|  |
| --- |
| Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie |
| Opracowanie aplikacji internetowej umożliwiającej interaktywną naukę języka angielskiego |
| Praca inżynierska |
|  |
| **Jakub Jaśkowiec** |
| **Kraków, 2012** |

|  |
| --- |
| [Wpisz tutaj streszczenie dokumentu. Streszczenie jest zazwyczaj krótkim podsumowaniem treści dokumentu. Wpisz tutaj streszczenie dokumentu. Streszczenie jest zazwyczaj krótkim podsumowaniem treści dokumentu.] |

Spis treści

[1. Wstęp 3](#_Toc312704803)

[2. Część teoretyczna 4](#_Toc312704804)

[Wykorzystane API 7](#_Toc312704805)

[Idea 8](#_Toc312704806)

[Diagramy Use Case 8](#_Toc312704807)

[3. Część praktyczna 8](#_Toc312704808)

[Tworzenie nowego słowa 8](#_Toc312704809)

[Zarządzanie kolekcjami 11](#_Toc312704810)

[Prezentacja kolekcji 11](#_Toc312704811)

# Wstęp

* 1. Cel pracy

Niniejsza praca ma na celu zaprezentowanie aplikacji internetowej umożliwiającej interaktywną naukę języka angielskiego oraz szczegółowe omówienie technologii, w jakiej została ona wykonana.

Głównym założeniem programu jest umożliwienie użytkownikowi uczenia się słów oraz zwrotów z języka angielskiego. Nauka powinna przebiegać w taki sposób, aby wymagała jak najmniejszego wysiłku, jednocześnie umożliwiając jak najlepsze zapamiętywanie prezentowanych treści.

Głównymi funkcjonalnościami aplikacji są:

1. Umożliwienie użytkownikowi tworzenia własnych słów wraz z ich szczegółowym opisem
2. Podział tworzonych słów na kolekcje tematyczne
3. Prezentacja stworzonych kolekcji w taki sposób, aby uczący mógł zapamiętać jak najwięcej słów

Pomysł na aplikację narodził się podczas aktywności na jednym z forów języka angielskiego. Umieszczone przeze mnie autorskie prezentacje wywołały pozytywny odzew wśród użytkowników. Aplikacja bazuje na wykorzystanym wtedy pomyśle, automatyzując proces tworzenia treści oraz wprowadzając nowe możliwości.

* 1. Zawartość pracy

Pierwsza część pracy to opis oraz analiza technologii użytej w projekcie. Zostanie dokonana analiza technologii Ruby on Rails, przyczyny jej użycia, zalety oraz wady. Przedstawione zostaną także inne technologie użyte w projekcie, takie jak: Javascript, MySql, używane gemy (pluginy) oraz zewnętrzne API.

Druga część to prezentacja aplikacji. Przedstawione zostaną funkcjonalności oraz scenariusze użycia programu, udokumentowane obrazami w formie zrzutów ekranu. Pokazana zostanie przydatność aplikacji w procesie uczenia się nowych słów oraz innowacyjność wykorzystanych w niej metod.

# Część teoretyczna

* 1. Podstawowe pojęcia używane w pracy:

Aplikacja, program, projekt – odnosi się tutaj do aplikacji internetowej, systemu komputerowego, który został stworzony na potrzeby tej pracy

Słowo, wyrażenie – wyraz lub kilka wyrazów, którymi użytkownik może zarządzać przy pomocy aplikacji

* 1. Technologie użyte w projekcie:

**Ruby** – jest językiem programowania stworzonym w 1995 w Japonii przez Yukihiro "Matza" Matsumoto. Posiada następujące cechy:

- interpretowany

nie jest kompilowany, lecz wykonywany poprzez interpreter

- zorientowany obiektowo

wszystko jest obiektem (włączając typy prymitywne, takie jak liczby całkowite, nil-wartości)

- dynamicznie typowany

sprawdzanie zgodności typów odbywa się w czasie wykonywania programu

- duck typing

rozpoznawanie obiektu odbywa się nie na podstawie deklaracji typu, lecz poprzez badanie metod udostępnionych przez obiekt

- garbage collector (odśmiecacz pamięci)

zarządzanie dynamiczną pamięcią wykonywane jest automatycznie, programista nie musi zwalniać przydzielonej pamięci

- domknięcia (closures)

elementy kodu mogą być traktowane jako obiekty

- obsługa wyjątków

obsługa sytuacji wyjątkowych możliwa jest poprzez mechanizm obsługi wyjątków

- system pluginów

dzięki systemowi zarządzania pakietami RubyGems, ułatwione jest zarządzanie zewnętrznymi pluginami

- implementacja na wszystkich popularnych platformach

Ruby działa m.in. na systemach Windows, Linux oraz Mac.

