

Project Assessment Skills Web Application

Samuel Goff

Kennedy Space Center

Major: Computer Science

KSC FO Summer Session

Date: 24 07 2013

Project Assessment Skills Web Application

Samuel J. Goff¹

University of Central Florida, Orlando, Florida, 32816

The purpose of this project is to utilize Ruby on Rails to create a web application that will replace a spreadsheet keeping track of training courses and tasks. The goal is to create a fast and easy to use web application that will allow users to track progress on training courses. This application will allow users to update and keep track of all of the training required of them. The training courses will be organized by group and by user, making readability easier. This will also allow group leads and administrators to get a sense of how everyone is progressing in training. Currently, updating and finding information from this spreadsheet is a long and tedious task. By upgrading to a web application, finding and updating information will be easier than ever as well as adding new training courses and tasks. Accessing this data will be much easier in that users just have to go to a website and log in with NDC credentials rather than request the relevant spreadsheet from the holder. In addition to Ruby on Rails, I will be using JavaScript, CSS, and jQuery to help add functionality and ease of use to my web application. This web application will include a number of features that will help update and track progress on training. For example, one feature will be to track progress of a whole group of users to be able to see how the group as a whole is progressing. Another feature will be to assign tasks to either a user or a group of users. All of these together will create a user friendly and functional web application.

Nomenclature

<i>AJAX</i>	= Asynchronous JavaScript and XML
<i>CSS</i>	= Cascading Style Sheets
<i>DRY</i>	= Don't Repeat Yourself
<i>HTML</i>	= Hypertext Markup Language
<i>HTTP</i>	= Hypertext Transfer Protocol
<i>IDE</i>	= Integrated Development Environment
<i>MVC</i>	= Model-View-Controller
<i>PDF</i>	= Portable Document Format
<i>RoR</i>	= Ruby on Rails
<i>RVM</i>	= Ruby Version Manager
<i>SQL</i>	= Structured Query Language
<i>URL</i>	= Uniform Resource Locator
<i>XML</i>	= Extensible Markup Language

I. Introduction

The Project Assessment Skills Web Application was brought into existence from a need to efficiently and easily manage and observe the skills that project groups are assigned. The old method of doing this is by keeping track of a complex Excel spreadsheet that required tedious duplication in order to keep track of skills. My task was to create a web application that would accomplish the same task as a spreadsheet while being easier to use and manage. I used Ruby on Rails to design the entire web application. Ruby on Rails is a web application framework that allows for easy manipulation of databases, web servers, and templates. While Ruby on Rails is the center of my project, it is not the only tool I used to create the web application. I also made use of JavaScript and jQuery to add functionality to my web application. JavaScript is a scripting language that mainly serves as a way to interface with the user in my project. jQuery is a library of JavaScript that further enhances the interface with the

¹ Intern, NE-C2, Kennedy Space Center. University of Central Florida

user. With these tools combined I was able to effectively make a web application that satisfied the criteria required of it.

