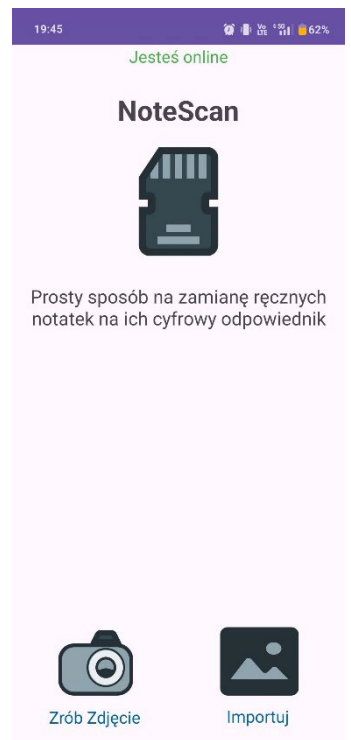
Po uruchomieniu aplikacji zobaczysz prosty interfejs z logo NoteScan i ikoną karty pamięci. Na ekranie głównym wyświetla się informacja "Prosty sposób na zamianę ręcznych notatek na ich cyfrowy odpowiednik".

#### Krok 2: Wybór metody wprowadzania

Na dole ekranu masz dwie opcje wprowadzania materiału:

"Zrób Zdjęcie" - pozwala zrobić zdjęcie notatek bezpośrednio z aplikacji

"Importuj" - umożliwia wybranie istniejącego zdjęcia z galerii



### Krok 3: Przetwarzanie tekstu

Po wykonaniu lub wybraniu zdjęcia, masz do dyspozycji dwie metody przetwarzania:

"OCR z LLM" - zaawansowane rozpoznawanie tekstu z wykorzystaniem modeli językowych

"Skopiuj tekst" - prosta konwersja do formatu tekstowego

Na górze ekranu w lewym rogu mamy przycisk do powrotu do menu głównego, w prawym natomiast możliwość pobrania tekstu



#### **Ważne informacje:**

Aplikacja wymaga połączenia z internetem (widoczny wskaźnik "Jesteś online"/"Brak połączenia z internetem")

Stan połączenia jest wyświetlany na górze ekranu

Aplikacja działa w trybie portretowym

Interfejs jest dostępny w języku polskim

Widoczny wskaźnik baterii i siły sygnału w górnej części ekranu

#### **Wskazówki dla lepszych rezultatów:**

Upewnij się, że masz dobre oświetlenie podczas robienia zdjęć

Trzymaj telefon stabilnie

Umieść tekst na kontrastowym tle

Sprawdź, czy masz stabilne połączenie internetowe przed rozpoczęciem pracy

NoteScan to idealne narzędzie dla studentów, profesjonalistów i wszystkich, którzy chcą szybko przenieść swoje odręczne notatki do formatu cyfrowego.