**Ruby on Rails** – jest frameworkiem do tworzenia aplikacji internetowych w języku Ruby. Został stworzony w 2004 roku przez Davida Heinemeier Hanssona. Udostępniony jest na zasadzie licencji Open Source.

Ruby on Rails korzysta z wzorca projektowego Model-View-Controller. Dzięki temu struktura aplikacji jest ściśle zdefiniowana, pozwala to na oddzielenie logiki aplikacji od interfejsu użytkownika. Za poszczególne moduły odpowiadają:

Model – ActiveRecord

Moduł odpowiedzialny za logikę aplikacji oraz komunikację z bazą danych.

Komunikacja z bazą danych jest standardowo realizowana przy pomocy systemu ORM, którym jest ActiveRecord. Framework umożliwia użycie innych systemów ORM.

Kod odpowiedzialny za logikę aplikacji umieszczony jest w plikach reprezentujących obiekty bazodanowe.

Kontroller – ActionController

Kontrolery odpowiedzialne są za sterowanie aplikacją i przepływ danych pomiędzy modelami a widokami

Widok – ActionView

Widoki tworzą interfejs użytkownika generowany przy użyciu danych z kontrolera oraz kodu HTML zapisanego w plikach widoków.

Framework dostarcza narzędzia, które ułatwiają programiście podstawowe czynności związane z tworzeniem aplikacji webowych. Jednym z takich narzędzi jest scaffolding, który pozwala na automatyczne generowanie kodu umożliwiającego uruchomienie podstawowej aplikacji.

Ruby on Rails posiada zintegrowane środowisko do testowania aplikacji. Domyślnym modułem testującym jest Test/Unit, jednak ze względu na rosnącą popularność narzędzia RSpec (używanego m.in. do Test Driven Development), został on użyty w tej aplikacji.

Do uruchomienia aplikacji wymagany jest odpowiedni serwer. W środowiskach deweloperskich dominuje serwer WEBrick oraz Mongrel, natomiast w środowisku produkcyjnym najpopularniejszym rozwiązaniem jest użycie modułu Phusion Passenger dla serwera Apache lub nginx.

Gemy:

Rspec

Haml

Active Scaffold

Wordnik

Paperclip

~~Devise~~

Coffee-rails

Sass

Mysql

Javascript – Jquery - Coffeescript

DOPISAC O JS, LIVE ITP

Interfejs został stworzony w języku angielskim. Dzięki temu używanie programu jest możliwe nie tylko przez ludzi polskojęzycznych, ale także innych obcokrajowców pragnących skorzystać z aplikacji w celu nauki języka angielskiego.

Przyczyny użycia Ruby on Rails:

- nowoczesny framework

- dobrze nadaje się do tego typu aplikacji

- duże wsparcie ze strony społeczności

*Dodać fragmenty kodu !*

## Wykorzystane API

Podczas procesu tworzenia nowego słowa, aplikacja wykorzystuje dwa zewnętrzne API:

1. Google Image Search API

Wykorzystywane jest w celu pobrania obrazów reprezentujących słowo.

Strona główna projektu: <http://code.google.com/apis/imagesearch/>

2. Wordnik API

API serwisu internetowego Wordnik.com umożliwia pobieranie definicji, przykładów użycia oraz synonimów dla danego słowa.

Używanie API umożliwia gem „wordnik” (<https://github.com/wordnik/wordnik-ruby>).

Gem jest używany w akcji „load data” kontrolera „expressions\_controller”.