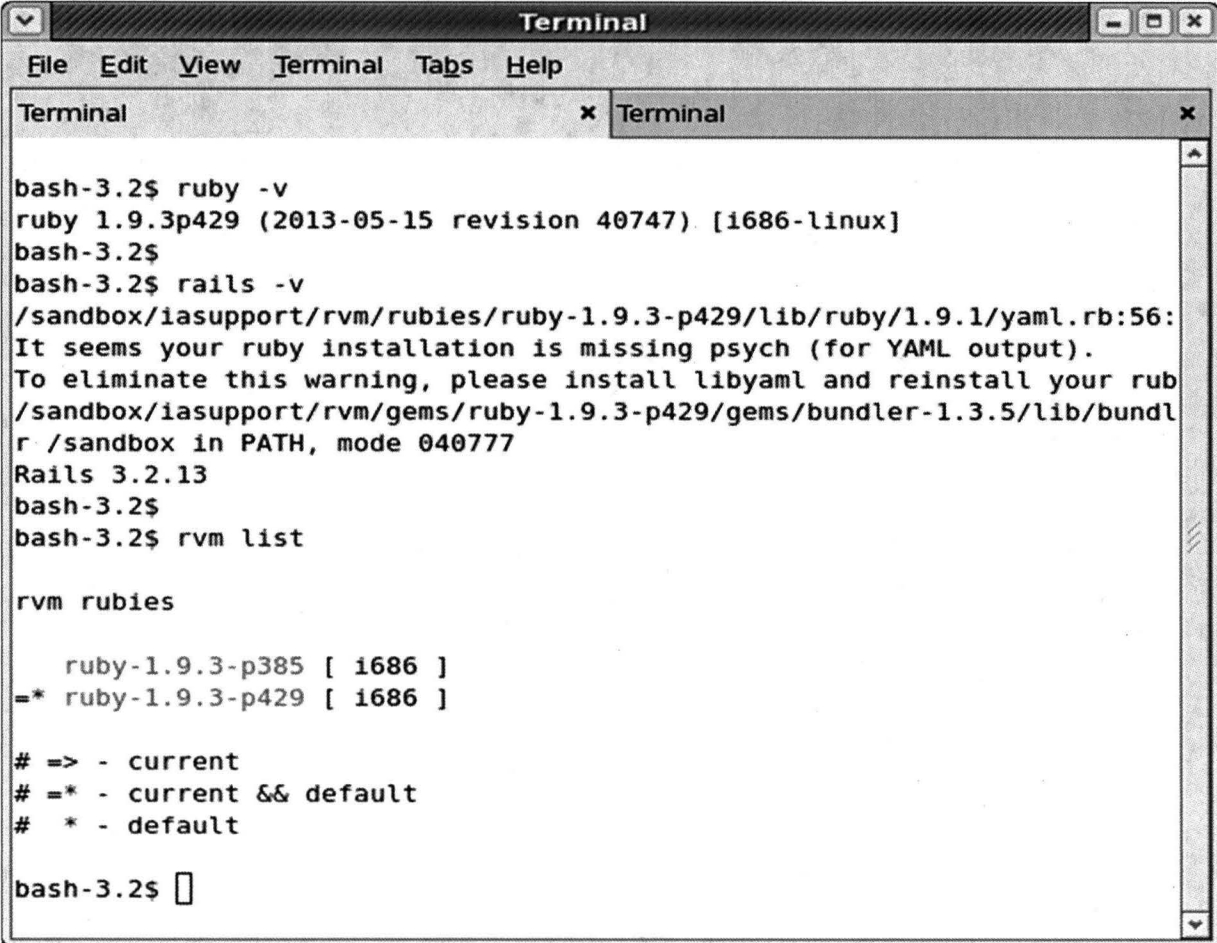
II. Tools Description

Ruby on Rails was developed by David Heinemeier Hansson and first released in 2005. Since then Ruby on Rails has grown significantly in usability and in community support. For my project, I used Ruby 1.9.3 with Rails 3.2 on Red Hat Enterprise Linux. Another tool that is essential to web development is the command line. In the command line I can run a server, test code in the Ruby console, and run tests. An important tool for setting up Ruby on Rails on my machine is RVM. RVM is a tool used to easily install and manage multiple versions of Ruby on a Linux machine. A tool that helped speed up the process of creating my web application was JetBrains RubyMine³. RubyMine is a Ruby IDE that makes editing a Ruby on Rails project easy. RubyMine includes a debugger for Ruby as well as a debugger for JavaScript.

Ruby has a package managing tool called RubyGems and it provides a great format for adding libraries and programs to your project through “gems”. I used gems frequently to add many different features to my project. Some examples of the features I incorporated include pagination, styling, testing, error handling, authentication, form building and data exportation. A particularly important gem that I used for testing would be the Cucumber gem. Cucumber allows me to run tests on my entire web application to make sure that nothing goes wrong when it shouldn't.

