



**Multi-projekt z przedmiotów
Inżynieria oprogramowania,
Współczesne bazy danych
i Programowanie w języku Java™**

Spis treści

1 Terminarz.....	3
2 Specyfikacja wymagań - założenia projektowe.....	4
2.1 Informacje ogólne - wprowadzenie.....	4
2.2 Określenie wymagań.....	4
2.2.1 Zagadnienia projektowe.....	4
2.2.2 Główne właściwości i zadania systemu.....	4
2.2.3 Role użytkowników.....	5
3 Analiza.....	7
3.1 Sposób pobierania danych z sieci.....	7
3.2 Omówienie zastosowanej techniki.....	7
3.3 Omówienie bazy danych.....	8
3.3.1 Rodzaj bazy danych.....	8
3.3.2 Rodzaje przechowywanych danych.....	8
3.4 Omówienie wyglądu aplikacji.....	8
3.5 Przypadki użycia.....	8
3.5.1 Wyszukiwanie podstawowe.....	8
3.5.2 Wyszukiwanie spersonalizowane.....	8
3.6 Inne kwestie.....	9
4 Projektowanie.....	10
4.1 Schemat bazy danych.....	10
4.2 Schemat aplikacji.....	10
4.2.1 Omówienie.....	10
4.2.2 Sposób działania wyszukiwarki połączonej.....	10
4.3 Problemy.....	11
5 Implementacja.....	12
6 Testowanie.....	13
7 Źródła.....	14

1 Terminarz

- Do 5 listopada - analiza wymagań oraz wstępne rozwiązania
- Do 3 grudnia - wstępna aplikacja wraz z podstawową bazą danych + diagram klas
- Do końca grudnia - ukończona aplikacja
- 1 tydzień stycznia - testowanie systemu

2 Specyfikacja wymagań - założenia projektowe

2.1 Informacje ogólne - wprowadzenie

Aplikacja typu desktop lub na stronie WWW umożliwiająca wyszukiwanie treści w internecie (w czasie rzeczywistym oraz przy użyciu własnej bazy danych zawierających ranking stron) przy użyciu ogólnodostępnych API wyszukiwarek z 3 wiodących na rynku firm.

Strona projektu: http://quidnam.net/Multi_wyszukiwarka,414.html

2.2 Określenie wymagań

2.2.1 Zagadnienia projektowe

a) Czy firmy Google™, Yahoo!® oraz Microsoft® udostępniają API wyszukiwarek?

- jeżeli tak, to w jaki sposób można pobierać dane, jaka jest licencja
- jeżeli nie, to czy da się wyłuskać dane ze stron internetowych

b) Jaką zastosować technikę?

- aplikacja desktop czy na serwerze
- jaki język programowania
- jaka baza danych i co w niej przechowywać

c) Jak ma wyglądać sama aplikacja?

- GUI
- możliwość wyszukiwania w jednej wyszukiwarce, wyszukiwanie równoległe i połączone (wyniki z 3 wyszukiwarek połączone w jeden rezultat)
- rodzaje wyszukiwania - web, grafika, inne
- ustawienia - ilość wyników na stronę, wersja językowa, [co jeszcze?]
- możliwości konfiguracji administracyjnej [co w tym?]

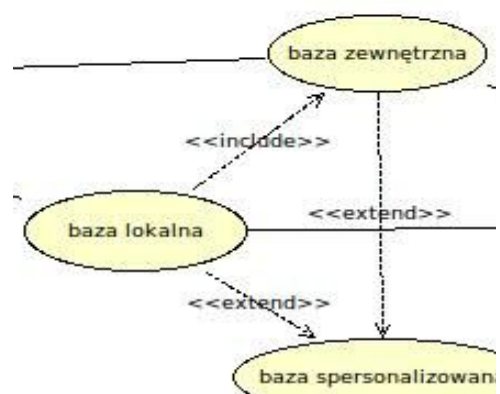
d) Co jeszcze?

