



**«Московский государственный технический университет  
имени Н.Э. Баумана»  
(МГТУ им. Н.Э. Баумана)**

---

ФАКУЛЬТЕТ \_\_\_\_\_ Информатика и системы управления

КАФЕДРА \_\_\_\_\_ Компьютерные системы и сети

**О т ч е т**  
**по зачетной работе**

Дисциплина: \_\_\_\_\_ Языки интернет программирования

---

Студент гр. ИУ6-31

15.12.15

А. В. Кирьяненко

(Подпись, дата)

(И.О. Фамилия)

Преподаватель

Р.С. Самарев

(Подпись дата)

(И.О. Фамилия)

## Зачетная работа.

**Задача:** Создать веб-приложение, которое позволяет владельцам магазинов, у которых установлены датчики Estimote Beacons, позволяющие отслеживать по Bluetooth местоположение в помещении, публиковать свои магазины, чтобы покупатели через приложение на смартфоне могли узнать, где находится тот или иной товар и как к нему пройти.

### Файл “MapShop\app\controllers\application\_controller.rb”

```
class ApplicationController < ActionController::Base
  # Prevent CSRF attacks by raising an exception.
  # For APIs, you may want to use :null_session instead.
  protect_from_forgery with: :exception

  before_action :set_locale
  def set_locale
    I18n.locale = params[:locale] || I18n.default_locale
  end
end
```

### Файл “MapShop\app\controllers\halls\_controller.rb”

```
require 'open-uri'
require 'nokogiri'

class HallsController < ApplicationController
  before_action :set_hall, only: [:show, :edit, :update, :destroy]

  before_filter :authenticate_user!, except: [:show, :index]

  # GET /halls
  # GET /halls.json
  def index
    @halls = Hall.all
  end

  # GET /halls/1
  # GET /halls/1.json
  def show
    if @hall.items_url != ''
      xml = open(@hall.items_url)
      @items = Nokogiri::XML(xml).xpath('/items/item').map do |item|
        p item
        {
          name: item.xpath('name/text()').to_s,
          description: item.xpath('description/text()').to_s,
          x: item.xpath('x/text()').to_s.to_i,
          y: item.xpath('y/text()').to_s.to_i,
          image: item.xpath('image/text()').to_s
        }
      end
    else
      @items = Item.where(hall_id: @hall.id)
      unless @items
        @items = @items.map do |item|
          {
            name: item.name,
            description: item.description,
            x: item.x,
            y: item.y,
            image: item.image.url
          }
        end
      end
    end
  end
end
```

```

        end
      end
    end

    # GET /halls/new
    def new
      @hall = Hall.new
    end

    # GET /halls/1/edit
    def edit
    end

    # POST /halls
    # POST /halls.json
    def create
      shop_id = hall_params[:shop_id]
      if shop_id.nil? ? false :
Brand.find_by_id(Shop.find_by_id(shop_id).brand_id).user_id == current_user.id
        @hall = Hall.new(hall_params)

        respond_to do |format|
          if @hall.save
            format.html { redirect_to @hall, notice: 'Hall was successfully
created.' }
            format.json { render :show, status: :created, location: @hall }
          else
            format.html { render :new }
            format.json { render json: @hall.errors, status: :unprocessable_entity }
          end
        end
      else
        # неправильный shop_id
        redirect_to '/shops/index'
      end
    end

    # PATCH/PUT /halls/1
    # PATCH/PUT /halls/1.json
    def update
      hall_params = params.require(:hall).permit(:name, :description, :scale,
:items_url, :map)
      respond_to do |format|
        if @hall.update(hall_params)
          format.html { redirect_to @hall, notice: 'Hall was successfully updated.'
}
          format.json { render :show, status: :ok, location: @hall }
        else
          format.html { render :edit }
          format.json { render json: @hall.errors, status: :unprocessable_entity }
        end
      end
    end

    # DELETE /halls/1
    # DELETE /halls/1.json
    def destroy
      @hall.destroy
      respond_to do |format|
        format.html { redirect_to halls_url, notice: 'Hall was successfully
destroyed.' }
        format.json { head :no_content }
      end
    end

    private
    # Use callbacks to share common setup or constraints between actions.

