POLITECHNIKA POZNAŃSKA
WYDZIAŁ ELEKTRYCZNY, INFORMATYKA
SEMESTR VI, GRUPA BSI-2

Podstawy Teleinformatyki
WebScrapper / Metawyszukiwarka

Paweł Soja
Numer indeksu: 122031
pawel.soja@student.put.poznan.pl

Krzysztof Łuczak
Numer indeksu: 122008
krzysztof.t.luczak@student.put.poznan.pl

Dawid Wiktorski
Numer indeksu: 122056
dawid.wiktorski@student.put.poznan.pl

Poznań, 14 kwietnia 2017

Spis treści

1 Opis i uzasadnienie wyboru tematu

Celem projektu jest zaprojektowanie i zbudowanie platformy do zbierania i prezentowania danych z różnych stron internetowych. Platforma składa się z serwisu internetowego prezentującego dane użytkownikom zalogowanym oraz z aplikacji zbierających te dane.

Temat wybraliśmy, ponieważ interesuje nas dziedzina przetwarzania danych. Chcielibyśmy poznać technologie scrapingu, parsowania stron internetowych oraz język Python, framework Django i technologie front-endowe tj. HTML5, Javascript. Jednocześnie nie znaleźliśmy zadowalającego nas serwisu, który udostępniałby takie usługi, dlatego sami zdecydowaliśmy zrobić swój.

2 Podział prac pomiędzy członków zespołu

Tablica 1: Podział prac

Lp.	**Opis**	**Osoby**
1.	Baza danych	Wszyscy
2.	Projekt interfejsu	Paweł Soja
3.	Front-end serwisu	Paweł Soja
4.	Back-end serwisu	Krzysztof Łuczak, Dawid Wiktorski
5.	Moduł I	Paweł Soja
6.	Moduł II	Krzysztof Łuczak
7.	Moduł III	Dawid Wiktorski
8.	Testowanie	Wszyscy

3 Opis funkcjonalności

Aktorzy

- użytkownik
  - użytkownik zalogowany - posiada prawa do użytkowania serwisu,

  - użytkownik niezalogowany - może dokonać rejestracji,
  - administrator - zarządza serwisem,
- aplikacja internetowa - prezentuje dane,
- moduł zbierający dane (scraper) - zbiera i przetwarza dane.

Tablica 2: Funkcjonalności

Funkcja	**Opis**	**Aktorzy**
Przeglądanie strony głównej	Możliwość przeglądania strony głównej serwisu.	Użytkownicy
Rejestracja	Możliwość zarejestrowania konta w serwisie.	Użytkownik niezalogowany
Potwierdzenie rejestracji, zmiany hasła lub zmiany adresu e-mail konta	Możliwość potwierdzenia rejestracji, zmiany hasła lub zmiany adresu e-mail konta poprzez kliknięcie link aktywacyjny wysłany pocztą elektroniczną.	Użytkownik niezalogowany
Logowanie	Możliwość logowania się do serwisu.	Użytkownik niezalogowany
Wylogowanie	Możliwość wylogowania się z serwisu.	Użytkownik zalogowany, administrator
Zmiana hasła do konta	Możliwość zmiany hasła do aktywnego konta.	Użytkownik zalogowany, administrator
Zmiana adresu e-mail konta	Możliwość zmiany adresu e-mail konta.	Użytkownik zalogowany, administrator
Ustawienie profilu źródeł	Możliwość wybrania źródeł, z których otrzymywane będą informacje.	Użytkownik zalogowany, administrator
Ustawienie profilu tagów	Możliwość wybrania tagów, na podstawie których filtrowane będą informacje.	Użytkownik zalogowany, administrator
Ustawienie filtra daty	Możliwość wybrania przedziału czasowego, na podstawie którego filtrowane będą informacje.	Użytkownik zalogowany, administrator

Tablica 2 – *Kontynuacja*

Funkcja	**Opis**	**Aktorzy**
Zbieranie danych ze strony i parsowanie ich	Scraper zbiera dane ze strony, parsuje je oraz zapisuje do bazy danych. Jeden scraper zbiera dane z jednej strony.	Scraper

4 Wybrane technologie i uzasadnienie

- Back-end - Python, Django, Celery
  - stosunkowo krótki czas tworzenia aplikacji przy jednoczesnym zachowaniu pełnej funkcjonalności, stabilności i wydajności
- Front-end - HTML5, Javascript
  - ???
- Moduły scrapujące - Python, biblioteka BeautifulSoup, Scrapy
  - technologie przeznaczone do parsowania stron,
  - duże możliwości,
  - łatwa implementacja architektury modułowej
- Baza danych - SQLite
  - łatwa integracja z językiem Python,
  - w przyszłości prawdopodobnie zostanie zastąpiona inną

5 Architektura rozwiązania

Tablica 3: Opis bazy danych

Tabela	**Opis**
Articles	Zawiera wszystkie sparsowane strony. (teraz kwestia ile je tam trzymać??)

Tablica 3 – *Kontynuacja*

Tabela	**Opis**
Tags	Zawiera wszystkie dostępne tagi. Dodanie nowego taga odbywa się automatycznie, gdy scraper podczas parsowania wykryje, że danego taga jeszcze nie ma w bazie.
ArticleTagMap	Łączy daną stronę z odpowiednim tagiem.
Sources	Zawiera wszystkie dostępne źródła, czyli strony internetowe, z których zbieramy dane. Dodanie odbywa się ręcznie. Administrator musi napisać moduł dla danej strony.
ArticleSourceMap	Łączy daną stronę z odpowiednią stroną z której pochodzi.
Users	Zawiera wszystkich użytkowników serwisu.
TagsProfile	Łączy użytkownika z tagami, które wybrał.
SourceProfile	Łączy użytkownika z źródłami danych, które wybrał.

6 Interesujące problemy i ich rozwiązania

7 Opis stron internetowych, z których zbierane są informacje

sekurak.pl

Tablica 4: Parametry artykułów

Tytuł	Data opublikowania	Tagi	Obrazek	Fragment tekstu	Link

dobreprogramy.pl/Blog.html

Tablica 5: Parametry artykułów

Tytuł	Data opublikowania	Tagi	Autor	Fragment tekstu	Link

niebezpiecznik.pl

Tablica 6: Parametry artykułów

Tytuł	Data opublikowania	Tagi	Autor	Obrazek	Fragment tekstu	Link

zaufanatrzeciastrona.pl

Tablica 7: Parametry artykułów

Tytuł	Data opublikowania	Tagi	Autor	Obrazek	Fragment tekstu	Link

wykop.pl

Tablica 8: Parametry artykułów

Tytuł	Data opublikowania	Tagi	Autor	Obrazek	Fragment tekstu	Link

8 Instrukcja użytkowania aplikacji