

# **APLIKACJA KONSOLOWA**

## **„CVRAPORT”**

### **Spis Treści**

#### **1. Wprowadzenie**

- 1.1. Cel dokumentu
- 1.2. Zakres produktu
- 1.3. Definicje, akronimy i skróty
- 1.4. Referencje, odsyłacze do innych dokumentów
- 1.5. Krótki przegląd

#### **2. Ogólny opis**

- 2.1. Relacje do bieżących projektów
- 2.2. Relacje do wcześniejszych i następnych projektów
- 2.4. Relacje do innych systemów
- 2.3. Ustalenia dotyczące środowiska
- 2.5. Ogólne ograniczenia
- 2.6. Założenia i zależności
- 2.7. Przykłady użycia
  - a.) analizy exif
  - b.) analizy graficznej

#### **3. Specyficzne wymagania**

- 3.1. Wymagania funkcjonalne (funkcje systemu)
  - a.) ALGORYTMY – analiza exif
  - b.)ALGORYTMY – analiza graficzna
  - c.)ALGORYTMY – umożliwiające zarządzanie bazą danych
  - d.)ALGORYTMY – analiza zgromadzonych danych
  - e.)USTALENIA DOTYCZĄCE MENU
  - f.)USTALENIA DOTYCZĄCE BAZY DANYCH JSON
  - \*.) analizy exif

\*) analizy graficznej

g.)USTALENIA DOTYCZĄCE INTERFEJSU

f.)FORMULARZE

f.)SKRÓTY Klawiszowe

3.2. Wymagania niefunkcjonalne (ograniczenia).

## 4.Dodatki

a.)Zalecane wymagania systemowe

b.)Wymagania odnośnie testowania programu

c.)Harmonogram prac

# Wprowadzenie

## Cel dokumentu

1.1 Zamieszczony niżej dokument zawiera informacje o założeniach projektu. Przedstawia analizę jego wymagań jakie są konieczne do prawidłowego jego działania jak i dane techniczne dla developerów mające na celu pomoc w jego realizacji. Dokument ten został napisany z myślą o potencjalnych jego użytkownikach, projektantach, programistach, osobach zajmujących się jego testowaniem oraz autorach jego dokumentacji użytkowej.

## Zakres produktu

1.2 W czasach rozwoju technologii produkujemy coraz więcej informacji i danych, które w miarę narastających potrzeb i rozwijających się technologii można coraz szerzej analizować. Przykładem typu jednych z nich są dane exif zapisywane w nagłówkach plików graficznych, które zawierają w sobie np. takie dane jak GPS, daty wykonanych zdjęć, model urządzenia którymi zdjęcia zostały wykonane a nawet czasem podstawowe dane identyfikacyjne jego autorów. Jeśli te dane zapisać do bazy danych, a następnie umieścić je na mapie można ustalić historię przemieszczania się jakiegoś modelu urządzenia. Zapisywane w tej postaci danych mogą stanowić również nie lada zagrożenie jeśli chcielibyśmy podzielić się naszymi świeżo zrobionym zdjęciami z naszymi przyjaciółmi. Zawsze może wtedy znaleźć się ktoś kto będzie chciał je wykorzystać w niekorzystny dla nas sposób. Warto jest więc zadbać o ich bezpieczeństwo i mieć nad nimi pełną kontrolę sprawdzając je zanim wyślemy je do internetu, abyśmy przez przypadek nie pokazali za dużo drugiej osobie.

Takie dane analizują również wszelkie służby specjalne, stanowią dla nich jedno z wielu źródeł informacji o poszukiwanej lub śledzącej przez nich osobie.

Dlatego właśnie istnieje potrzeba, abyśmy w pełni kontrolowali to co wrzucamy do internetu lub przechowujemy na dyskach. Ten właśnie program ma nam to umożliwić.

Dodatkowo możemy być zainteresowani na jakich zdjęciach się znajdujemy lub znajduje się poszukiwana przez nas osoba. Może chcielibyśmy sprawdzić jaki mamy z kimś związek, czy spotkaliśmy tą osobę już wcześniej. Analiza związana z porównywaniem obrazów i analizy danych exif może dać nam szybką odpowiedź. Gdybyśmy chcieli ręcznie jej szukać było by to bez sensu. Po prostu nie było by to możliwe, ale komputery dają nam właśnie taką możliwość..

