

**Uniwersytet Warszawski**  
Wydział Matematyki, Informatyki i Mechaniki

**Patryk Hes, Paweł Kamiński, Borys Popławski, Robert  
Tomkowski**

Nr albumu: 320327, 320408, 320372, 321179

# **Zrąb do aplikacji usprawniających pracę biurową**

**Praca licencjacka  
na kierunku INFORMATYKA**

Praca wykonana pod kierunkiem  
**dra Janusza Jabłonowskiego**

Czerwiec 2014

## **Oświadczenie kierującego pracą**

Potwierdzam, że niniejsza praca została przygotowana pod moim kierunkiem i kwalifikuje się do przedstawienia jej w postępowaniu o nadanie tytułu zawodowego.

Data

Podpis kierującego pracą

## **Oświadczenie autora (autorów) pracy**

Świadom odpowiedzialności prawnej oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie i nie zawiera treści uzyskanych w sposób niezgodny z obowiązującymi przepisami.

Oświadczam również, że przedstawiona praca nie była wcześniej przedmiotem procedur związanych z uzyskaniem tytułu zawodowego w wyższej uczelni.

Oświadczam ponadto, że niniejsza wersja pracy jest identyczna z załączoną wersją elektroniczną.

Data

Podpis autora (autorów) pracy

## **Streszczenie**

## **Słowa kluczowe**

zrąb, REST, WWW

## **Dziedzina pracy (kody wg programu Socrates-Erasmus)**

11.3 Informatyka

## **Klasyfikacja tematyczna**

Information systems  
World Wide Web  
Web services  
RESTful web services

## **Tytuł pracy w języku angielskim**

Framework for application improving office work



# Spis treści

Wprowadzenie . . . . .	5
1. Podstawowe pojęcia . . . . .	7
1.1. Definicje . . . . .	7
2. Opis projektu . . . . .	9
3. Rozwiązania konkurencyjne . . . . .	11
3.1. Bitbucket . . . . .	11
4. Zastosowane rozwiązania . . . . .	13
5. Włożona w projekt praca . . . . .	15
6. Osiągnięcia . . . . .	17
7. Wyczka Zasoby . . . . .	19
8. Wykonana praca przez poszczególne osoby w projekcie . . . . .	21
9. Zawartość kodu dostarczonego wraz z pracą . . . . .	23
Bibliografia . . . . .	25



# Wprowadzenie

Blabalizator różnicowy jest podstawowym narzędziem blabalii fetorycznej. Dlatego naukowcy z całego świata prześcigają się w próbach efektywnej implementacji. Opracowana przez prof. Fifaka teoria fetorów  $\sigma$ - $\rho$  otwiera w tej dziedzinie nowe możliwości. Wykorzystujemy je w niniejszej pracy.

Przystępne wprowadzenie do blabalii fetorycznej można znaleźć w pracy Fifaka i Gryzogrzechotalskiego [?]. Dlatego w niniejszym tekście ograniczymy się do przypomnienia pojęć podstawowych.

Praca składa się z pięciu rozdziałów i dodatków. W rozdziale 1 przypomniano podstawowe pojęcia blabalii fetorycznej. Dotychczasowe próby implementacji blablizatora różnicowego zestawiono w rozdziale ?? . Rozdział ?? przedstawia teorię Fifaka i wyjaśnia sposób jej wykorzystania w implementacji blabalizatora. W rozdziale ?? przedstawiono algorytm blabalizy i realizujący go program komputerowy. Ostatni rozdział zawiera przemyślenia dotyczące możliwego wpływu dostępności efektywnej blabalizy numerycznej na rozwój blabalii fetorycznej. W dodatkach umieszczono najciekawszy fragment programu, przykładowe dane i wyniki działania programu.





# Rozdział 1

## Podstawowe pojęcia

Pojęciem pierwotnym blabalii fetorycznej jest *blaba*. Blabaliści nie podają jego definicji, mówiąc za Ciach-Pfe t-ām Kûn (fooistyczny mędrzec, XIX w. p.n.e.):

### 1.1. Definicje

- eenet – Nazwa własna głównej aplikacji.
- framework – Bezpośrednio tłumaczony jako zrąb, jest to szkielet do budowy aplikacji
- wtyczka – Aplikacja korzystająca z udogodnień środowiska zapewnianego przez eenet.
- wtyczki przykładowe – Wtyczki, których wykonanie jest przewidziane w ramach wykonania aplikacji eenet.
- przeglądarka internetowa - Aplikacja instalowana na urządzeniu użytkownika umożliwiająca wyświetlanie stron WWW dostarczanych przez serwer.
- serwer produkcyjny – Komputer serwujący stronę WWW, posiada również w sobie bazę danych aplikacji.
- serwer developerski/testowy – Maszyna, która służy do testowania każdej wersji aplikacji zanim ta trafi na serwer produkcyjny.
- kopia bezpieczeństwa – Informacje, które mają służyć do odtworzenia oryginalnych danych, znajdujących się na serwerze produkcyjnym, w przypadku ich utraty lub uszkodzenia.
- front-end – Część kodu zapewniająca graficzny i testowy interfejs, jest interpretowana przez przeglądarkę, tzn że działa po stronie użytkownika.
- back-end – Część kodu działająca po stronie serwera, odpowiada za logikę operacji na danych.
- aplikacja internetowa – Aplikacja w formie strony WWW, posiada front-end i back-end.
- one page application – Podczas użytkowania aplikacji typu "one page" nigdy nie zostaje przeładowana strona, dane pomiędzy klientem a serwerem, są przesyłane przy pomocy zapytań asynchronicznych.

- UX – *User experience*, doświadczenia wizualne użytkownika aplikacji internetowej, pozwalają określić łatwość, z jaką korzysta się z aplikacji
- małe i średnie przedsiębiorstwa – Przedsiębiorstwa które zatrudniają od 10 do 250 pracowników.
- DRY – *Don't repeat yourself*, reguła stosowana podczas wytwarzania oprogramowania, zalecająca unikanie różnego rodzaju powtórzeń wykonywanych przez programistów – na przykład unikanie tych samych czynności podczas kompilowania, unikanie wklejania (lub pisania) tych samych (lub bardzo podobnych) fragmentów kodu w wielu miejscach. W przypadku powtarzającego się kodu źródłowego można separować powtarzający się fragment i jedynie odwoływać się do niego.
- API – *application programming interface*; tutaj: ściśle określony zestaw reguł i ich opisów, w jaki wtyczka komunikuje się z aplikacją eenet

## Rozdział 2

# Opis projektu



## Rozdział 3

# Rozwiązania konkurencyjne

### 3.1. Bitbucket



## Rozdział 4

# Zastosowane rozwiązania





## Rozdział 5

# Włożona w projekt praca



## Rozdział 6

# Osiągnięcia



## Rozdział 7

# Wyczka Zasoby



## Rozdział 8

# Wykonana praca przez poszczególne osoby w projekcie





## Rozdział 9

# Zawartość kodu dostarczonego wraz z pracą



# Bibliografia

- [1] dokumentacja AngularJS - <https://docs.angularjs.org/api>
- [2] dokumentacja Lo-dash - <http://lodash.com/docs>
- [3] dokumentacja jQuery - <http://lodash.com/docs>
- [4] dokumentacja Django - <https://docs.djangoproject.com/en/1.6/>
- [5] dokumentacja PostgreSQL - <http://www.postgresql.org/docs/>