TODO

2.2.2 Główne właściwości i zadania systemu

a) Aplikacja ma umożliwić użytkownikom szerokie wyszukiwanie informacji w internecie uwzględniając następujące kryteria podstawowe:

- ujednoczenie wyników z różnych wyszukiwarek (Google™, Yahoo!®, Microsoft® Bing®) - baza



Ilustracja 1: Metody wyszukiwania

zewnętrzna

- możliwość przeszukiwania lokalnych baz danych, zawierających przede wszystkim lokalny ranking już wcześniej ujednoczonych wyników – baza lokalna
- wyszukiwanie spersonalizowane – wg dodatkowych zadanych kryteriów – baza spersonalizowana
- możliwość podłączenia bazy z konta internauty za pomocą strumieni OpenID oferowanych przez ww. firmy

b) Gromadzenie danych pomocniczych

- logi dotyczące wyszukiwania
- statystyki wyszukiwania
- sesje użytkowników

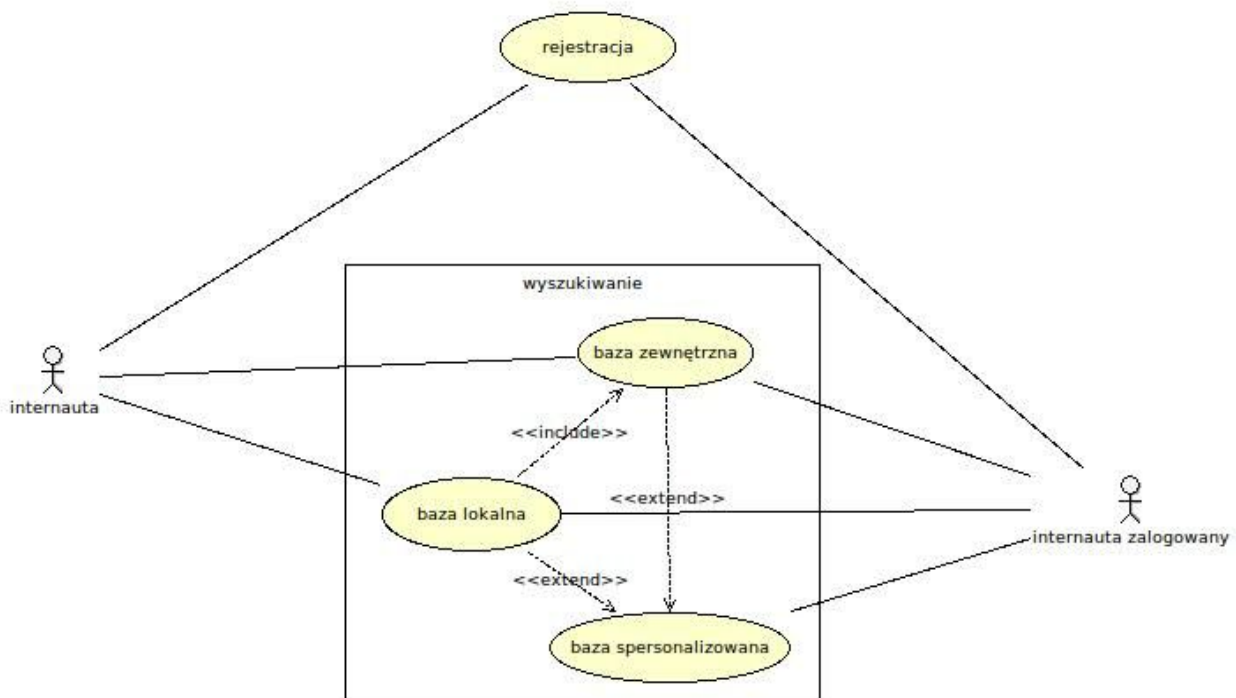
2.2.3 Role użytkowników

a) Internauta

- wyszukiwanie informacji
- ustawienia filtrów i sposobów wyszukiwania

b) Internauta zalogowany

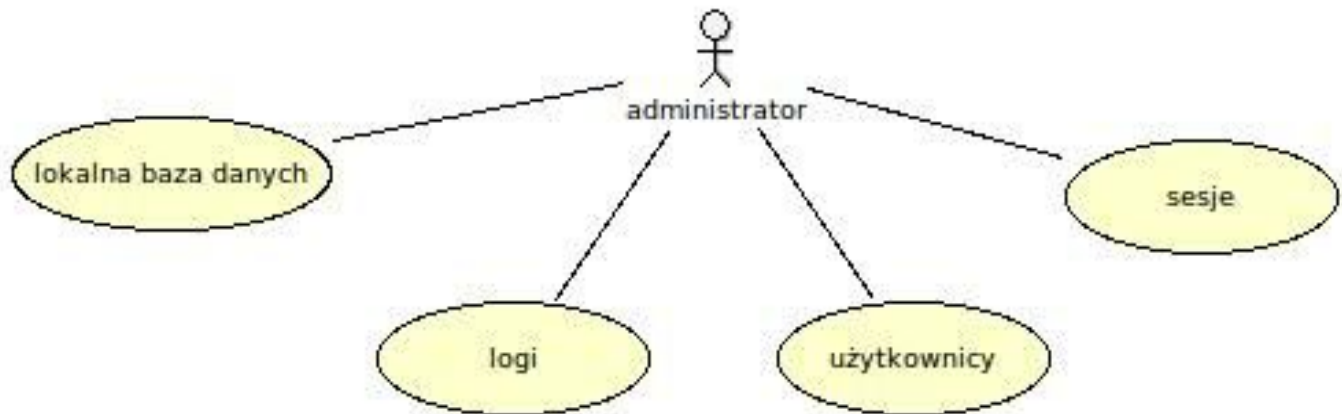
- j.w.
- możliwość podłączenia bazy danych z zewnętrznego serwera
- ustawienia spersonalizowane na podstawie wcześniejszych wyszukiwań i innych preferencji



Ilustracja 2: Role internautów – porównanie

c) Administrator

- zarządzanie logami
- zarządzanie lokalną bazą danych (czyszczenie, tworzenie i aktualizacja wpisów)
- zarządzanie użytkownikami
- zarządzanie sesjami



Ilustracja 3: Role administratora

3 Analiza

3.1 Sposób pobierania danych z sieci

Wszystkie 3 firmy umożliwiają pobieranie wyników wyszukiwania za pomocą XML'a. Szczegółowe informacje opisujące sposób działania API na poniższych stronach.

a) Google™

API

<http://code.google.com/apis/ajaxsearch/>