## Definicje, akronimy i skróty

**1.3 EXIF** – standard metadanych dla plików w których zapisane są między innymi dane GPS  
kernel – jądro linuxa

**JSON** – baza danych dla programów pisanych między innymi w C stworzona przez firmę oracle.

**Opencv** – biblioteka umożliwiająca analizę obrazów, umożliwiająca między innymi wejście w porównywanie pojedynczych pikseli zdjęć.

**SAS** – Oprogramowanie służące do analizy danych.

**MIT** – Licencja na której łamach można bezprawnie edytować i kopiować kod oraz go sprzedawać.

**Open-source** – otwarte oprogramowanie, to znaczy, że możliwy jest wgląd do kodu źródłowego programu

**BASH** – powłoka systemowa unix umożliwiająca uruchamianie linii poleceń

## Referencje, odsyłacze do innych dokumentów

**1.4**

Odwołanie do literatury

[https://pl.wikipedia.org/wiki/Exchangeable\\_Image\\_File\\_Format](https://pl.wikipedia.org/wiki/Exchangeable_Image_File_Format) |(EXIF)

<https://pl.wikipedia.org/wiki/Bash>(BASH)

## Krótki przegląd dokumentu

**1.5**

## **Ogólny opis**

### **2.1 Relacje do bieżących projektów**

Istnieje obecnie wiele projektów analizujących dane jak np. SAS, ale są one dostępne jedynie za wysoką opłatą co wiąże się z znacznym zawężeniem grona osób jego użytkowników. Niniejszym projekt będzie zgodny z ideą ogólnie przyjętego open-source i będzie korzystał jedynie z otwarto źródłowych bibliotek opublikowanych na licencji MIT. Budżet i rozwój projektu zapewnią prywatni dobroczyńcy, którzy będą zainteresowani dalszym rozwijaniem owego projektu.

### **2.2 Relacje do wcześniejszych i następnych projektów**

Projekt będzie mógł również stanowić bazę dla innych bardziej rozbudowanych projektów w przyszłości służących do analizy danych zapisanych między innymi w metadanych „exif” jak i programów analizujących w przeróżny sposób zgromadzone przez nas informacje.

### **Relacje do innych systemów**

**2.3** Aktualnie na rynku istnieją podobne aplikacje umożliwiające wgląd do danych exif, ale ich funkcjonalności ograniczają się zazwyczaj tylko do analizy danych pojedynczych plików graficznych. Połączenie analizy danych exif z analizą plików graficznych z porównywaniem i znajdowaniem na nich konkretnych osób daje nam nieskończone możliwości.

### **Ustalenia dotyczące środowiska**

**2.4** Program będzie stworzony na otwartą platformę Linuksa. Powinien być kompatybilny przede wszystkim z systemami opartymi na debianie takimi dystrybucjami jak np. mint, Ubuntu, Debian, Fedora. Korzystanie z aplikacji będzie również możliwe na dystrybucjach, które są kompatybilne z Archem jak np. Manjaro, Arch, dzięki dostępności jej kodu źródłowego na platformie github.

Zaprezentowana tutaj program dedykowany jest zarówno do użytkowania profesjonalnego jak i amatorskiego. Dzięki prostemu interfejsowi graficznemu będzie możliwe łatwe uruchomienie programu na dużych serwerach w celu analizy zawartych na nich informacji.

## Ogólne ograniczenia

**2.5** System musi spełniać wymagania odnośnie ustawy o ochronie danych osobowych, użytkownik decydujący się korzystać z niniejszego oprogramowania musi samemu zadbać o to, aby przypadkiem nie wykorzystać jej w sposób niezgodny z prawem. Natomiast jej użytkownik powinien uważać, aby nie otworzyć jej na środowisku do którego nie posiada pełnych praw autorskich. To znaczy w miejscach gdzie prawa do analizy i posiadania informacji osobowych mogą należeć już do zupełnie innej osoby.

## Założenia i zależności

**2.6** Program ma być napisany w języku C, zgodnie z normą C99. Interfejs programu zbudowany ma być w oparciu o bibliotekę ncurses, a analiza graficzna ma być przeprowadzana w oparciu o bibliotekę opencv. Program będzie łączył się z bazą danych JSON w celu zapisu i wyciągnięcia potrzebnych danych z tabeli utworzonych przez programistę według schematu zamieszonego na końcu tego dokumentu. Program będzie jedynie generował gotowe polecenia BASHA, aby wykorzystać zainstalowane algorytmy już w systemie. Komunikacja między bashem czyli generowanie poleceń których wyniki były zapisywane w tworzonych plikach graficznych będzie odbywała się za pomocą formularzy.

