

# Dokument koncepcyjny

---



AKADEMIA GÓRNICZO-HUTNICZA  
IM. STANISŁAWA STASZICA  
W KRAKOWIE

Autorzy: Anna Marciniec, Mateusz Najdek, Yurii Vyzhha, Mateusz Woś

## Spis treści

---

	0
<b>1. Opis problemu</b>	<b>1</b>
1.1. Lista potrzeb	2
1.2. Lista możliwości	2
1.3. Podsumowanie	3
<b>2. Wizja rozwiązania</b>	<b>4</b>
<b>3. Koncepcja rozwiązania</b>	<b>6</b>
3.1. Wprowadzenie	6
3.2. Architektura systemu	6
3.3. Moduły:	8
3.4. Zakres odpowiedzialności modułów:	9
3.5. Współpraca modułów	10

# 1. Opis problemu

---

Systemy i aplikacje w dzisiejszych czasach stają się coraz bardziej rozbudowane. Ilość i rozmiar danych, które służą nam za dane wejściowe jak i wyjściowe nawet w najprostszych pracach coraz bardziej wzrasta. Każdemu z nas przytrafiło się pewnie nie raz zgubić w tym gąszczu plików, które trzymamy na swoich komputerach osobistych czy chmurze sieciowej i stracić kilka lub kilkanaście minut na znalezienie interesujących nas w nich informacji. Często zdarza się też sytuacja, w której nie wiemy jakiego rodzaju informacja jest zawarta w wybranym pliku.

Wyobraźmy sobie taką sytuację: użytkownik serfuje po internecie w poszukiwaniu potrzebnych mu informacji. Po znalezieniu plików, które mogą zawierać taką informację, użytkownik ściąga takie pliki na komputer. Ponieważ odpowiednie segregowanie informacji jest sprawą dość nietrywialną, takie pliki są często wrzucane do jednego katalogu, który ma nazwę "Pobrane". Po kilkugodzinnej sesji poszukiwań, taki katalog może zawierać dziesiątki albo i setki plików. Jest oczywistym fakt, że odnaleźć potrzebną informację w takim katalogu jest nadzwyczaj ciężko.

Sytuację tę pogarszają też często niezrozumiałe i nic nie mówiące nazwy plików. Szczególnie dotyczy to artykułów naukowych, których nazwa składa się z numeru identyfikacyjnego. Wyszukiwanie po takich plikach staje się dość problematyczne.

Dobrym przykładem wyżej wymienionej sytuacji jest praca fotografa, który robi dużą ilość zdjęć, gdzie nazwy tych plików noszących zdjęcia często są numerami nadanymi automatycznie przez aparat.

Nierzadko występuje problem, kiedy ważne dla użytkownika pliki są porzucane po różnych katalogach. Kiedy pojawia się problem, żeby znaleźć potrzebny plik, użytkownik nie pamięta, do jakiego katalogu taki plik został zapisany i wtedy musi przeszukiwać wiele katalogów z różnym skutkiem.

Użytkownik w dzisiejszych czasach często pracuje na różnych systemach operacyjnych z różnymi systemami plików, które wnoszą swoje ograniczenie na wyszukiwanie danych.

### **1.1. Lista potrzeb**

- Możliwość tagowania wybranych plików np. za pomocą komentarzy, słów kluczowych.
- Wyszukiwanie plików według dodanych komentarzy, słów kluczowych wśród listy katalogów.
- Wyszukiwanie po zawartości w plikach tekstowych.
- Grupowanie plików w zbiory widoczne tylko na poziomie aplikacji.
- Obsługa różnych systemów plików.

### **1.2. Lista możliwości**

- Tagowanie plików można zapewnić za pomocą stworzenia i zapamiętywania listy plików, do których użytkownik chce dodać pola - komentarz lub słowa kluczowe.
- Wyszukiwanie po wybranych kryteriach można zapewnić wykorzystując gotowe rozwiązania (biblioteki języków programowania), które należy uruchomić dla wybranych pól.
- Wyszukiwanie po zawartości w plikach tekstowych można zrealizować traktując tekst jako dodatkowe pole dla wyszukiwarki.