http://code.google.com/apis/searchappliance/documentation/60/xml_reference.html

przykład

<http://www.google.com/search?output=xml&q=Krak%C3%B3w>

b) Yahoo!®

API

<http://developer.yahoo.com/search/>

<http://developer.yahoo.com/search/web/V1/webSearch.html>

<http://developer.yahoo.com/search/rest.html>

http://developer.yahoo.com/search/boss/boss_guide/index.html

przykład

<http://search.yahooapis.com/WebSearchService/V1/webSearch?appid=YahooDemo&query=Poland&view=keyterms>

c) Microsoft Bing®

API

<http://msdn.microsoft.com/en-us/library/dd250852.aspx>

przykład

<http://api.search.live.net/xml.aspx?Appid=APPID&query=sushi&sources=web>

<http://api.search.live.net/rss.aspx?source=web&query=Krak%C3%B3w>

d) inne

<http://www.antezeta.com/blog/ask-web-search-api>

3.2 Omówienie zastosowanej techniki

Aplikacja JAVA EE działająca na serwerze np. Tomcat 6

- zalety

- nie trzeba instalować aplikacji na komputerze klienta – wystarczy zwykła przeglądarka internetowa
- brak bezpośredniego kontaktu użytkownika z aplikacją
- brak bezpośredniego kontaktu użytkownika z bazą danych
- wady
 - wymaga stałego połączenia z serwerem – brak możliwości wyszukiwania w bazie lokalnego komputera

3.3 Omówienie bazy danych

3.3.1 Rodzaj bazy danych

Baza danych PostgreSQL

- zalety
- wady

3.3.2 Rodzaje przechowywanych danych

- logi aplikacji
- dane „offline” do szybkiego wyszukiwania
- statystyki
- sesje użytkowników

3.4 Omówienie wyglądu aplikacji

Ponieważ zastosowana technika to aplikacja Java EE, to całość będzie opierać się na bazie HTML'a i CSS'a. Zatem całość może być w prosty sposób konfigurowalna, bez ingerencji w działanie samej aplikacji.

