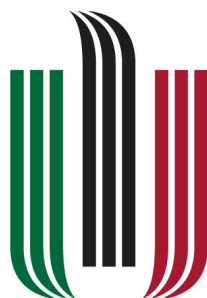


Dokument specyfikacyjny



AGH

**AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA
W KRAKOWIE**

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

Spis treści

	2
1. Lista funkcjonalności programu Forganizer	3
2. Historyjki użycia	4
3. Przypadki użycia	5
3.1. Przypadek użycia: Wyszukiwanie pliku po nazwie	5
Scenariusz podstawowy:	5
3.2. Przypadek użycia: Dodawanie komentarzy	5
Scenariusz podstawowy:	6
Scenariusz alternatywny:	6
3.3. Przypadek użycia: Wyszukiwanie po komentarzu	6
Scenariusz podstawowy:	7
3.4. Przypadek użycia: Dodawanie tagu do pliku	7
Scenariusz podstawowy:	7
Scenariusz alternatywny:	8
3.5. Przypadek użycia: Wyszukiwanie plików po tagu	8
Scenariusz podstawowy:	8
3.6. Przypadek użycia: Dodawanie pliku do systemu	9
Scenariusz podstawowy:	9
Scenariusz alternatywny:	9
3.7. Przypadek użycia: Wyszukiwanie pliku po zawartości	9
Scenariusz podstawowy:	10
4. Architektura logiczna	10
5. Opis działania modułów	11
6. Interakcja użytkownika z programem	11

1. Lista funkcjonalności programu *Forganizer*

Poniższy punkt służy wstępnemu określeniu funkcji spełnianych przez program *Forganizer*.

- Wyszukanie pliku po nazwie
- Użytkownik wpisuje w pasek wyszukiwania nazwę, lub jej część, i pojawiają mu się pliki po nazwie.
- Dodawanie komentarzy
- Użytkownikowi po kliknięciu w przycisk “dodaj komentarz” pojawi się pole do wpisania tekstu, gdzie użytkownik będzie mógł wpisać komentarz a następnie go zapisać.
- Wyszukiwanie po tagu
- Użytkownik wpisuje w pasek wyszukiwania tag i pojawiają mu się pliki po tagu
- Wyszukiwanie po komentarzu
- Użytkownik wpisuje w pasek wyszukiwania komentarz, lub jego część, i pojawiają mu się pliki po komentarzu.

2. Historyjki użytkowników

- Jako standardowy użytkownik chciałbym móc komentować pliki, aby w przyszłości pamiętać jakie zastosowanie miał dany plik.
- Jako standardowy użytkownik chciałbym wyszukiwać pliki po tagu, aby znajdować podobne pliki w błyskawiczny sposób.
- Jako standardowy użytkownik chciałbym móc grupować pliki w zbiory bez zmiany ich lokacji, aby nie popsuć działania innych aplikacji i mieć łatwy dostęp do podobnych plików.
- Jako standardowy użytkownik chciałbym dodać pliki do programu aby mogły być zindeksowane w naszym programie.
- Jako standardowy użytkownik wyszukiwać po zawartości plików tekstowych aby znaleźć to czego szukam bez przeglądania całych plików.
- Jako standardowy użytkownik chciałbym mieć łatwo oznaczać pliki robocze bez zmiany bez zmiany, aby ułatwić sobie pracę z wieloma klientami.
- Jako standardowy użytkownik chciałbym wyszukiwać plik po nazwie, aby zwiększyć produktywność.

3. Przypadki użycia

3.1. Przypadek użycia: Wyszukiwanie pliku po nazwie

Aktor główny: Użytkownik

Streszczenie: Użytkownik wpisuje w pasek wyszukiwania nazwę, lub jej część, i pojawiają mu się pliki po nazwie.

Warunek początkowy: Użytkownik chce wyszukać plik pamiętając nazwę pliku lub tylko jego część.

Scenariusz podstawowy:

1. Użytkownik wpisuje w pole wyszukiwania dostępne dla systemu słowo będące nazwą pliku lub jego częścią.
 2. *Lucene* na podstawie tej nazwy wyszukuje zindeksowane wcześniej nazwy plików i przekazuje je do systemu.
 3. System prezentuje użytkownikowi otrzymane wyniki.
-

3.2. Przypadek użycia: Dodawanie komentarzy

Aktor główny: Użytkownik

Streszczenie: Użytkownik chce dodać komentarz do konkretnego pliku.

Warunek początkowy: Chęć dodania dodatkowej informacji do pliku przez użytkownika celem późniejszego łatwego wyszukiwania.

Scenariusz podstawowy:

1. Użytkownik wybiera plik i klika przycisk “dodaj komentarz”.
2. System wyświetla użytkownikowi pole do wpisania tekstu (komentarza).
3. Użytkownik wprowadza komentarz zatwierdzając go odpowiednim przyciskiem.
4. System zapisuje komentarz dla danego pliku.