- Obsługę różnych systemów plików można zapewnić za pomocą wykorzystania bibliotek, które są kompatybilne z różnymi systemami plików.
- Możliwość grupowania plików w zbiory, umożliwiające łatwe wyszukanie plików o tych samych zastosowaniach.

### 1.3. Podsumowanie

Porównując powyższe zestawienie, wnioskujemy, że spora część przytoczonych potrzeb użytkownika nie jest obecnie spełniona. Dotyczy to przede wszystkim następujących przeszkód :

- niemożliwość podzielenia plików w grupy według kryteriów użytkownika, czyli dodania tak zwanych wielu tagów do różnych plików
- niemożliwość dodania do plików dodatkowych informacji opisujących plik poza nazwą, tak zwanych komentarzy
- niemożliwość wyszukiwania po zawartości wielu plików jednocześnie

Potrzeby dzisiejszego użytkownika komputera w zakresie zarządzania danymi są duże. Niejednokrotnie użytkownik ma problem ze znalezieniem tego konkretnego pliku o który mu chodzi, co spowodowane jest ograniczoną ilością sposobów wyszukiwania. Powszechnie dostępne możliwości wyszukiwania plików mimo iż działają przeważnie dokładnie i niezawodnie, to nie oferują zbyt rozbudowanej możliwości zarządzania nimi. Dzięki naszemu programowi użytkownik dostaje narzędzie które dysponuje wszystkimi tymi funkcjami, jest też dodatkowo przenośne na inne platformy i posiada przyjazny interfejs graficzny.

## 2. Wizja rozwiązania

---

Aplikacja będzie stworzona w jednej wersji - wersji stand alone na komputery osobiste. Będzie zapewniona przenośność. Aplikacja będzie miała możliwość uruchomienia na systemie Windows i Linux.

Użytkownik uruchamiając program będzie miał możliwość dodania jak i usunięcia plików lub katalogów do naszego programu.

Dodane pliki i katalogi będą wyświetlane w hierarchii plików, gdzie użytkownik będzie mógł znaleźć plik przechodząc po coraz głębiej zagnieżdżonych katalogach. Użytkownik będzie miał możliwość odnaleźć plik za pomocą inteligentnego wyszukiwania wpisując jego nazwę albo jej część.

Będzie również możliwość dodania komentarza i słów kluczowych przez użytkownika, słowa kluczowe lub komentarz będą istniały tylko i wyłącznie wewnątrz naszego programu.

Pliki, które zostały skomentowane lub tagowane przez użytkownika z użyciem naszego programu będą mogły być wyszukane dodatkowo przy użyciu tych nowo dodanych cech.

Rozszerzeniem podstawowej wersji będzie możliwość instalacji naszego oprogramowania na systemach mobilnych typu Android/iOS.

Po uruchomieniu program poinformuje użytkownika o plikach które zostały przeniesione lub usunięte od ostatniej sesji. W przypadku zmiany ścieżki dostępu do pliku poza programem system poinformuje o tym użytkownika i zaktualizuje dane u siebie. Podczas zapisywania nowego pliku system dodatkowo będzie informował o pojawieniu się możliwych duplikatów plików już istniejących, co będzie służyło unikaniu nadmiernej

ilości tych samych plików oraz zapychaniu dysku tymi samymi danymi. Docelowo funkcjonalność ta będzie także przeszukiwała w tym celu także inne katalogi.

System będzie zbierał informacje o najczęściej wyszukiwanych plikach i rozszerzeniach i na tej podstawie podczas kolejnych poszukiwań sugerował użytkownikowi nazwy plików. Ta sama funkcjonalność opcjonalnie będzie dotyczyła tagowania.

System w zamiarze ma oferować również ciekawy graficzny interfejs pokazujący dane statyczne przydatne użytkownikowi. Jeżeli będzie taka możliwość, to planowana jest też opcja integracji tych danych pomiędzy urządzeniami użytkownika, tzn. miałby on możliwość podglądu, np. jakie pliki znajdują się na jego komputerze ze swojego telefonu.