3.5 Przypadki użycia

3.5.1 Wyszukiwanie podstawowe

1. Scenariusz główny
 1. internauta wpisuje adres wyszukiwarki w pasku adresu przeglądarki
 2. następnie wpisuje wyszukiwaną frazę w pole wyszukiwarki i naciska przycisk Szukaj
 3. na stronie wyświetlają się wyniki wyszukiwania
2. Rozszerzenia
 1. internauta zleca wyszukiwanie tylko w wybranym serwisie
 2. internauta zmienia ilość wyświetlanych wyników na stronę
 3. system wyświetla informacje o braku wyników wyszukiwania
 1. internauta jest proszony o wprowadzenie innej frazy w polu wyszukiwania

3.5.2 Wyszukiwanie spersonalizowane

1. Scenariusz główny
 1. internauta wpisuje adres wyszukiwarki w pasku adresu przeglądarki
 2. internauta jest proszony o zalogowanie się do serwisu
 1. w przypadku nieudanej autoryzacji, internauta jest proszony o ponowienie logowania

3. teraz można swobodnie wyszukiwać jw.

3. Rozszerzenia

1. internauta może zlecić dodatkowe personalne wyszukiwanie uwzględniające

- jego poprzednie wyszukiwania

poprzednie kliknięcia w wynikach

3.6 Inne kwestie

Aplikacja w wersji podstawowej będzie obsługiwała tylko wyszukiwanie. Kwestie wyszukiwania spersonalizowanego oraz administrowania serwisem są sprawą drugorzędną, więc ich implementacja w programie nie będzie wykonana w pierwszej odsłonie aplikacji.

4 Projektowanie

4.1 Schemat bazy danych

Baza danych składa się z 2 głównych tabel. Tabela z konfiguracją (apikeys wyszukiwarek, ustawienia wstępne):

```
CREATE TABLE multisearch_config
(
  id integer NOT NULL,
  "name" text,
  "value" text,
  CONSTRAINT idd PRIMARY KEY (id)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE multisearch_config OWNER TO bartek;
```

Tabela przechowująca wyniki wyszukiwania:

```
CREATE TABLE multisearch_results
(
  query text NOT NULL,
  rank integer NOT NULL,
  total integer NOT NULL,
  url text NOT NULL,
  title text NOT NULL,
  CONSTRAINT "key" PRIMARY KEY (query, url, rank, total)
)
WITH (
  OIDS=FALSE
);
ALTER TABLE multisearch_results OWNER TO bartek;
```

4.2 Schemat aplikacji

4.2.1 Omówienie

Aplikacja składa się z formularza widocznego na stronie głównej oraz odnośników do poszczególnych rodzajów wyszukiwania – na ich stronach także jest dostępny formularz wyszukiwania.

Za pomocą formularza, użytkownik może wprowadzić poszukiwane hasło (frazę), wybrać ilość wyników do wyświetlenia na stronie oraz (w głównym formularzu) wybrać rodzaj wyszukiwania.

4.2.2 Sposób działania wyszukiwarki połączonej

Każda wyszukana strona otrzymuje wagę w zależności od pozycji, którą zajmuje w wynikach oraz od ilości wyników. Przykładowo, jeżeli jest na 1 miejscu, a wyników jest 10, to waga będzie 10. Dla strony na pozycji 4, dla 10 wyników waga wynosi 6.

Jeżeli jakaś strona powtórzy się w wynikach innej wyszukiwarki, to do jej wagi się sumują.

4.3 Problemy

Aplikacja nie może pokazywać wyników wyszukiwania z Google™, ponieważ możliwość prezentacji wyników w formacie XML jest dostępna tylko za opłatą (źródło: <http://www.google.com/cse/compare>).

