Przykładowa aplikacja

KROK1.

Tworzymy nową aplikację rails new kwejk Inicjujemy 1 commita. Git init

git add . Git commit -m "first commit"

KROK2. Dodajemy nową aplikację na bitbucket oraz dodajemy źródło i robimy pierwszego commita na masterze.

KROK3. Dodanie rails_admin: Tworzymy nową branch railsadmin git checkout -b railsadmin

Dodajemy do Gemfile:

```
gem 'rails_admin'
```

Wykonujemy: bundle install

Następnie instalujemy rails_admin:

rails g rails_admin:install

Wykonujemy migrację bazy danych: rake db:migrate

Odpalamy serwer:

rails s

Do seeds dodajemy przykładowego Admina:

u = Admin.new(:email => "admin@example.com", :password => 'password', :password_confirmation => 'password')

u.save

Tworzymy commita: git commit -m "added railsadmin" Następnie mergujemy zmiany: git checkout master git merge railsadmin

git branch -D railsadmin

KROK4. Dodajemy devise. Git checkout -b devise

rails generate devise User

rails g devise:views

rake db:migrate dodajemy commita i mergujemy nasze zmiany.

KROK5. Instalujemy bootstrapa: git checkout -b twitter-bootstrap
Dodajemy gem:

```
gem "twitter-bootstrap-rails"
```

Instalujemy go

bundle install

Wykonujemy następnie polecenie:

```
rails generate bootstrap:install static
```

Instalujemy przykładowy szablon:

```
rails g bootstrap:layout application fluid
```

Dodajemy commita i mergujemy zmiany.

KROK6. Generujemy przykładowy Kontroler home z akcją index , ustawiamy root na tą akcję. Mergujemy zmiany.

KROK7. Dodajemy memy.

git checkout -b added_mems

rails g scaffold Mem name:string description:text user:references

rake db:migrate

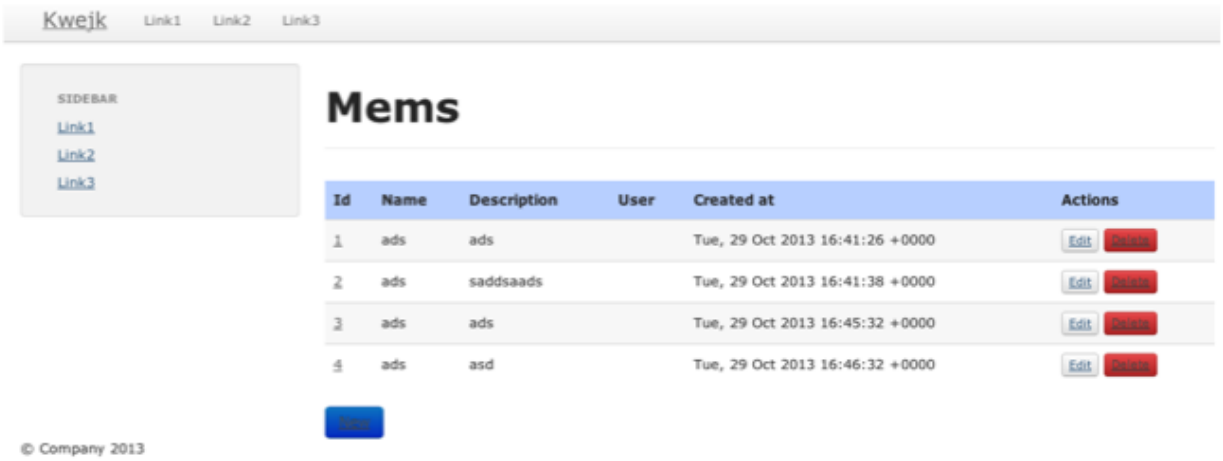
```
rails g bootstrap:themed Mems
```

Usuwamy z mems/_form.html.erb linijkę:

 <%= f.label :user_id, :class => 'control-label' %>

<%= f.text_field :user_id, :class => 'text_field' %>

Efekt powinien być następujący:



Następnie mergujemy zmiany.