Główną zaletą w porównaniu do innych podobnych rozwiązań tego typu ma być wyciągnięcie najważniejszych funkcjonalności i wyłożenie ich przed użytkownikiem w postaci ikon i skrótów, co znacznie umili i ułatwi pracę z naszym programem.

Przydatną funkcjonalnością niewątpliwie będzie możliwość skanowania wybranych folderów i automatyczne dodawanie plików w nich zawartych do systemu.

## 3. Koncepcja rozwiązania

---

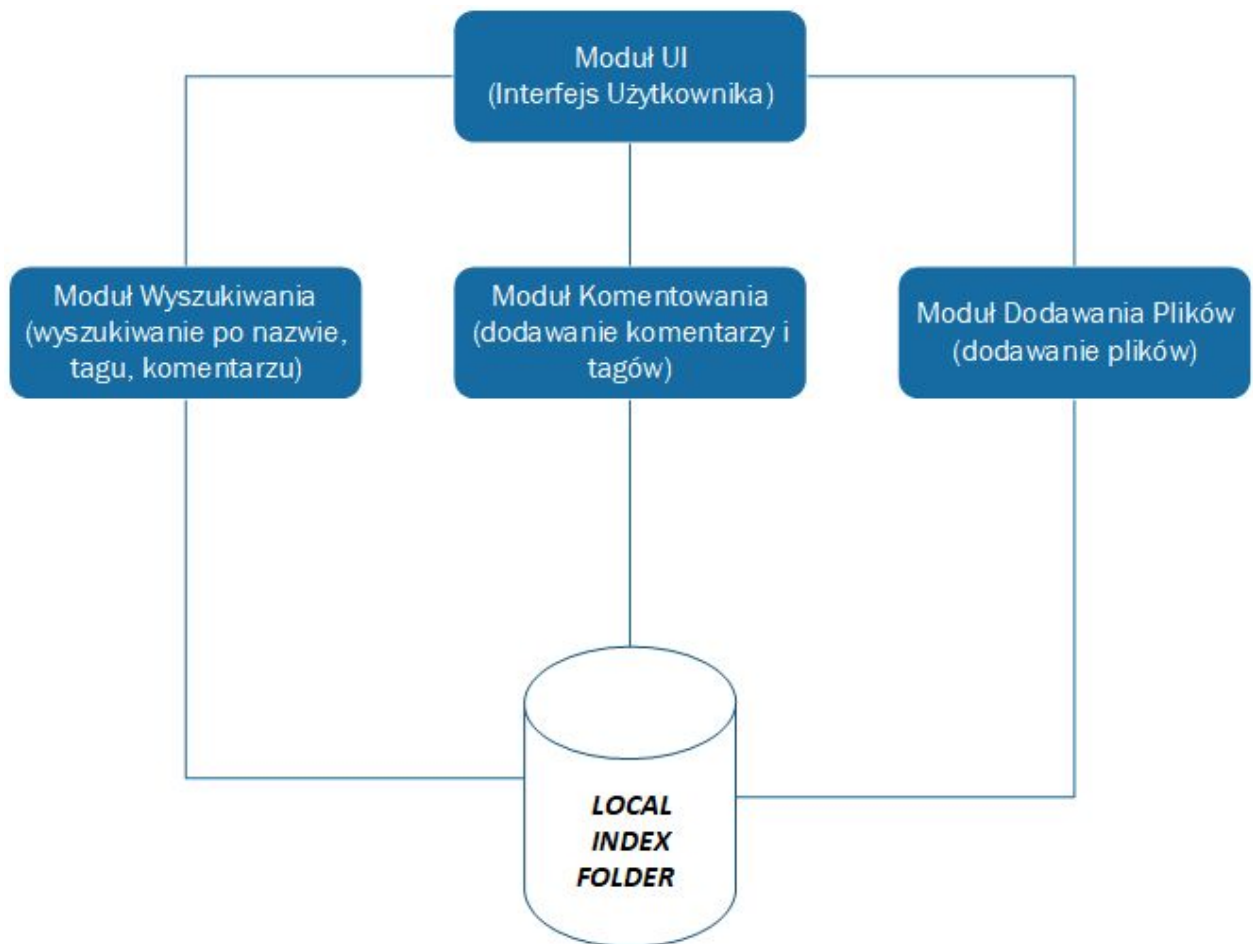
### 3.1. Wprowadzenie

Aplikacja będzie stand alone aplikacją desktopową. Użytkownik będzie mógł wyszukać wpisując do paska, po nazwie, tagu i komentarzu. Wyniki pojawią się od w kolejności od najbardziej trafnego. Użytkownik będzie mógł dodać do programu *Forganizer* plik, aby on został zaindeksowany w celu wyszukiwania. Pliki znajdujące się w programie *Forganizer* będą mogły mieć dodany komentarz.