III. Setting up Ruby on Rails

Ruby on Rails has a few dependencies that are required to make web applications. The first and most important



```

Terminal
File Edit View Terminal Tabs Help

Terminal x Terminal x

bash-3.2$ ruby -v
ruby 1.9.3p429 (2013-05-15 revision 40747) [i686-linux]
bash-3.2$
bash-3.2$ rails -v
/sandbox/iasupport/rvm/rubies/ruby-1.9.3-p429/lib/ruby/1.9.1/yaml.rb:56:
It seems your ruby installation is missing psych (for YAML output).
To eliminate this warning, please install libyaml and reinstall your rub
/sandbox/iasupport/rvm/gems/ruby-1.9.3-p429/gems/bundler-1.3.5/lib/bundl
r /sandbox in PATH, mode 040777
Rails 3.2.13
bash-3.2$
bash-3.2$ rvm list

rvm rubies

   ruby-1.9.3-p385 [ 1686 ]
=*  ruby-1.9.3-p429 [ 1686 ]

# => - current
# =* - current && default
# *  - default

bash-3.2$ █

```

Figure 1. Ruby and Ruby on Rails Versions with RVM.

of these is Ruby. On Linux, Ruby is a little difficult to install manually. This is where RVM comes in to play, with RVM I installed Ruby 1.9.3 easily and installed Ruby on Rails 3.2.

In order to start doing work with Ruby on Rails, I had to accomplish two things. First, I needed to start up RubyMine and open up my project directory. Then in the console I needed to start my server. The server is started by the command `$ rails s`. Once the server is running I can now check what the web application displays by simply going to the URL `localhost:3000` in any web browser². Once I am at the URL I can see exactly what my web application displays and make changes dynamically to change what the application displays.

Once that step is complete Ruby on Rails is fully set up and ready to create whatever web application the user desires. This ease of use is in line with the Ruby on Rails philosophy of maximizing programmer happiness. Ruby on Rails also utilizes many customizations that fit the user’s needs. For example, in Ruby on Rails one can choose what type of database they would like to use, whether it be SQLite or PostgreSQL. For my project I used PostgreSQL to manage my database. Ruby on Rails follows a Model-View-Controller design principle. MVC is the principle that the model should contain data for an application and most of the logic. The controller interacts with the model and sends data to the view. The view interacts with the user by presenting and sending data to and from the controller.

IV. Project Description

The Project Assessment Skills Web Application was created from a spreadsheet called the Project Assessment Skills Spreadsheet. The purpose is to create a way for project group leads to manage the training courses for the users in their respective groups. In my project each training course is called a task, and a task has many attributes that it is associated with. These tasks can be assigned to users or project groups easily by administrators or project leads. Once assigned the users can then look to see what tasks they must complete. A user can input information such as a planned date of completion and any notes they would like to relay to the product group lead about the task.

