

WYŻSZA SZKOŁA TECHNOLOGII
INFORMATYCZNYCH W WARSZAWIE

WYDZIAŁ INFORMATYKI
KIERUNEK INFORMATYKA

Paweł KŁOSIEWICZ

Aplikacja
wspomagająca katalogowanie plików.

PRACA DYPLOMOWA
wykonana pod kierunkiem
prof. dr. hab. inż. A. Barczak

Warszawa 2012

Spis treści

Wykaz ważniejszych skrótów i oznaczeń	1
1. Wstęp	2
1.1. Cel i zakres pracy	2
1.2. Teza pracy	2
2. Specyfikacja wymagań	2
2.1. Wymagania funkcjonalne	2
2.2. Wymagania niefunkcjonalne	3
3. (1)state-of-art - realia obszaru badawczego	3
4. (2)własne propozycje rozwiązań	3
5. (3)eksperymenty z wynikami - czy też przedstawienie analizy	3
6. Wykorzystane technologie	3
6.1. Zastosowane języki i technologie	3
6.1.1. MongoDB	3
6.1.2. JRuby	3
6.2. Wykorzystane narzędzia wspomagające	4
6.2.1. L ^A T _E X	4
6.2.2. Git	4
7. Diagramy	5
7.1. Diagramy klas	5
7.2. Diagramy sekwencji	5
7.3. Diagramy przypadków	5
8. Opis funkcjonalności	5
8.1. Dostępne funkcjonalności dla zgłoszeń..telefon	5
8.2. Dostępne funkcjonalności dla zgłoszeń..sms	5
9. Przykład działania aplikacji	5
10.Zakończenie	5
Literatura	6

Spis tabel	7
Spis schematów	8
Spis fragmentów kodów źródłowych programów komputerowych	9
Załączniki	9
A. Kod źródłowy	10
A.1. JRuby	10
A.2. MongoDB	10
B. else ??	10
Streszczenie	10
Słowa kluczowe	10

Wykaz ważniejszych skrótów i oznaczeń

MVC wzorzec projektowy Model–Widok–Kontroler

1. Wstęp

Zarządzanie plikami składowanymi na dyskach komputerów przy coraz bardziej postępującym procesie przetwarzania tekstu, obrazu, dźwięku na zapis cyfrowym coraz trudniej jest nadążyć wszystko na bieżąco uspecyfikować - ubrać w odpowiednią strukturę a priori, nie zawsze wychodzi. Budujemy coraz bardziej wyrafinowane struktury danych, które coraz trudniej jest porównać a co dopiero odpowiednio przypisać. Nasuwa się też pytanie jak odróżnić pliki które zostały wygenerowane przez użytkownika a które są jedynie kopią danych ściągniętych z internetu i ogólnie dostępnych?

Taka problematyka nastroja problemów przy ocenie ważności dokumentów. Użytkownik powinien posiadać odpowiednie narzędzie ułatwiające pracę z jego plikami, pozwalając w miarę przystępny sposób operować na nich w bieżącej pracy.

1.1. Cel i zakres pracy

Celem pracy jest opracowanie i implementacja aplikacji usprawniającej zarządzanie plikami. Najważniejszymi cele szczegółowe to:

- Pozwolenie użytkownikowi na wyszukiwanie plików po metadanych
-

1.2. Teza pracy

Tutaj teza pracy, jeśli trzeba...

2. Specyfikacja wymagań

Tutaj treść rozdziału. Poniżej kolejne podrozdziały...

2.1. Wymagania funkcjonalne

Aplikacja powinna pozwalać na szybkie znajdowanie dodanych do niej plików bez potrzeby znajomości nazwy pliku a tylko danych określających go.

Aplikacja powinna mieć możliwość automatycznego uporządkowania plików wg. wyznaczonego schematu.

Aplikacja powinna dawać możliwość rozszerzenia o dodatkową funkcjonalność - ściąganie napisów do wybranych filmów czy okładek do znalezionych albumów z muzyką.

2.2. Wymagania niefunkcjonalne

Aplikacja powinna być w stanie znaleźć opisy do filmów aktualnie będących w kinach.

3. (1)state-of-art - realia obszaru badawczego

parę zdań wprowadzających

4. (2)własne propozycje rozwiązań

5. (3)eksperymenty z wynikami - czy też przedstawienie analizy

6. Wykorzystane technologie

6.1. Zastosowane języki i technologie

6.1.1. MongoDB

Do projektu użyłem obiektowej bazy danych MongoDB. W przeciwieństwie do używania związków w nazach relacyjnych nie musimy tworzyć na sztywno związków (nawet nie mamy standardowo takiej opcji), wszystkie operacje kojarzenia ze sobą encji muszą być definiowane w logice programu.

Język zapytań w MongoDB różni się znacznie od tego do którego zostaliśmy przyzwyczajeni w przypadku relacyjnych baz danych. Nie używa on bowiem języka SQL - co daje mu możliwość porzucenia pewnych intensywnie obciążających instrukcji takich jak np. JOIN. Zamiast tego posługujemy się językiem JavaScript, który pierwotnie był wykorzystywany w programowaniu części aplikacji w aplikacjach internetowych.