```

```

def set_hall
  @hall = Hall.find(params[:id])
end

# Never trust parameters from the scary internet, only allow the white list
through.
def hall_params
  params.require(:hall).permit(:name, :description, :scale, :items_url,
                                :shop_id, :map)
end
end

```

### Файл “MapShop\app\controllers\user\_controller.rb”

```

class UserController < ApplicationController
  before_filter :authenticate_user!

  def index
  end
end

```

### Файл “MapShop\app\models\hall.rb”

```

class Hall < ActiveRecord::Base
  has_attached_file :map
  validates_attachment_content_type :map, :content_type => %w(image/jpeg image/jpg
image/png)
  validates :scale, numericality: true
end

```

### Файл “MapShop\app\model\item.rb”

```

class Item < ActiveRecord::Base
  has_attached_file :image
  validates_attachment_content_type :image, :content_type => %w(image/jpeg
image/jpg image/png)
  validates :x, :y, numericality: true
end

```

### Файл “MapShop\app\views\halls\show.html.erb”

```

<p id="notice"><%= notice %></p>
<% shop = Shop.find_by_id(@hall.shop_id)
   brand = Brand.find_by_id(shop.brand_id) %>
<div class="row">
  <div class="col s12 m4 l3">
    <h1><%= brand.name %></h1>
    <h2><%= shop.name %></h2>
    <h3><%= @hall.name %></h3>

    <blockquote>
      <b>Описание:</b>
      <%= @hall.description %>
    </blockquote>

    <p>
      <b>Масштаб:</b>
      <%= @hall.scale %>
    </p>
  </div>
  <div class="col s12 m8 l9">
    <h4>Карта:</h4>
    <div style="position: relative;">
      <%= image_tag @hall.map.url%>
      <%= image_tag 'marker.png', id: 'marker', style: 'position: absolute;
display: none' %>
    </div>
  </div>
</div>
<ul class="collapsible popout" data-collapsible="accordion"><li>

```

```

<div class="collapsible-header">Товары:</div>
<div class="collapsible-body container">
  <div class="row">
    <% if @items
      @items.each do |item| %>
        <div class="col s12 m4" onMouseOut="mouseOut(this)"
          onMouseOver="mouseOver(this, <%= item[:x] * @hall.scale %>, <%=
item[:y] * @hall.scale %>)" ">
          <%= image_tag item[:image] %>
          <h5><i class="material-icons">shopping_basket</i><%= item[:name]
%></h5>
          <p><%= item[:description] %></p>
        </div>
      <% end %>
    <% end %>
  </div>
</div>
</li></ul>

<%= link_to 'Edit', edit_hall_path(@hall) %> |
<%= link_to 'Back', halls_path %>

```

### Файл “MapShop\app\views\halls\show.xml.erb”

```

<% shop = Shop.find_by_id(@hall.shop_id)
  brand = Brand.find_by_id(shop.brand_id)
  url = 'http://localhost:3000' %>
<?xml version="1.0" encoding="UTF-8" ?>
<hall>
  <brand>
    <name><%= brand.name %></name>
    <description><%= brand.description %></description>
  </brand>
  <shop>
    <name><%= shop.name %></name>
    <description><%= shop.description %></description>
    <location><%= shop.location %></location>
  </shop>

  <name><%= @hall.name %></name>
  <description><%= @hall.description %></description>
  <map><%= url + @hall.map.url %></map>
  <scale><%= @hall.scale %></scale>

  <items>
    <% if items = Item.where(hall_id: @hall.id)
      items.each do |item| %>
        <item>
          <name><%= item.name %></name>
          <description><%= item.description %></description>
          <image><%= url + item.image.url %></image>
          <x><%= item.x %></x>
          <y><%= item.y %></y>
        </item>
      <% end %>
    <% end %>
  </items>
</hall>

```

### Файл “MapShop\app\views\items\\_form.xml.erb”

```

<% hall = Hall.find_by_id(@item.hall_id) %>
<%= form_for(@item) do |f| %>
  <%= f.hidden_field(:hall_id, :value => params[:hall_id]) %>
  <% if @item.errors.any? %>
    <div id="error_explanation">
      <h2><%= pluralize(@item.errors.count, "error") %> prohibited this item
from being saved:</h2>