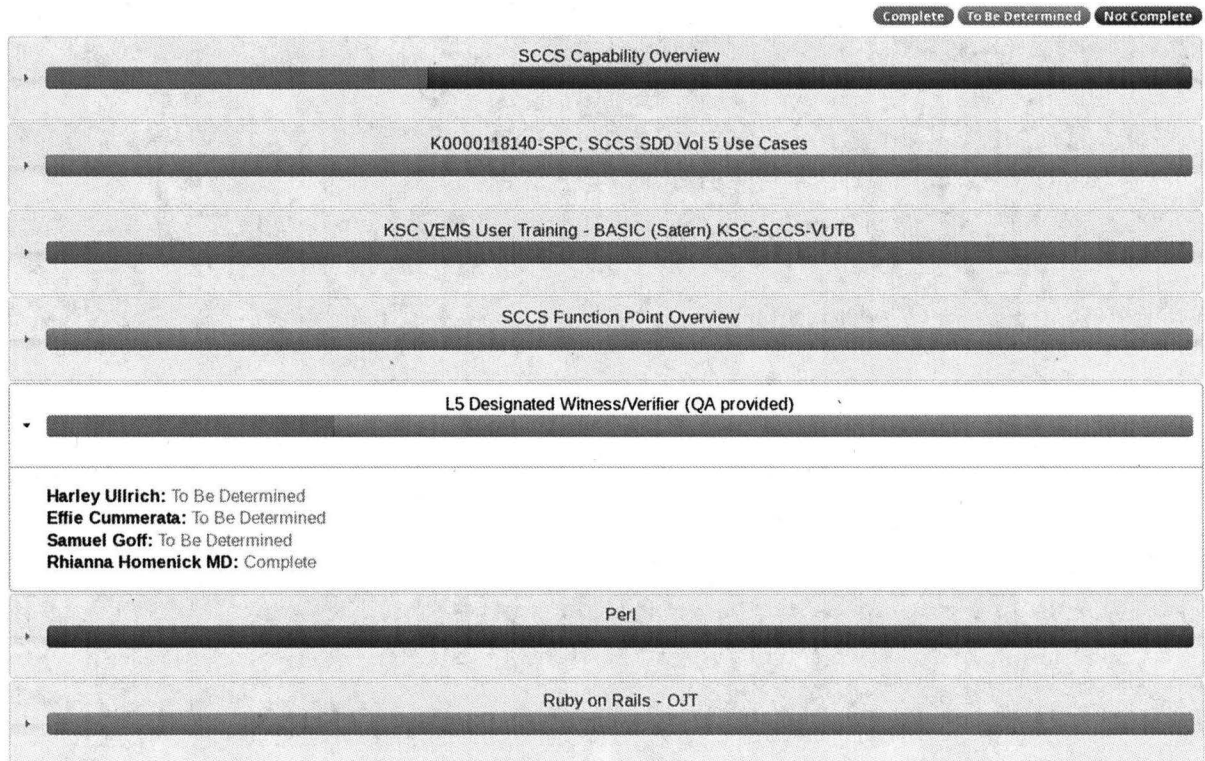


Figure 2. A view of the IA Group that displays all of the tasks and users for the group.

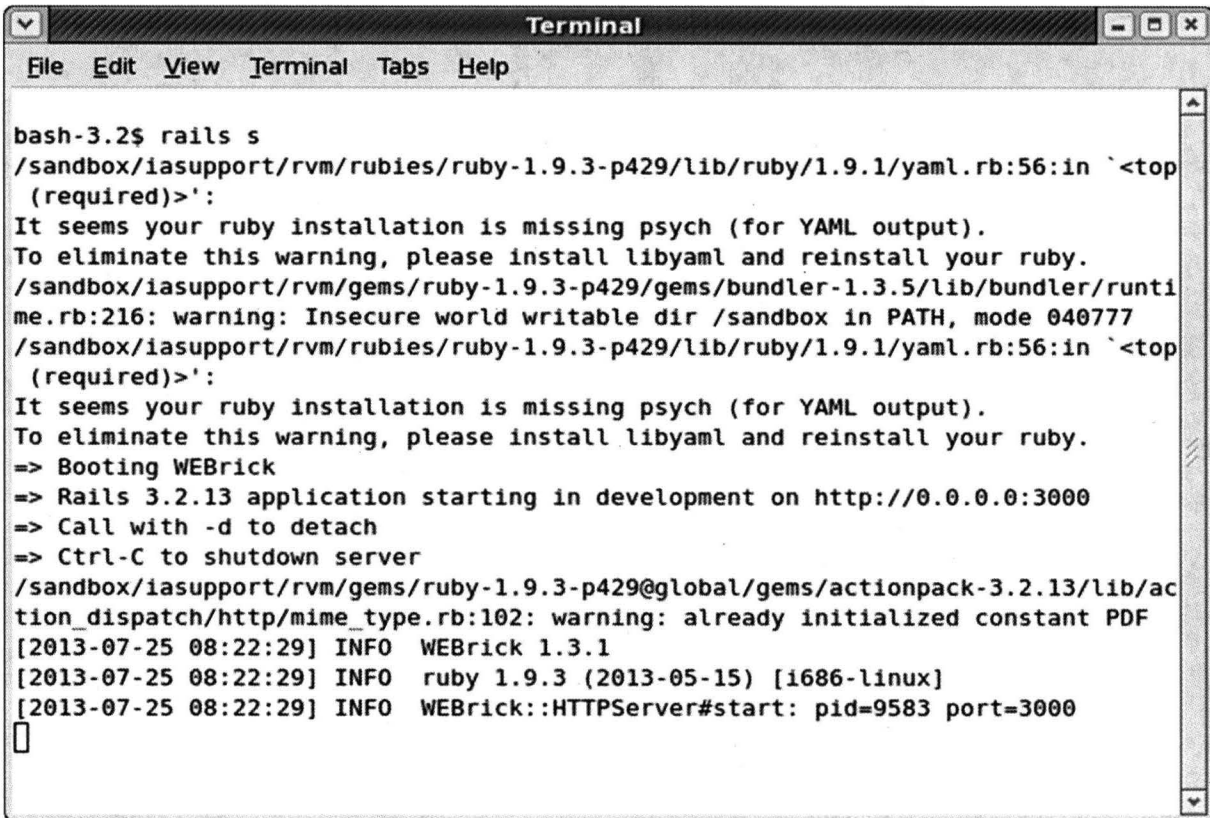
In addition to assigning tasks administrators can also create new tasks. An administrator can also create groups and assign users to a group within this application. A project group lead or administrator has a variety of ways to view tasks. The application sorts tasks by group, by user, and by all. Another feature is that leads will be able to view the progress of a task in a group or by all of the users that task is assigned to. This is displayed as a progress