## Przykłady użycia

### 2.7

#### ANALIZA EXIF

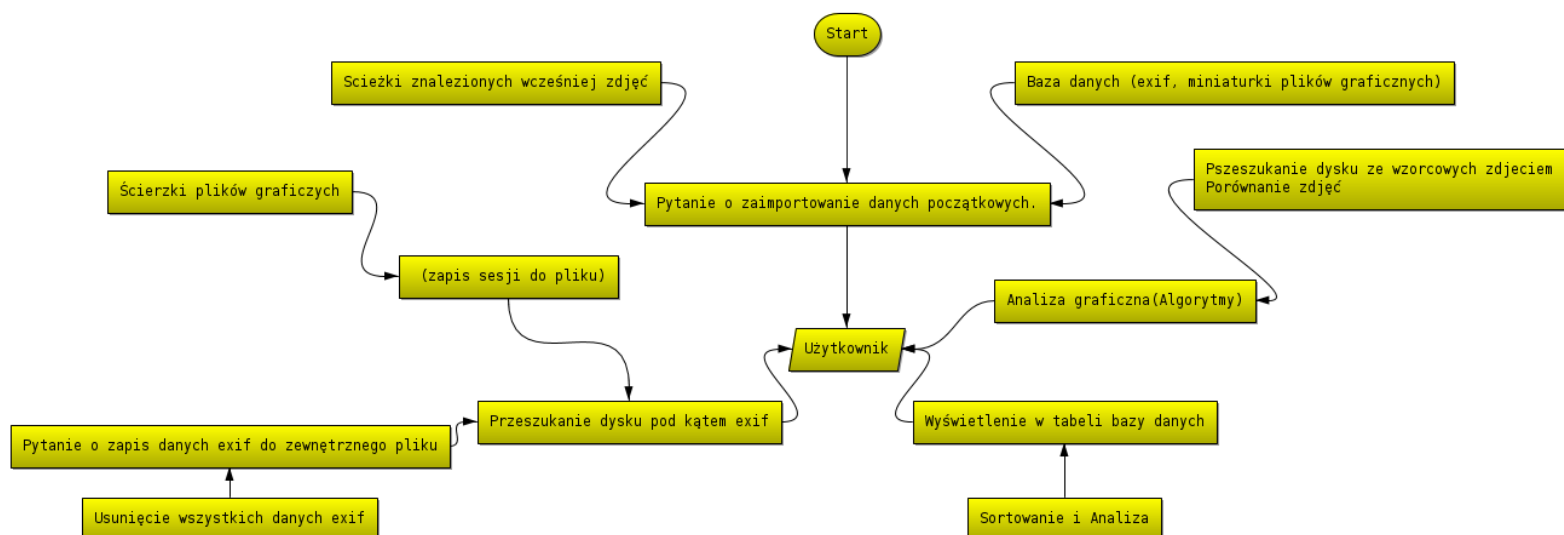
##### Wariant 1:

Użytkownik chce podglądnąć czego może się dowiedzieć o sobie na podstawie zdjęć które przechowuje u siebie na dysku. Chcąc zmniejszyć ryzyko wycieku swoich danych przy wrzucaniu przy różnych okazjach zdjęć do internetu decyduje się usunąć wszystkie dane EXIF znalezione na dysku i zapisać je do zewnętrznego bezpiecznego pliku jakby chciał jeszcze kiedyś do nich wrócić.

##### Wariant 2

Użytkownik chce przypomnieć sobie co robił w wakacje. Może w tym wypadku.

- a.) Wczytać plik z danymi exif, który utworzył wcześniej usuwając dane exifi wyświetlić je na mapie.
- b.) Przeszukać dysk w poszukiwaniu danych exif i nanieść je na mapę.



Adnotacje do powyżej fotografii:

- Przeszukiwanie dysku pod kątem danych EXIF == analiza exif(Algorytmy) z pt. 3.1 „Analiza EXIF”.
- Wyświetlenie w tabeli bazy danych == Analiza danych(algorytmy) z pt. 3.1 „Analiza zgromadzonych danych”

## ANALIZA GRAFICZNA

### Wariant 3

Użytkownik chce sprawdzić na jakich zdjęciach znajduje się jego twarz. Będzie mógł dodać zdjęcie wzorcowe, które posłuży do znalezienia podobnych osób na zdjęciach zapisanych na dysku. (jego twarz) Odpowiedni algorytm zwróci mu ścieżki zdjęć odpowiadającemu jego wyszukiwaniu.

# Specyficzne wymagania

## 3.1. Wymagania funkcjonalne(funkcje systemu)

0: Algorytm importujący zapisanie danych.