KROK8. Dodajemy gem paperclip:

gem 'paperclip', github: 'thoughtbot/paperclip'

Wcześniej na linuxie musimy mieć zainstalowany imagemagick

(http://superuser.com/questions/163818/how-to-install-rmagick-on-ubuntu-10-04)

```
sudo apt-get install libmagickwand-dev imagemagick
```

KROK9. Dodajemy obrazek do Mema:

rails generate paperclip Mem image

rake db:migrate

Zmodyfikujmy plik modelu Mem:

class Mem < ActiveRecord::Base

belongs_to :user

has_attached_file :image, :styles => { :medium => "300x300>", :thumb => "100x100>" },

:default_url => "/images/:style/missing.png"

validates_attachment :image,

  :content_type => { :content_type => ["image/jpg", "image/gif", "image/png"] }

Oraz plik kontrolera dodając strong_parameters:

params.require(:mem).permit(:name, :description, :user_id,:image)

Dodatkowo modyfikujemy plik widoku:

Dodając:

```
 <div class="control-group">
    <%= f.label :image, :class => 'control-label' %>
    <div class="controls">
      <%= f.file_field :image, :class => 'file_field' %>
    </div>
  </div>
```

Oraz plik mems/show.html.erb Dodając:

<dd><%= image_tag @mem.image.url(:medium) %></dd>

KROK 10.

Dodajmy migrację dodającą aktywne i nie aktywne Memy: rails g migration addActiveToMem active:boolean

Modyfikujemy migracje:

```
class AddActiveToMem < ActiveRecord::Migration
  def change
    add_column :mems, :active, :boolean
    Mem.update_all(active: false)
  end
end
```

rake db:migrate

Dodajemy odpowiedni callback do modelu:

```
after_validation(on: :create) do
   self.active = false
 end
```

Modyfikujemy routes.rb:

```
 devise_for :admins
 devise_for :users
```

KROK11.

Zmieńmy plik mems/index.html.erb :

```
<%- model_class = Mem -%>
<div class="page-header">
  <% @mems.each do |mem| %>
      <p>
      <h1> <%= mem.name %> </h1>
      <%= link_to image_tag(mem.image.url),mem_path(mem) %>
     </p>
   </br>
    <% end %>

</div>
<%= link_to t('.new', :default => t("helpers.links.new")),
            new_mem_path,
```

:class => 'btn btn-primary' %>

Dodajmy także walidacje wystąpienia name do Mem:

validates :name, presence: true

KROK 12. Dodajmy asocjacje do User , który może mieć wiele memów.

Zmieniamy pliki:

User:

class User < ActiveRecord::Base # Include default devise modules. Others available are: # :confirmable, :lockable, :timeoutable and :omniauthable

devise :database_authenticatable, :registerable, :recoverable, :rememberable, :trackable, :validatable

has_many :mems

end

config/routes.rb:

resources :mems do

collection do

get 'my'

get 'inactive'

end

end

Modyfikujemy Mem.rb:

```
scope :active, ->{where active:true}
scope :inactive, ->{where active:false}
```

Modyfikujemy kontroler:

before_action :authenticate_user!, only: [:create,:my]

```
def index
  @mems = Mem.active
end

def my
  @mems = current_user.mems
  render :index
end

def inactive
```

```
    @mems = Mem.inactive
    render :index
  end
def create
    @mem = current_users.mems.new(mem_params)
```

KROK 13. Modyfikujemy widoki oraz routes.

```
root 'mems#index'
```

```
<ul class="nav">
            <li><%= link_to "Poczekelania", inactive_mems_path  %></li>
            <li><%= link_to "Memy", mems_path  %></li>
            <li><%= link_to "Moje memy", my_mems_path  %></li>
</ul>
 <ul class="nav nav-list">
            <li class="nav-header">Sidebar</li>
            <% if user_signed_in? %>
            <li>
            <%= link_to 'Edit account', edit_user_registration_path %>
            </li>
            <li>
            <%= link_to 'Logout', destroy_user_session_path, :method=>'delete' %>
            </li>
            <% else %>
            <li>
            <%= link_to 'Login', new_user_session_path %>
            </li>
            <li>
            <%= link_to 'Sign up', new_user_registration_path %>
            </li>
            <% end -%>
 </ul>
```

DALSZE KROKI:

-poprawić ładniejsze wyświetlanie obrazków -dodać losowo wybrane memy -dodać paginację -zmodyfikować i poprawić panel administracyjny np. o możliwość update zdjęć memów) -wrzucić na heroku (skorzystać z Amazonu s3 , konfiguracja)

-dodać integrację z facebookiem (logowanie , możliwość sharowania na facebooku). -dodać przykładowy formularz kontaktowy -dodać możliwość udostępniania przez API elementów dla aplikacji mobilnej