```

```

<ul>
  <% @item.errors.full_messages.each do |message| %>
    <li><%= message %></li>
  <% end %>
</ul>
</div>
<% end %>
<div class="row">
  <div class="col s12 m4 l3">
    <div class="field">
      <%= f.label :name %><br>
      <%= f.text_field :name %>
    </div>
    <div class="field">
      <%= f.label :description %><br>
      <%= f.text_area :description %>
    </div>
    <div class="field">
      <%= f.label :image %><br>
      <%= f.file_field :image %>
    </div>
    <div class="field">
      <%= f.label :x %><br>
      <%= f.text_field :x, id: 'field_x', oninput: "setMarker(this.value *
#{hall.scale})" %>
    </div>
    <div class="field">
      <%= f.label :y %><br>
      <%= f.text_field :y, id: 'field_y', oninput: "setMarker(undefined,
this.value * #{hall.scale})" %>
    </div>
    <div class="actions">
      <%= f.submit %>
    </div>
  </div>
  <div class="col s12 m8 l9">
    <h4>На карте:</h4>
    <div style="position: absolute;">
      <%= image_tag hall.map.url, onclick: "coords(event, this,
#{hall.scale})", id: 'map_id' %>
      <%= image_tag 'marker.png', id: 'marker',
style: "position: absolute; left: #{@item.x *
hall.scale}px; top: #{@item.y * hall.scale}px;" %>
    </div>
  </div>
</div>
<% end %>

```

### Файл “MapShop\app\assets\javascripts\hall.js”

```

function setMarkerHall(x, y) {
  var marker = document.getElementById('marker');
  marker.style.display = 'block';
  marker.style.left = (x - 12) + 'px';
  marker.style.top = (y - 24) + 'px';
}

function mouseOver(el, x, y) {
  setMarkerHall(x, y);
  el.style.backgroundColor = "cornsilk";
}

function mouseOut(el) {
  el.style.backgroundColor = "white";
}

```

### Файл “MapShop\app\assets\javascripts\item.js”

```

function coords(event, el, scale) { // Вставляет в поля текущие координаты при
  клике
    var event = event || window.event;
    var x = event.pageX - el.x
    var y = event.pageY - el.y
    setMarker(x, y);
    document.getElementById('field_x').value = (x + 0.0) / scale;
    document.getElementById('field_y').value = (y + 0.0) / scale;
  }

function setMarker(x, y) {
  var marker = document.getElementById('marker');
  if (x != undefined) marker.style.left = (x - 12) + 'px';
  if (y != undefined) marker.style.top = (y - 24) + 'px';
}

```

### Файл “MapShop\test\integration\integration\_test.rb”

```

require 'test_helper'
class UserFlowsTest < ActionDispatch::IntegrationTest
  test "login and browse site" do
    get "/users/sign_up"
    assert_response :success
    user = { email: 'test@test.ru', password: '12345678', password_confirmation:
'12345678' }
    post_via_redirect '/users', :user => user
    get '/user/index'
    assert_equal '/user/index', path
  end
  test "not login and not browse site" do
    get "/users/sign_in"
    assert_response :success
    user = { email: 'error@test.ru', password: 'error' }
    post_via_redirect '/users/sign_in', :user => user
    get '/user/index'
    assert_equal '/unauthenticated', path
  end
end

```

### Файл “MapShop\test\controllers\halls\_controller\_test.rb”

```

require 'test_helper'

class HallsControllerTest < ActionController::TestCase
  setup do
    @brand = Brand.create()
    @shop = Shop.create(brand_id: @brand.id)
    @hall = Hall.create(scale: 1, shop_id: @shop.id, items_url: '')
  end

  test "should get index" do
    get :index
    assert_response :success
    assert_not_nil assigns(:halls)
  end

  test "should get show" do
    get :show, id: @hall
    assert_response :success
  end

  test "should get format html" do
    get :show, id: @hall
    assert_select 'html', true
  end

  test "should get format xml" do

```

```

    get :show, id: @hall, format: 'xml'
    assert_select 'hall', true
  end
end