Po dłuższym poszukiwaniu internetu udało się jednak znaleźć kod API umożliwiający korzystanie z XML'owych wyników wyszukiwania w Google™ na stronach firmy Atmel (kod: <http://www.google.com/cse/home?cx=008864967094569334548:eyfn6tp4lcm>).

5 Implementacja

Zgodnie z założeniami, aplikacja została napisana przy użyciu techniki Java™ EE.

Silnikiem aplikacji jest zbiór klas w języku Java™. Wszelkie relacje między nimi zostały opracowane zgodnie z założeniami programowania obiektowego.

Aby aplikacja mogła łączyć się z bazą danych, przetwarzać pliki XML i budować tablice danych na podstawie tych plików niezbędne są odpowiednie biblioteki, także dołączone do całości.

Główne funkcje aplikacji są wywoływane z poziomu plików JSP – zmienne przyjmują wartości zmiennych z klas i funkcji, a następnie są odpowiednio wyświetlane. Zaletą takiego rozwiązania jest niezależność jądra aplikacji od jej interfejsu.

W efekcie użytkownik widzi przystępną stronę internetową.

Wszelkie źródła w dodatkowym załączniku.

6 Testowanie

Możliwy błąd	Reakcja aplikacji
Brak włączonego serwera baz danych	<ul style="list-style-type: none"> W przypadku serwera Tomcat uruchamianego przez Eclipse'a powyższy serwer wyłącza się
Brak API key wyszukiwarki	<ul style="list-style-type: none"> komunikat: <i>Brak wyników wyszukiwania.</i>
Użytkownik nie wpisze frazy wyszukiwania	<ul style="list-style-type: none"> Wyniki nie wyświetlą się - komunikat: <i>Brak wyników wyszukiwania.</i>
Zapytanie zawierające pojedynczy cudzysłów	<ul style="list-style-type: none"> Wyniki wyświetlają się poprawnie W polu wyszukiwanego hasła nie wyświetla się nic W przypadku zapytania połączonego wyniki nie dodają się do bazy danych
Wyników jest mniej niż ustawiono w opcji „Ilość wyników na stronę”	<ul style="list-style-type: none"> Nadmiarowe pozycje nie wyświetlają się.
Wyjątki	<ul style="list-style-type: none"> W przypadku poniższych wyjątków, wyświetla się strona z błędem <code>/multiSearch/ErrorDisplay</code> java.lang.NullPointerException java.io.FileNotFoundException java.sql.SQLException org.postgresql.util.PSQLException java.lang.ClassNotFoundException java.lang.Exception tools.myExeption W przypadku pozostałych wyjątków nic nie jest wyświetlane użytkownikowi - podgląd możliwy tylko w logach serwera Konfiguracja w pliku <code>/multiSearch/WebContent/WEB-INF/web.xml</code>
Błędy HTTP	<ul style="list-style-type: none"> W przypadku poniższych błędów, wyświetla się strona z błędem <code>/multiSearch/ErrorDisplay</code> Kod 400 Kod 404 Kod 500 W przypadku pozostałych błędów nic nie jest wyświetlane użytkownikowi - podgląd możliwy tylko w logach serwera Konfiguracja w pliku <code>/multiSearch/WebContent/WEB-INF/web.xml</code>

7 Źródła

- a) Java™ Tutorial
<http://java.sun.com/docs/books/tutorial/index.html>
http://www.java2s.com/Tutorial/Java/0400__Servlet/0360__Error-Page.htm
- b) Zdjęcie na stronie tytułowej
<http://www.cobbcatt.org/images/OneSearchGlass.jpg>
- c) Zastosowane oprogramowanie UML - Eclipse AmaterasUML
http://amateras.sourceforge.jp/cgi-bin/fswiki_en/wiki.cgi?page=AmaterasUML

Google, Yahoo!, Microsoft Bing i inne nazwy lub grafiki są znakami towarowymi lub zastrzeżonymi znakami towarowymi ich właścicieli.