Przykładowe użycie (pobieranie definicji):

@definitions = Wordnik.word.get\_definitions(@query, :useCanonical => 'true').map { |d| d["text"]}

## Idea

Dlaczego aplikacja jest innowacyjna do nauki:

- podczas opis słowa stara się korzystać z różnych technik zapamiętywania:

wzrokowo – obrazek

słuchowo– plik z wymową

budowanie sieci skojarzeń poprzez synonimy

zapamiętywanie kontekstu w jakim słowo zostało użyte – przykłady użycia

definicja - wytłumaczenie

- podczas przeglądania kolekcji (TODO)

## Diagramy Use Case

# Część praktyczna

## Tworzenie nowego słowa

Proces tworzenia nowego słowa został zaprojektowany tak, aby użytkownik na podstawie sugerowanych danych mógł samodzielnie opisać interesujące go słowo. Umożliwia to lepsze zapamiętywanie nowego słownictwa, gdyż pobudza kreatywność oraz pozwala stworzyć własne skojarzenia.

Na opis słowa składają się następujące elementy:

- nazwa,

- obraz,

- definicja,

- przykłady użycia,

- synonimy,

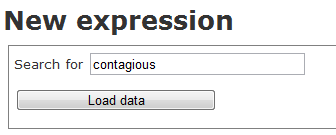
- notatki.

Dodatkowo, każde słowo można umieścić w kolekcji.

Poniżej zostanie omówiony proces tworzenia nowego wyrażenia, na przykładzie słowa ‘contagious’.

- w polu ‘Search for’ wpisujemy słowo / frazę

- klikamy ‘Load data’



Wyniki wyszukiwania zostają załadowane do ramki poniżej. Poszczególne pola zostały oddzielone poziomą linią.

Poniżej zaprezentowany zostanie stan poszczególnych pól po załadowaniu wyników.

Nazwa (Name)

Nazwa, pod jaką słowo zostanie zapisane w systemie. Domyślnie wypełniane jest wyrażeniem wpisanym do pola „Search for”. Po wielokrotnym skorzystaniu z akcji „Load data” wyświetlana jest historia wyszukiwania.

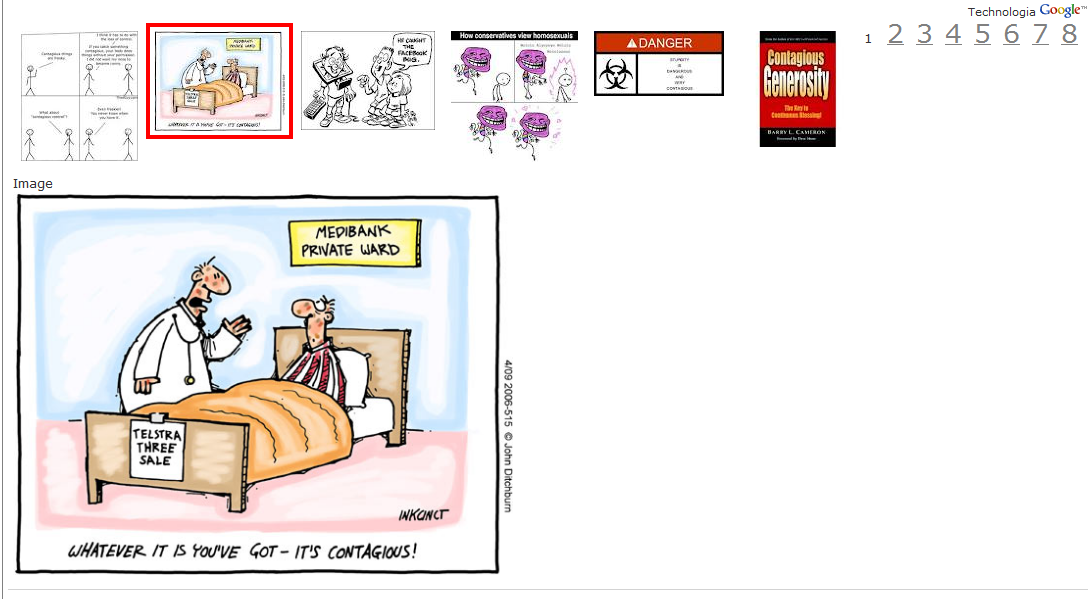
Obraz (Image)

Obraz, który jest dołączany do słowa.

Obrazy pobierane są z wykorzystaniem technologii Google, przy użyciu Google Images API.