Scenariusz alternatywny:

1. Użytkownik wybiera plik i klika przycisk “dodaj komentarz”.
 2. System wyświetla użytkownikowi pole do wpisania tekstu (komentarza).
 3. Użytkownik odrzuca wprowadzony przez siebie komentarz odpowiednim przyciskiem.
 4. System nie zapisuje komentarza, anulując tę operację.
-

3.3. Przypadek użycia: Wyszukiwanie po komentarzu

Aktor główny: Użytkownik

Streszczenie: Użytkownik chce wyszukać plik pamiętając komentarz lub jego część, jaką dodał do tego pliku.

Warunek początkowy: Użytkownik wybiera opcję wyszukiwania po komentarzu.

Scenariusz podstawowy:

1. Użytkownik wpisuje w pole wyszukiwania dostępne dla systemu słowo będące komentarzem lub jego częścią dla plików, które je zawierają.
 2. System wyszuka, posegreguje według trafności i wyświetli użytkownikowi pliki na podstawie wprowadzonych informacji.
-

3.4. Przypadek użycia: Dodawanie tagu do pliku

Aktor główny: Użytkownik

Streszczenie: Użytkownik chce dodać tag do konkretnego pliku.

Warunek początkowy: Chęć dodania dodatkowej informacji do pliku przez użytkownika celem pogrupowania ich w zbiory o kryteriach zadanych przez użytkownika, służące łatwiejszemu ich wyszukiwaniu.

Scenariusz podstawowy:

1. Użytkownik klika dla pliku w przycisk “dodaj tag”.
2. System wyświetla użytkownikowi pole do wpisania tekstu (tag).
3. System na bieżąco podrzuca użytkownikowi propozycje nazw tagów już istniejących w systemie.
4. Użytkownik wybiera z wcześniejszych tagów lub wprowadza nowy tag.
5. Użytkownik zatwierdza swój wybór odpowiednim przyciskiem.
6. System dopisuje dany tag do pliku.

Scenariusz alternatywny:

1. Użytkownik klika dla pliku w przycisk “dodaj tag”.
 2. System wyświetla użytkownikowi pole do wpisania tekstu (tag).
 3. System na bieżąco podrzuca użytkownikowi propozycje nazw tagów już istniejących w systemie.
 4. Użytkownik wybiera z wcześniejszych tagów lub wprowadza nowy tag.
 5. Użytkownik odrzuca swój wybór odpowiednim przyciskiem.
 6. System nie zapisuje zmian powiązań dla pliku.
-

3.5. Przypadek użycia: Wyszukiwanie plików po tagu

Aktor główny: Użytkownik

Streszczenie: Użytkownik chce wyszukać pliki otagowane konkretną etykietą.

Warunek początkowy: Użytkownik wybiera opcję wyszukiwania po tagu.

Scenariusz podstawowy:

1. Użytkownik wpisuje w pole wyszukiwania dostępne dla systemu słowo będące tagiem dla wyszukiwanych plików.
 2. System wyszuka, posegreguje według trafności i wyświetli użytkownikowi pliki na podstawie wprowadzonych tagów.
-

3.6. Przypadek użycia: Dodawanie pliku do systemu

Aktor główny: Użytkownik Streszczenie: Użytkownik chce dodać plik do systemu, celem zaindeksowania go. Warunek początkowy: Plik nie jest jeszcze w systemie i użytkownik wybrał opcję dodania go.

Scenariusz podstawowy:

1. Użytkownik klika przycisk "dodaj plik".
2. Użytkownik wybiera plik spośród listy plików, nie będących w systemie, lub podając jego lokalizację.
3. System sprawdza, czy ma już ten plik zindeksowany w systemie.
4. System dodaje plik do systemu, informując o tym użytkownika.

Scenariusz alternatywny:

1. Użytkownik klika przycisk "dodaj plik".
 2. Użytkownik wybiera plik spośród listy plików, nie będących w systemie, lub podając jego lokalizację.
 3. System sprawdza, czy ma już ten plik zindeksowany w systemie.
 4. System nie dodaje plik do systemu, informując użytkownika o tym, że ma już zindeksowany taki plik.
-