### a.)ALGORYTMY - ANALIZA EXIF

1. Przeszukanie dysków w poszukiwaniu zdjęć. Wykorzystanie programu „find”.

[https://pl.wikipedia.org/wiki/Binarne\\_drzewo\\_poszukiwań](https://pl.wikipedia.org/wiki/Binarne_drzewo_poszukiwań)

2. Zapisanie znalezionych ścieżek do pliku txt. Wykorzystanie programu „find”.

3. Wyodrębnianie danych exif z pojedynczych plików graficznych. Wykorzystanie programu „exif”.

4. Zapisanie powyższych danych exif do tabeli bazy danych JSON, takich jak:

Konieczna samodzielna implementacja.

a.) model urządzenia

b.) data wykonania zdjęcia

c.) imię (jeśli jest)

d.) nazwisko (jeśli jest)

e.) lokalizacja GPS(jeśli jest)

f.) dodatkowo(według życzenia programisty)

5. Algorytm usuwający dane exif z plików graficznych znajdujących się na dysku. **Wykorzystanie programu find i exif.**

### b.)ALGORYTMY - ANALIZA GRAFICZNA (opencv)

1. Możliwość identyfikacji osób znajdujących się na zdjęciach w taki sposób, że algorytm porównuje znalezione zdjęcia ze wzorcami zdjęć które zapisane są w jednym z utworzonym przez program katalogu. Natomiast wszystkie te zdjęcia są podlinkowane z bazy danych JSON, która przechowuje profile poszukiwanych osób oraz linki do zapisanych w tym katalogu zdjęć.

2. Identyfikacja i zliczanie osób towarzyszących badanej przez nas osobie.

3.Stwierdzenie czy na znalezionych zdjęciach znajduje się ta sama interesująca nas osoba i posortowanie i następnie przedstawienie tych danych według daty i lokalizacji GPS na mapie.

4.Na jakich zdjęciach które mamy na dysku znajduje się poszukiwana przez nas osoba.

### c.)ALGORYTMY – ZARZĄDZANIE BAZĄ DANYCH

- 1.Algorytm umożliwiający dodawanie tabel.
- 2.Algorytm umożliwiający zmianę rekordów w tabeli.
- 3.Algorytm umożliwiający dodawanie nowych rekordów do tabeli w bazie danych.
- 4.Algorytm usuwający tabele z bazy danych.
- 5.Algorytm usuwający pojedyncze rekordy z tabeli w bazie danych.

**W przyszłości można dodać:**

- 1.Połączenie się z serwerem przez SSH i wykonanie na nim potrzebnych obliczeń przy porównaniu zdjęć i szukaniu naszej osoby.
- 2.Wykorzystanie karty graficznej do analizowania danych.

**d.)ALGORYTMY - ANALIZA ZGROMADZONYCH DANYCH**

1.Zapisane dane w bazie danych JSON należy wyświetlić je w tabeli z możliwością ich sortowania według.

- a.) modelu urządzenia
- b.) daty wykonania zdjęcia
- c.) imienia (jeśli jest)
- d.) nazwiska (jeśli jest)
- e.) lokalizacji GPS(jeśli jest)

**e.)USTALENIA DOTYCZĄCE MENU**

Opcja otwierająca się automatycznie przy uruchamianiu programu wygląda następująco. Jest to możliwość importu wcześniej zapisanych danych, aplikacja pyta użytkownika czy zaimportować dane z wcześniejszych sesji.

0: Czy zaimportować wcześniej zapisane dane?

tak/nie

- 1.Importuj pliku z ścieżkami do plików graficznych.
- 2.Import pliku bazy danych JSON z danymi exif.

Struktura drzewa:

1: Analiza EXIF

- 1.Przeszukanie danych dysku w poszukiwaniu danych exif.
- 2.Zapis znalezionych ścieżek do pliku.



- 3. Usunięcie danych exif
- 2: Analiza graficzna
  - 1. Znajdź osobę.
- 3: Analiza zgromadzonych danych
  - 1. Wyświetl tabelę z danymi posortowanymi jako.
    - a.) modelu urządzenia
    - b.) daty wykonania zdjęcia
    - c.) imienia (jeśli jest)
    - d.) nazwiska (jeśli jest)
    - e.) lokalizacji GPS (jeśli jest)
    - f.) dodatkowo według uznania
  - 2. Analiza danych exif na mapie.
  - 3. Wyniki analizy graficznej
  - 4. Zarządzaj bazą danych.
  - 5. Pomoc
- 4. Exit

