**Red Team**

**Stargazer**

**Automatic Telescope Control System**

**Testing Plan**

**Team Members:**

Rob Grmek

Robert Smith

**Instructor:**

**Youry Khmelevsky**

**Course:**

**COSC 471**

**Date:**

**April 10th, 2009**

# ****Revision History****

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/21/2009 | 1.0 | Document created. | Robert, Rob |
| 11/08/2009 | 1.1 | Added unit testing information. | Rob |
| 11/18/2009 | 1.2 | Added unit tests. | Rob |
| 11/29/2009 | 1.3 | Updated unit tests. | Rob |
| 1/22/2010 | 1.4 | Minor updates. | Rob |
| 2/15/2010 | 1.5 | Minor updates. | Rob |
| 4/10/2010 | 1.6 | Updated test case code examples. | Rob |

# ****Table of Contents****

**Page**

|  |  |
| --- | --- |
| Unit Tests | 1 |
| Acceptance Tests | 4 |

# Unit Tests

For unit testing, we will be using Ruby’s built-in unit testing. These tests will build on each other to form our test suite. Also, for accessing the server and setting up a Ruby on Rails development environment, see the developer’s guide.

For tutorials on using Ruby’s unit testing with Ruby on Rails, see here:

* <http://www.tutorialspoint.com/ruby-on-rails-2.1/rails-unit-testing.htm>
* <http://guides.rubyonrails.org/testing.html>
* <http://andrzejonsoftware.blogspot.com/2007/05/15-tdd-steps-to-create-rails.html>

**Example 1: Miscellaneous examples and commands:**

Creating a test case:

|  |
| --- |
| require 'test\_helper'  class PostTest < ActiveSupport::TestCase  def test\_should\_not\_save\_post\_without\_title  post = Post.new  assert !post.save, "Saved the post without a title"  end  end |

To run that single test method:

|  |
| --- |
| # from project directory root  cd test  ruby unit/post\_test.rb -n test\_should\_not\_save\_post\_without\_title |

To run the single test case:

|  |
| --- |
| # from project directory root  cd test  ruby unit/post\_test.rb |

To run all unit tests and functional tests:

|  |
| --- |
| rake test |

To run only all of the unit tests:

|  |
| --- |
| rake test:units |

To run only all of the functional tests:

|  |
| --- |
| rake test:functionals |

To run only recent tests for models and controllers that have been modified in last 10 minutes:

|  |
| --- |
| rake test:recent |

**Example 2: Partial Functional Test (from SPIKE project):**

|  |
| --- |
| ###############################################################  # File: posts\_controller\_test.rb  # Project: SPIKE  # Author: Red Team  # Desc: Functional test for the "posts" controller.  ###############################################################  require 'test\_helper'  class PostsControllerTest < ActionController::TestCase  # Test index action of the Post controller will collect all Post objects  # and displays a listing of all Post objects  test "should get index" do  get :index  assert\_response :success  assert\_not\_nil assigns(:posts)  end  # Test new action of the Post controller displays the new.html.erb view  test "should get new" do  get :new  assert\_response :success  end  # Test create action of the Post controller creates a Post object  test "should create post" do  assert\_difference('Post.count') do  post :create, :post => { :name => "Bob", :title => "Something"}  end  assert\_redirected\_to post\_path(assigns(:post))  assert\_equal "Your post is saved!", flash[:notice]  end  # Test show action of the Post controller displays the Post object’s data  test "should show post" do  get :show, :id => posts(:one).to\_param  assert\_response :success  end  # Test destroy action of the Post controller deletes the Post object  test "should destroy post" do  assert\_difference('Post.count', -1) do  delete :destroy, :id => posts(:one).to\_param  end  # ensure the user is redirected back to the index after the Post is  # deleted  assert\_redirected\_to posts\_path  end  end |

**Example 3: Partial Unit Test for the User Model (in /Stargazer/test/unit/):**

|  |
| --- |
| ################################################################  # File: user\_test.rb  # Project: Stargazer  # Author: Red Team  # Desc: Unit test for the "user" model.  ################################################################  require 'test\_helper'  class UserTest < ActiveSupport::TestCase    # Create a user with necessary, correct input values for the  # following tests  def setup  @user = User.new  @user.username = "unit\_test"  @user.email = "unit@test.com"  @user.password = "test"  @user.password\_confirmation = "test"  end  # Ensure user’s email address must look like a valid address  def test\_email\_must\_look\_like\_an\_email\_address  # give it a non-valid looking email which should fail  @user.email = "test.ca"  assert !@user.save  @user.email = "test@ca"  assert !@user.save  @user.email = "test@testing.testing"  assert !@user.save  @user.email = "@testing.testing"  assert !@user.save  @user.email = "test@.com"  assert !@user.save    # give it a valid looking email which should pass  @user.email = "test@test.ca"  assert @user.save  end  end |

To see the full file and other unit test files, please view the source code for our unit tests on our Stargazer repository on GitHub: <http://github.com/RedTeamCOSC470/Stargazer>

# Acceptance Tests

#### Information for Testing Live Web Application:

Website can be seen at: <http://cis470star09f.okanagan.bc.ca>

HTTPS is also enabled but the server’s certificate is self-signed, currently. The website with SSL can be seen at: <https://cis470star09f.okanagan.bc.ca>

To login with an account with admin privileges, use:

Username: admin

Password: stargazer09

To login with an account with regular user privileges, use:

Username: user

Password: test

Reminder: the telescope must be turned on before the jobs which control the telescope are scheduled to run.