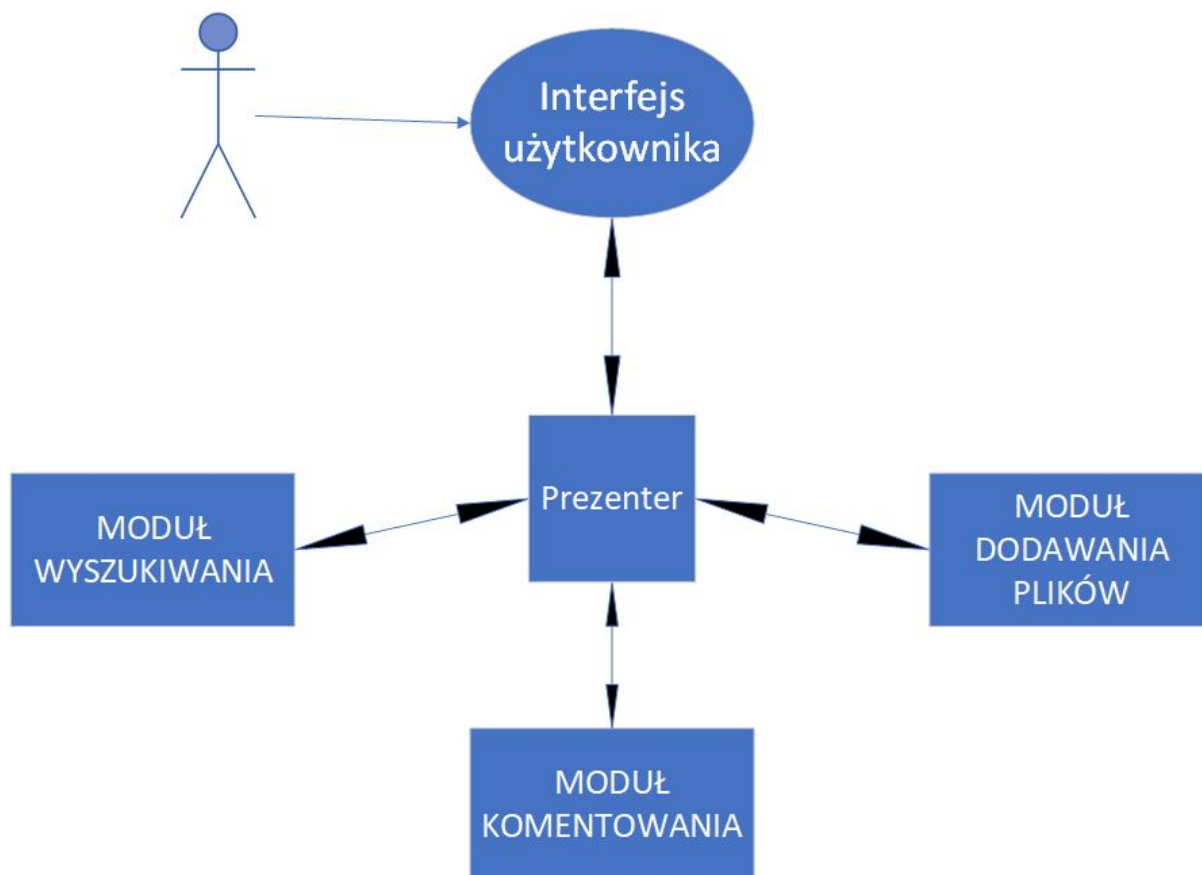
3.7. Przypadek użycia: Wyszukiwanie pliku po zawartości

Aktor główny: Użytkownik Streszczenie: Użytkownik wpisuje w pasek wyszukiwania jakąś frazę z pliku. Warunek początkowy: Użytkownik chce wyszukać plik pamiętając fragment zawartości z tego pliku oraz wybierze opcję wyszukiwania po zawartości.

Scenariusz podstawowy:

1. Użytkownik wpisuje w pole wyszukiwania dostępne dla systemu frazę występującą w zawartości szukanego pliku.
2. *Lucene* na podstawie tej frazy wyszukuje zindeksowane wcześniej zawartości plików i zwraca dopasowania przekazując je do systemu.
3. System prezentuje użytkownikowi otrzymane wyniki.

4. Architektura logiczna



5. Opis działania modułów

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

6. Interakcja użytkownika z programem

Użytkownik będzie korzystał z programu tylko i wyłącznie za pośrednictwem wcześniej wspomnianego interfejsu graficznego, który on już komunikując się z odpowiednimi modułami poprzez pośrednika zwanego kontrolerem będzie wykonywał docelowe polecenia użytkownika. Dzięki takiemu zastosowaniu przed użytkownikiem zostanie ukryta cała zawartość programu, a postawiony zostanie przed nim przyjazny i co najważniejsze intuicyjny interfejs graficzny.

7. Opis biblioteki *Lucene*TM

Podstawową funkcjonalność programu - czyli indeksowanie i wyszukiwanie plików według wybranych kryteriów zabezpiecza biblioteka *Apache Lucene Core*TM (w dokumentacji wspomniana jako *Lucene*). *Lucene* - to wysokowydajna, w pełni funkcjonalna biblioteka dla wyszukiwarek, napisana w całości w języku Java.

Pełną dokumentację biblioteki można zobaczyć [tutaj](#):

Lucene operuje na tak zwanych dokumentach. Dokumentem jest obiekt, który ma pola tekstowe. Każde pole ma swoją nazwę oraz zawartość. W naszym programie dokument jest reprezentacją plików.

W Lucene wyróżniamy dwie podstawowe operacje: indeksowanie dokumentów i ich wyszukiwanie.

Indeksowaniem jest proces dodania pliku do tak zwanego indeksu. Indeksem jest struktura danych, która przechowuje dokumenty w specjalny sposób, umożliwiając ich szybkie wyszukiwanie. W Lucene indeks zawiera macierz tf-idf, w której są przechowywane wektory, reprezentujące dokumenty. Więcej o macierzy tf-idf można przeczytać [tutaj](#):

Proces wyszukiwania zaczyna się od tworzenia dokumentu z zapytania. Taki dokument jest porównywany z innymi dokumentami w indeksie przez obliczanie kąta odległości między nim a każdym dokumentem w indeksie.