```

## Файл “MapShop\test\models\item\_test.rb”

```

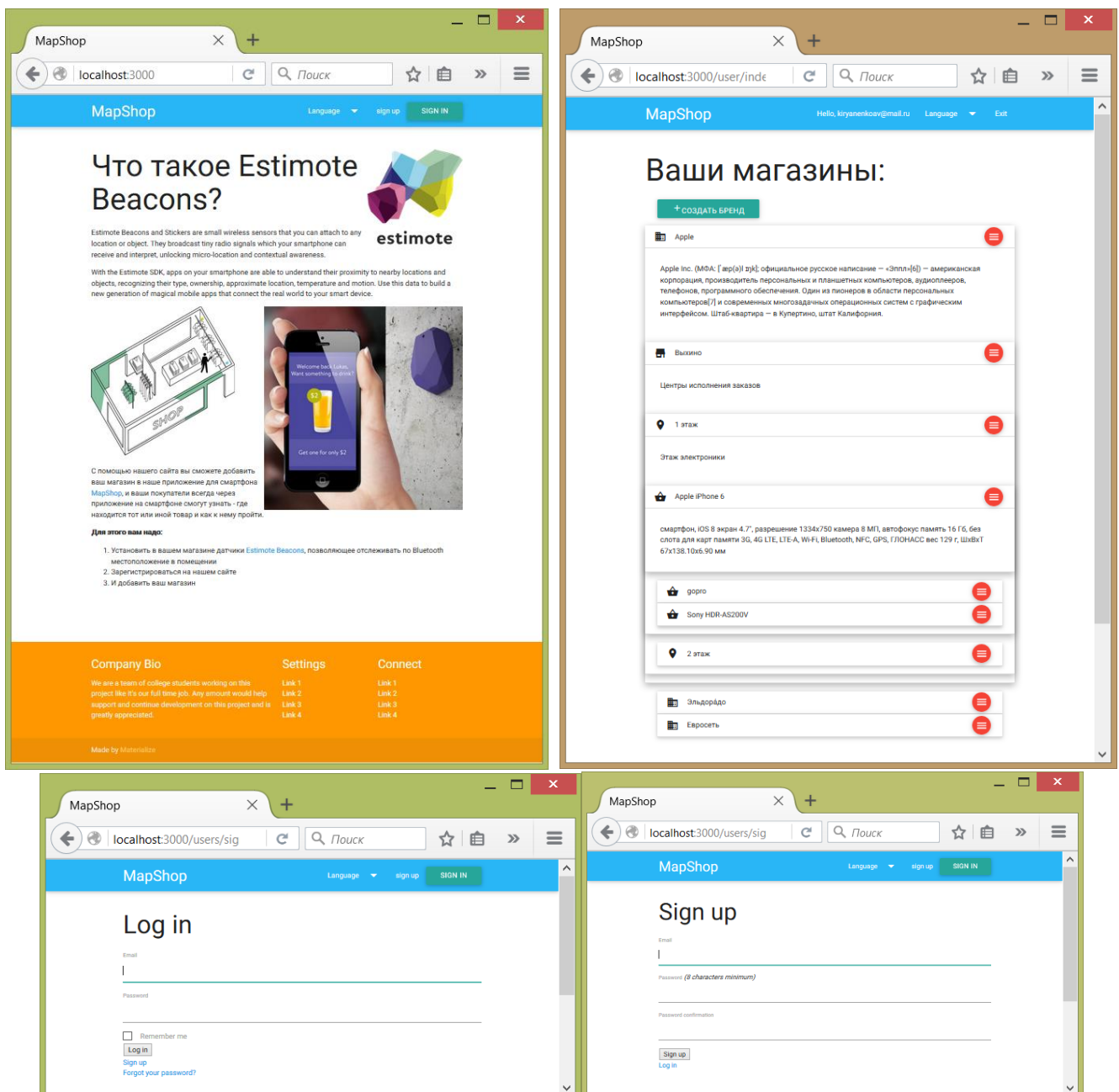
require 'test_helper'

class ItemTest < ActiveSupport::TestCase
  test 'should not save string x' do
    rec = Item.new(x: 'kgui')
    assert !rec.save
  end

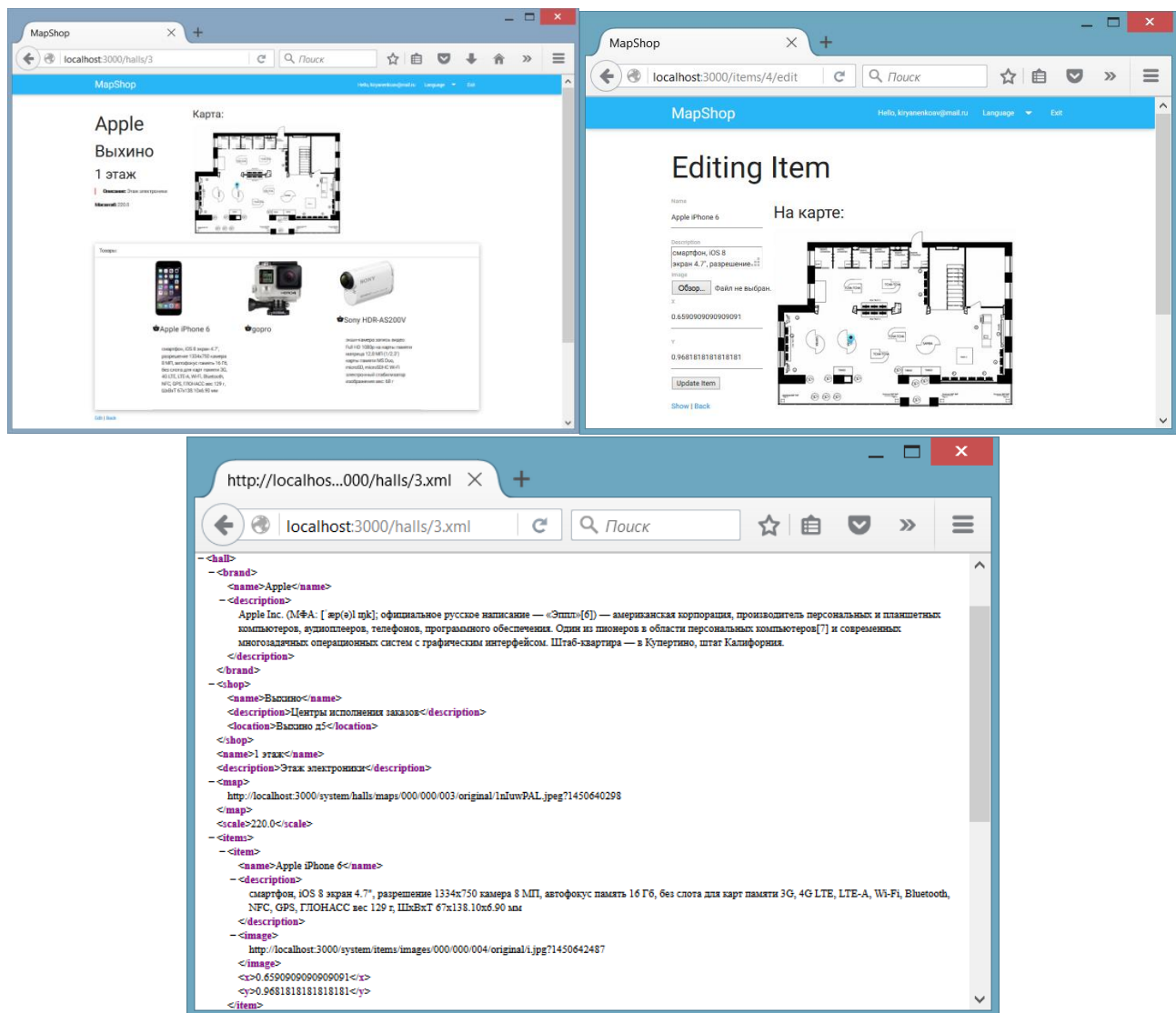
  test 'should not save string i' do
    rec = Item.new(y: 'kgui')
    assert !rec.save
  end
end

```

## Изображения форм интерфейса







## Результаты тестов

```

C:\Users\Kirya_000\Documents\Rails\ZR\MapShop>rake test
DL is deprecated, please use Fiddle
C:/Ruby21/bin/rake: No such file or directory - chcp
Run options: --seed 64135

# Running:

.....

Finished in 0.908568s, 17.6101 runs/s, 24.2139 assertions/s.

16 runs, 22 assertions, 0 failures, 0 errors, 0 skips
  
```

**Вывод:** Таким образом, я на этом курсе познакомился с фреймворком Ruby on Rails и научился создавать сложные веб-приложения.