bar under the task name. Users can only view and edit their own tasks but administrators can create, edit, and remove tasks.

In addition to viewing the tasks on the web site, there is also functionality to download the task data as a PDF document or an Excel spread sheet. This allows better distribution of the data and will provide leads with more ways to keep track of their users. Any data that is displayed on my web application is also paginated if there is too much for one page. All of these features combined creates a clean, easy to use application that is accessible to anyone with a computer.

V. Project Methodology

Once my server was running and RubyMine was open I was able to begin creating my application. The project



```

Terminal
File Edit View Terminal Tabs Help

bash-3.2$ rails s
/sandbox/iasupport/rvm/rubies/ruby-1.9.3-p429/lib/ruby/1.9.1/yaml.rb:56:in `<top
(required)>':
It seems your ruby installation is missing psych (for YAML output).
To eliminate this warning, please install libyaml and reinstall your ruby.
/sandbox/iasupport/rvm/gems/ruby-1.9.3-p429/gems/bundler-1.3.5/lib/bundler/runti
me.rb:216: warning: Insecure world writable dir /sandbox in PATH, mode 040777
/sandbox/iasupport/rvm/rubies/ruby-1.9.3-p429/lib/ruby/1.9.1/yaml.rb:56:in `<top
(required)>':
It seems your ruby installation is missing psych (for YAML output).
To eliminate this warning, please install libyaml and reinstall your ruby.
=> Booting WEBrick
=> Rails 3.2.13 application starting in development on http://0.0.0.0:3000
=> Call with -d to detach
=> Ctrl-C to shutdown server
/sandbox/iasupport/rvm/gems/ruby-1.9.3-p429@global/gems/actionpack-3.2.13/lib/ac
tion_dispatch/http/mime_type.rb:102: warning: already initialized constant PDF
[2013-07-25 08:22:29] INFO WEBrick 1.3.1
[2013-07-25 08:22:29] INFO ruby 1.9.3 (2013-05-15) [i686-linux]
[2013-07-25 08:22:29] INFO WEBrick::HTTPServer#start: pid=9583 port=3000

```

Figure 3. Example of a server running with rails.

was first devised as a list of tasks, with a task just having a name, a skill, and a category. Once that was completed I added user functionality through the devise gem. The devise gem automatically creates a login/logout form as well as a column in the database to store user information. Once that tasks and users were created I then created groups to help sort the users. I used unobtrusive JavaScript to add or remove users without leaving the group page. I also made use of jQuery and AJAX requests in most of my forms. I used jQuery to make a calendar popup when a user enters information in a date field. In every form I created in my project, I used jQuery to make the form appear as a modal. A modal is a window that appears over the “parent” window and can be opened and closed at the touch of a button. jQuery has a widget called jQuery accordion that allowed me to hide secondary information on the group page but still display it all in an intuitive way. I then created the progress bar functionality by querying the server for an array of all the tasks in a group and performing some math to get a percentage based on the user’s statuses.

