Inżynieria e-systemów java

Autor:
Tymon Tobolski (181037)
Jacek Wieczorek (181043)
Mateusz Lenik(181142)

Prowadzący:
Dr inż. Katarzyna Nowak

Wydział Elektroniki
III rok
Śr 7.30 - 9.00

1 Opis projektu

Celem projektu jest stworzenie portalu, pozwalającego użytkownikowi na zakładanie i prowadzenie mikroblogów wraz zmożliwością follow'ania wpisów innych użytkowników. Aplikacja zostałą napisana w języku ruby z wykorzystaniem framework'a Ruby on Rails.

- Platforma : JVM
- Język impementacji : ruby, coffescript
- Framework : Ruby on Rails

2 Funkcjonalności systemu

- Obsługa użytkowników
  - Tworzenie konta
  - Edycja użytkownika
  - System autentykacji
  - Usuwanie użytkowników
  - Follow user
- Obsługa blogów
  - Tworzenie bloga
  - Zarządzanie blogiem
  - Tablica z postami użytkowników "follow"
- Obsługa postów
  - Dodawanie posta
  - Usuwanie posta
- Dyskusje
  - Tworzenie dyskusji
  - Zarządzanie dyskusjami
  - Dyskusje publiczne i prywatne

  - Zapraszanie użytkowników
  - Dodawanie postów w dyskusji
- Dodatkowe funkcjonalności
  - Powiązanie kont użytkowników z kontami facebook (logowanie przy użyciu facebook'a)
  - Pobieranie avatarów z gravatara lub facebook (zdjęcie profilowe)

3 Implementacja

System został zaimplementowany w architekturze MVC z wykorzystaniem architektury $REST$.

3.1 Kontrolery

3.1.1 Blogs Controller

Kontroler odpowiadający za funkcjonalności blogów : tworzenie, edytowanie, usuwanie, wyświetlanie, zabezpieczenie dostępu do edycji osób nie mających odpowiednich uprawnień.

3.1.2 Home Conroller

Kontroler odpowiadający za wyświetlanei stron statycznych np : About, Contact, etc.

3.1.3 Invitations Controller

Kontroler odpowiedzialny za dodawanie i usuwanie użytkowników z dyskusji.

3.1.4 OmniauthCallbacks Controller

Kontroler odpowiedzialny za możliwość logowania się przez konto Facebook.

3.1.5 Posts Controller

Kontroler odpowiedzialny za funkcjonalnośći postów.

3.1.6 Registration Controller

Kontroler odpowiedzialny za niestandardowe przeładowanie po rejestracji użytkownika.

3.1.7 Relationships Controller

Kontroler odpowiedzialny za akcje typu Follow/Unfollow user.

3.1.8 Sessions Controller

Kontroler odpowiedzialny za edycję profilu użytkownika

3.1.9 Inne

W celu stowrzenia systemu atentykacji i rejestracji użytkowników skorzystaliśmy z gotowej biblioteki *Devise*. Wszelkie dodatkowe akcje zawarte w kotrolerach : Session, Registration i OmniauthCallbacks pozwalają nam na dostosowanie systemu do naszych potrzeb w szybki i wygodny sposób.

3.2 Views

Wszelkei widoki zostały napisane przy uzyciu języka znaczników *Haml*. W celu generowania formularzy użyty został gem *SimpleForm*, a cały layout oparty o bibliotekę styli css - *Bootstrap*. Skrypty wykorzystujące *jQuery*, napisane zostały przy pomocy języka *CoffeScript*, kompilowalnego do kodu javascript.

Wykorzystane biblioteki js i css :

- bootstrap css
- bootstrap tabs
- facebox
- jQuery
- jQuery ui

3.3 Model

W projekcie wykorzystana została baza danych *PostgreSql* w wersji produkcyjnej i *SqlLite* w wersji deweloperskiej.

W celu łatwego dostepu do bazy danych i zarządzania nimi skorzystalimy z biblioteki *ActiveRecords* implementującej *ORM*.

W projekcie wystpują następujące modele :

- Blog
- Invitation
- Post
- Relationship
- User

3.4 Routes

Path name	Method	Path	Action
root		/	home#index
new_user_session	GET	/users/sign_in(.:format)	devise/sessions#new
user_session	POST	/users/sign_in(.:format)	devise/sessions#create
destroy_user_session	DELETE	/users/sign_out(.:format)	devise/sessions#destroy
user_omniauth_callback		/users/auth/:action/callback(.:format)	omniauth_callbacks#(?-mix:facebook)
user_password	POST	/users/password(.:format)	devise/passwords#create
new_user_password	GET	/users/password/new(.:format)	devise/passwords#new
edit_user_password	GET	/users/password/edit(.:format)	devise/passwords#edit
	PUT	/users/password(.:format)	devise/passwords#update
cancel_user_registration	GET	/users/cancel(.:format)	registrations#cancel
user_registration	POST	/users(.:format)	registrations#create
new_user_registration	GET	/users/sign_up(.:format)	registrations#new
edit_user_registration	GET	/users/edit(.:format)	registrations#edit
	PUT	/users(.:format)	registrations#update
	DELETE	/users(.:format)	registrations#destroy
profile	GET	/profile/:id(.:format)	sessions#show_profile
edit_profile	GET	/profile/:id/edit(.:format)	sessions#edit_profile
profile	PUT	/profile/:id(.:format)	sessions#update_profile
following	GET	/profile/:id/following(.:format)	sessions#following
followers	GET	/profile/:id/followers(.:format)	sessions#followers
delte_profile	DELETE	/profile/:id/delete(.:format)	sessions#delete_profile
blog_posts	POST	/blogs/:blog_id/posts(.:format)	posts#create
blog_post	DELETE	/blogs/:blog_id/posts/:id(.:format)	posts#destroy
blog_invitations	POST	/blogs/:blog_id/invitations(.:format)	invitations#create
blog_invitation	DELETE	/blogs/:blog_id/invitations/:id(.:format)	invitations#destroy
blogs	GET	/blogs(.:format)	blogs#index
	POST	/blogs(.:format)	blogs#create
new_blog	GET	/blogs/new(.:format)	blogs#new
edit_blog	GET	/blogs/:id/edit(.:format)	blogs#edit
blog	GET	/blogs/:id(.:format)	blogs#show
	PUT	/blogs/:id(.:format)	blogs#update
	DELETE	/blogs/:id(.:format)	blogs#destroy
relationships	POST	/relationships(.:format)	relationships#create
relationship	DELETE	/relationships/:id(.:format)	relationships#destroy
help		/help(.:format)	home#help
about		/about(.:format)	home#about
contact		/contact(.:format)	home#contact
userlist		/userlist(.:format)	home#user_list
unauthorized		/unauthorized(.:format)	home#unauthorized

Tabela 1: Ścieżki dostępne w aplikacji

4 Testy

TODO