6.1.2. JRuby

Ruby jest obiektywnym językiem programowania trzeciej generacji. Swoją składnię wzoruje na Perlu, Smalltalku, Eiffel i Lispiewik [6]. Moc języka drzemie w uproszczonej składni, porzucającej niektóre niepotrzebne znaki m.in. średnika na końcach instrukcji czy też nawiasów, przy przekazywaniu parametrów do funkcji gdy podawany jest tylko jeden parametr. Wszystkie wprowadzone zabiegi mają na celu

zwiększenie czytelności kodu.

Od samych początków język jest dostosowany do budowania aplikacji w paradygmacie ¹.

JRuby zaś jest implementacją języka Ruby napisaną pod wirtualną maszynę Javy. Zaletą takiego rozwiązania jest że zamiast możliwości użycia języka Ruby można zawrzeć normalny kod Javy. Pozwala to na użycie wielu bibliotek które mogą nie mieć swojego odpowiednika w świecie referencyjnego interpretera Rubiego.

6.2. Wykorzystane narzędzia wspomagające

6.2.1. \LaTeX

Do składania pracy został użyty język \LaTeX który wyewoluował z języka \TeX . Służy on do definiowania ..

6.2.2. Git

Pisząc jakikolwiek program potrzebujący nakładów większej ilości czasu i/lub ludzi pojawia się potrzeba trzymania wersji wytwarzanego dobra - programu, dokumentacji czy też specyfikacji. Właśnie aby zapobiec zamrożeniu wykonywania projektu ze względu na awarię systemu operacyjnego, pojedynczego pliku, nośnika danych albo nawet przypadkowym nadpisaniem przez programistę kodu jest możliwość przywrócenia kodu do poprzedniego (czyt. działającego) stanu.

Oczywiście przywracanie stanu dają nam różne programy do tworzenia kopii zapasowych, więc co więcej daje nam GIT? Otóż jesteśmy w stanie scalać wyniki pracy wielu osób i rozwiązywać konflikty spowodowane zmiany jednego pliku przez kilka osób - przy czym rozwiązywanie konfliktów jest możliwe do przeprowadzenia często w sposób automatyczny.

¹TDD – (Test Driven Development) proces wytwarzania oprogramowania opierający się na powtarzaniu krótkich cykli - tworzenia testu, zawaleniu go, napisaniu kawałka programu, przetestowaniu czy kod osiągnął oczekiwany rezultat.

7. Diagramy

7.1. Diagramy klas

7.2. Diagramy sekwencji

7.3. Diagramy przypadków

8. Opis funkcjonalności

8.1. Dostępne funkcjonalności dla zgłoszeń..telefon

8.2. Dostępne funkcjonalności dla zgłoszeń..sms

9. Przykład działania aplikacji

10. Zakończenie

- Ocena przydatności
- wskazanie zastosowań
- wskazanie dalszego rozwoju
- przytoczenie celu pracy
- krótkie streszczenie przeprowadzanych analiz czy badań, w kilku zdaniach streszcza poszczególne rozdziały
- konkluzja - podkreślenie osiągnięcie postanowionego we wstępie pracy celu

Literatura

- [1] J. Slepian. *Extinction of an AC arc*. Transactions AIEE Vol. 47 (1928) No. 4, pp. 1398-1407.
- [2] I. N. Bronsztejn and K. A. Siemiendiajew. *Matematyka. Poradnik Encyklopedyczny*. Wydawnictwo Naukowe PWN, Warszawa 1999.
- [3] A. Greenwood. *Electrical Transients in Power Systems*. John Wiley & Sons, INC., 1991.
- [4] R. P. Feynman, R. B. Leighton, and M. Sands. *Feynmana wykłady z fizyki*. PWN, Warszawa, 1974.
- [5] G. M. Fichtenholz. *Rachunek różniczkowy i całkowy*. PWN, Warszawa 1980.
- [6] Ruby (programming language). [online], maj 2012. [dostęp: 2012-05-30 16:52Z].

Spis tablic

Spis rysunków

Załączniki

A. Kod źródłowy

A.1. JRuby

A.2. MongoDB

B. else ??

Oświadczam, że niniejszą pracę napisałem samodzielnie, nie posługując się nie-
dozwolonymi sposobami korzystania z cudzych opracowań. Pozycje wymienione w
Bibliografii wykorzystywałem, mając do nich bezpośredni dostęp. Jeśli korzystałem
ze źródeł, z którymi nie miałem bezpośredniego kontaktu, zaznaczałem to w tekście
pracy. (-) Paweł Kłosiewicz

Oświadczam niniejszym, że przedkładaną pracę inżynierską napi-
sałem samodzielnie. Oznacza to, że przy pisaniu pracy poza niezbędnymi konsulta-
cjami nie korzystałem z pomocy innych osób, a w szczególności nie zlecałem opraco-
wania rozprawy innym osobom. Wszystkie dane, istotne myśli pochodzą z literatury
i opatrzone są odpowiednim przypisem. Jednocześnie Przyjmuje do wiadomości, że
gdyby powyższe oświadczenie okazało się nieprawdziwe, decyzja o wydaniu dyplomu
zostanie cofnięta.

własnoręczny podpis