### 3.2. Architektura systemu

*System składa się z:*

- Modułu interfejsu użytkownika który będzie przyjmował od użytkownika, zapytania i kryterium do wyszukiwania oraz pliki do dodania i komentarze.
- Modułu wyszukiwania który będzie wyszukiwał indeksowane pliki według zapytania i kryterium od modułu UI.
- Moduł komentowania który będzie dodawał do plików komentarz, otrzymany przez moduł UI.
- Moduł dodawania plików, który będzie dodawał nazwę pliku i ścieżkę do niego otrzymane od modułu UI do lokalnego folderu w którym będzie indeksował je.



**Rys. 1. Schemat architektury systemu.**



### 3.3. Moduły:

a) po stronie interfejsu użytkownika:

- Moduł UI

b) po stronie modułu wyszukiwania:

- Moduł wyszukiwania po nazwie
- Moduł wyszukiwania po tagu
- Moduł wyszukiwania po komentarzu

c) dodawanie komentarzy:

- Moduł dodawania komentarzy do pliku

d) dodawanie pliku

- Moduł dodawania pliku
- Moduł indeksowania pliku

### 3.4. Zakres odpowiedzialności modułów:

a) Moduł UI:

- dostarczenie interfejsu użytkownikowi
- pobieranie danych od użytkownika
- zwracanie wyników

b) Moduł wyszukiwania po nazwie:

- wyszukanie zindeksowanego pliku po nazwie

c) Moduł wyszukiwania po tagu:

- wyszukanie zindeksowanego pliku po tagu pod jakim się on znajduje

d) Moduł wyszukiwania po komentarzu:

- wyszukanie zindeksowanego pliku po komentarzu dodanym do niego

e) Moduł dodawania komentarzy do pliku:

- dodanie do pliku komentarzu

f) Moduł dodawania pliku:

- dodanie do folderu z indeksami tj. jego nazwy i ścieżki do niego.

g) Moduł indeksowania pliku:

- indeksowanie plików w celu ich późniejszego wyszukiwania

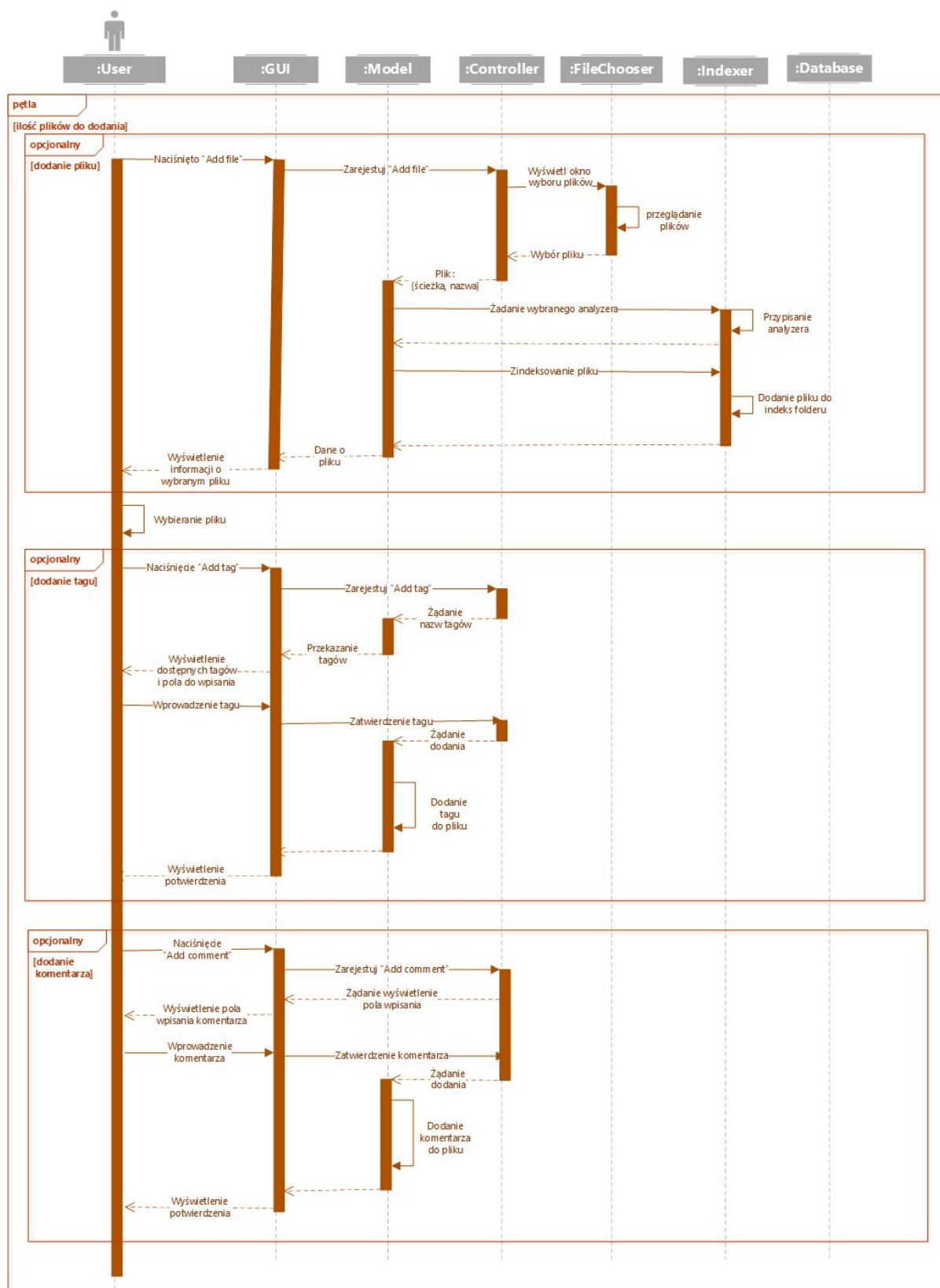
### 3.5. Współpraca modułów

Przepływ informacji między modułami odbywa się przy pomocy struktur języka programowania Java. Dane o plikach znajdują się w folderze lokalnym łącznie z wszystkimi indeksami.

Użytkownik wprowadza do Modułu UI (Interfejs Użytkownika) informacje o tym, w jaki sposób chciałby wyszukać plik oraz kryterium wyszukiwania. Moduł UI jest również odpowiedzialny za wyświetlanie danych i kryterium. Następnie jest tworzony obiekt, który jest przesyłany do modułu wyszukiwania, który korzystając z folderu z indeksami wyszukuje pliki spełniające kryterium wyszukiwania. Moduł wyszukiwania mając już listę plików przekazuje je do Modułu GUI, który je otrzymuje i wyświetla dla użytkownika.

Użytkownik wprowadza do Modułu UI (Interfejs Użytkownika) informacje o tym, w jaki sposób chciałby nadać plikowi jakąś własność (komentarz bądź tag) i wprowadza potrzebne do tego informacje. Moduł UI również wyświetla wprowadzone dane. Następnie jest tworzony obiekt, który jest dostarczany modułowi komentowania, który zrobi odpowiednią zmianę w folderze lokalnym w postaci komentarzu/tagu.

Użytkownik wprowadza do Modułu UI (Interfejs Użytkownika) informacje o pliku, który chciałby dodać do programu, wraz z niezbędnymi danymi, następnie tworzy obiekt, który będzie przekazany modułowi dodawania plików. Moduł UI również wyświetla wprowadzone dane. Moduł dodawania plików dodaje do lokalnego indeks folderu nowy plik.



Rys. 1. Diagram sekwencji.