Po załadowaniu danych ukazują się miniatury. Po kliknięciu w miniaturę, zostaje ona otoczona czerwoną ramką, a poniżej zostaje załadowany obraz w rozmiarach, w jakich może być zapisany w systemie. Po prawej widzimy szare cyfry, dzięki którym możemy załadować więcej miniatur.

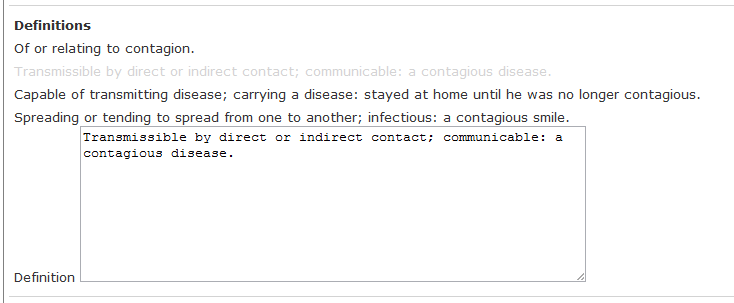
W ten sposób użytkownik z listy miniatur wybiera obraz, który reprezentuje słowo.



Definicje (Definitions)

Dostępna jest lista definicji do wyboru. Po kliknięciu w definicję, zostaje ona wpisana do pola tekstowego poniżej i oznaczona kolorem szarym.

W systemie jako definicja zostanie zapisana treść pola tekstowego. Użytkownik może dowolnie wybierać definicje z listy: może wybrać jedną lub wiele definicji, może także wpisać swoje własne, jeśli żadna z proponowanych nie jest satysfakcjonująca.



Przykłady użycia (Examples)

Synonimy (Synonyms)

Ich uzycie jest analogiczne do pola „definicja”

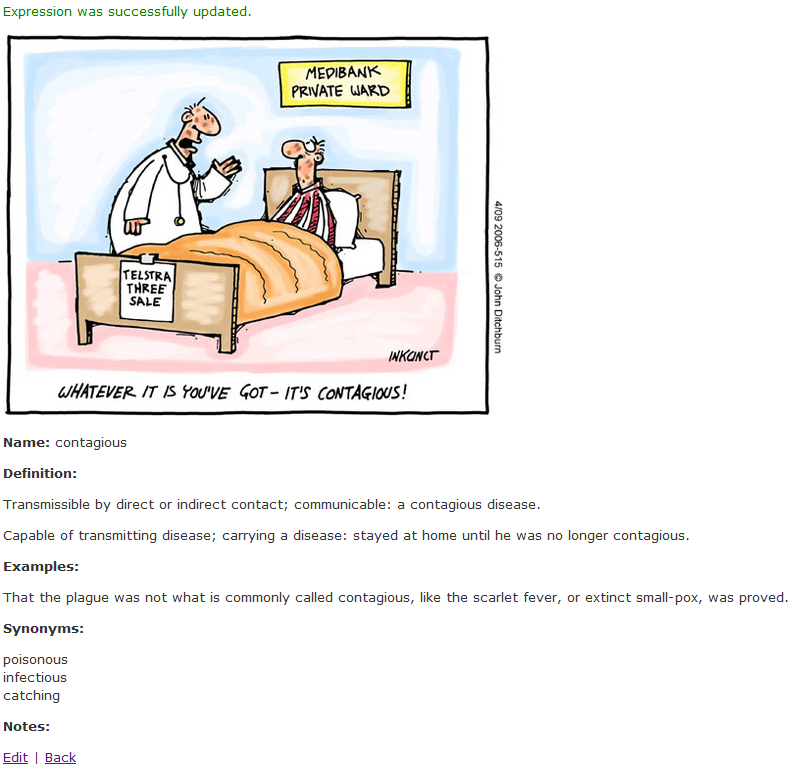
Kolekcja (Collection)

Umożliwia dodanie słowa do wybranej kolekcji.

Notatki (Notes)

Umożliwia dodanie własnych notatek dla słowa.

Po zapisaniu danych, użytkownik zostaje przeniesiony na stronę prezentującą stworzone słowo. Widnieją na niej wszystkie dane, które zostały wybrane podczas procesu tworzenia słowa.



## Zarządzanie kolekcjami

## Prezentacja kolekcji

Źródła:

wikipedia.org