#### **f.) USTALENIA DOTYCZĄCE BAZY DANYCH JSON**

**Dla analizy EXIF** tworzymy bazę danych w której schemat wygląda następująco:

Tworzymy tabelę z następującymi kolumnami:

- a.) model urządzenia
- b.) data wykonania zdjęcia
- c.) imię (jeśli jest)
- d.) nazwisko (jeśli jest)
- e.) lokalizacja GPS (jeśli jest)
- f.) dodatkowo (według życzenia programisty)

#### **g.) USTALENIA DOTYCZĄCE INTERFEJSU**

Interfejs ma być zbudowany w oparciu o bibliotekę ncurses. Opcje menu mają być wyświetlane na środku małego ekranu o wielkości  $\frac{1}{4}$  naszego monitora o minimalnej rozdzielności monitora (480x320) na szarym tle w konsoli terminala. Pozostała część ekranu ma być uzupełniona niebieskim tłem. Wybranie opcji z menu ma wykonywać algorytmy opisane w pt. 3.1 tej specyfikacji.

Komunikaty programu typu tak/nie mają się wyświetlać w prostym szarym oknie o wielkości  $\frac{3}{4}$  rozdzielczości naszego monitora.

**Dla analizy graficznej** tworzymy następującą tabelę.

- a.) profile\_jpg (tylko ścieżka do pliku jpg w który wrzucamy same zdjęcia)
- b.) imię
- c.) nazwisko

#### **f.) FORMULARZE**

Są konieczną częścią interfejsu graficznego gdyż to właśnie w nich użytkownik wprowadza parametry potrzebne do prawidłowego działania owego systemu ich schemat wygląda następująco:

##### **1.Przeszukanie dysku i wyciągnięcie exif.**

- a.)Pole do wprowadzenia katalogu w którym zostanie przeprowadzone wyszukiwanie.
- b.)Pole w którym zostaje podane hasło do roota.

##### **2.)Usuwanie danych exif z pliku**

- a.)Pole w którym użytkownik podaje katalog z którego zostaną usunięte dane exif/
- b.)Pole w którym zostaje podane hasło do roota.

#### **g.)SKRÓTY KLAWISZOWE**

**:w – zapisz ogólny stan programu**

**:x – zapisz i wyjdź z programu**

**:q – wyjdź natychmiast bez zapisu**

#### **3.2. Wymagania нефunkcjonalne (ograniczenia).**

1. Zdjęcia które będą podlinkowane z bazy danych json na potrzeby analizy graficznej, które są zapisane w utworzonym przez nas katalogu muszą być w rozdzielności 320x240.

2. Zdjęcia będące profilami twarzy muszą być wykonane tak jak do dowodu i zapisane w przeznaczonym dla nich katalogu.

3. Program nie będzie miał możliwości bezpośredniego podglądu plików graficznych. W tym celu program będzie kopiował i konwertował pliki graficzne do jednego katalogu, który będzie utworzony na potrzeby konkretnej analizy graficznej. W celu podglądu plików graficznych jeśli to będzie możliwe program uruchomi zewnętrzną aplikację np. gwenview, która otworzy się tylko w przypadku gdy użytkownik ma uruchomione środowisko graficzne. W przeciwnym przypadku wyświetli się komunikat błędu z informacją w jaki sposób można ten problem rozwiązać.

## **DODATKI**

### **Zalecane wymagania systemowe:**

Procesor: core i7 3gen

Ram: 8GB

System operacyjny: Linux oparty na debianie.

Pojemność dysku: 240 GB

## **HARMONOGRAM PRAC**

1. 17-24 listopad - budowa interfejsu systemu podstawowego menu
2. 24-1 grudnia – budowa interfejsu systemu oraz menu
3. 1-8 grudnia – podłączenie pod interfejs i zaprojektowane formularze algorytmów find
4. 8-15 grudnia – podłączenie pod interfejsu oraz zaprojektowane formularze algorytmów exif
5. 15-22 grudnia- budowa interfejsu zarządzania bazą danych
6. 22-29 grudnia - podłączenie bazy danych JSON, algorytmy sortujące dane
7. 29-5 stycznia- budowa interfejsu analizy graficznej
8. 5-12 stycznia – podłączenie algorytmów opencv

12 stycznia – wrzucenie napisanego oprogramowania na forum [forum.pasja-informatyki.pl](http://forum.pasja-informatyki.pl) w celu przeprowadzenia pełnego testu oprogramowania, ten czas do sesji jest także przeznaczony na naprawę ewentualnych nieciągłości