For styling I used a free style framework called Twitter Bootstrap. Bootstrap allows me to effortlessly implement style sheets and features to my project by including the bootstrap gem. Bootstrap is what I used to display everything in my application from the forms to the navigation menu to the progress bars. There are also themes for Bootstrap with I used for my application. These themes are called Bootswatch themes and they are easy to incorporate into any web application.

VI. Testing and Debugging

In my project all of the testing was done by Cucumber¹. In Cucumber, I can write out features that I want in my

```

Terminal
File Edit View Terminal Tabs Help
Then I should see a completion status error # features/step_definitions/task_errors_steps.rb:19

Scenario: I cannot create a Task for a Group with no Users # features/manage_task_errors.feature:28
  Given I am on the Group show page # features/step_definitions/task_group_steps.rb:9
  When I click on the task button # features/step_definitions/task_group_steps.rb:35
  And I create the task # features/step_definitions/task_group_steps.rb:39
  Then I should see a no users error # features/step_definitions/task_errors_steps.rb:24

Scenario: I cannot create the same Task more than once per User # features/manage_task_errors.feature:34
  Given I click on the task button # features/step_definitions/task_group_steps.rb:35
  And I create the task for a User # features/step_definitions/task_user_steps.rb:5
  Then I should see the task # features/step_definitions/task_group_steps.rb:47
  Given I am on the Task index page # features/step_definitions/task_user_steps.rb:1
  When I click on the New Task button # features/step_definitions/task_errors_steps.rb:1
  And I create the task for a User # features/step_definitions/task_user_steps.rb:5
  Then I should see duplication error # features/step_definitions/task_errors_steps.rb:28

Feature: Manage Tasks for Group

Background: # features/manage_tasks_group.feature:2
  Given I am logged in # features/step_definitions/task_group_steps.rb:1
  And I have an Element # features/step_definitions/task_group_steps.rb:21
  And I have Completion Statuses # features/step_definitions/task_group_steps.rb:25
  And I am on the Group show page # features/step_definitions/task_group_steps.rb:9
  And the Group has at least one User # features/step_definitions/task_group_steps.rb:14

@javascript
Scenario: Create Task # features/manage_tasks_group.feature:10
  When I click on the task button # features/step_definitions/task_group_steps.rb:35
  And I create the task # features/step_definitions/task_group_steps.rb:39
  Then I should see the task # features/step_definitions/task_group_steps.rb:47

@javascript
Feature: Manage Task for Current User

Background: # features/manage_tasks_user.feature:3
  Given I am logged in # features/step_definitions/task_group_steps.rb:1
  And I have an Element # features/step_definitions/task_group_steps.rb:21
  And I have Completion Statuses # features/step_definitions/task_group_steps.rb:25
  And I am on the Task index page # features/step_definitions/task_user_steps.rb:1

Scenario: Create Task # features/manage_tasks_user.feature:9
  When I click on the task button # features/step_definitions/task_group_steps.rb:35
  And I create the task for a User # features/step_definitions/task_user_steps.rb:5
  Then I should see the task # features/step_definitions/task_group_steps.rb:47

Scenario: Complete Task # features/manage_tasks_user.feature:15
  When I click on the task button # features/step_definitions/task_group_steps.rb:35
  And I create the task for a User # features/step_definitions/task_user_steps.rb:5
  And I click on the complete button # features/step_definitions/task_user_steps.rb:13
  And I fill out the complete form # features/step_definitions/task_user_steps.rb:18
  Then I should see a complete status for the Task # features/step_definitions/task_user_steps.rb:23

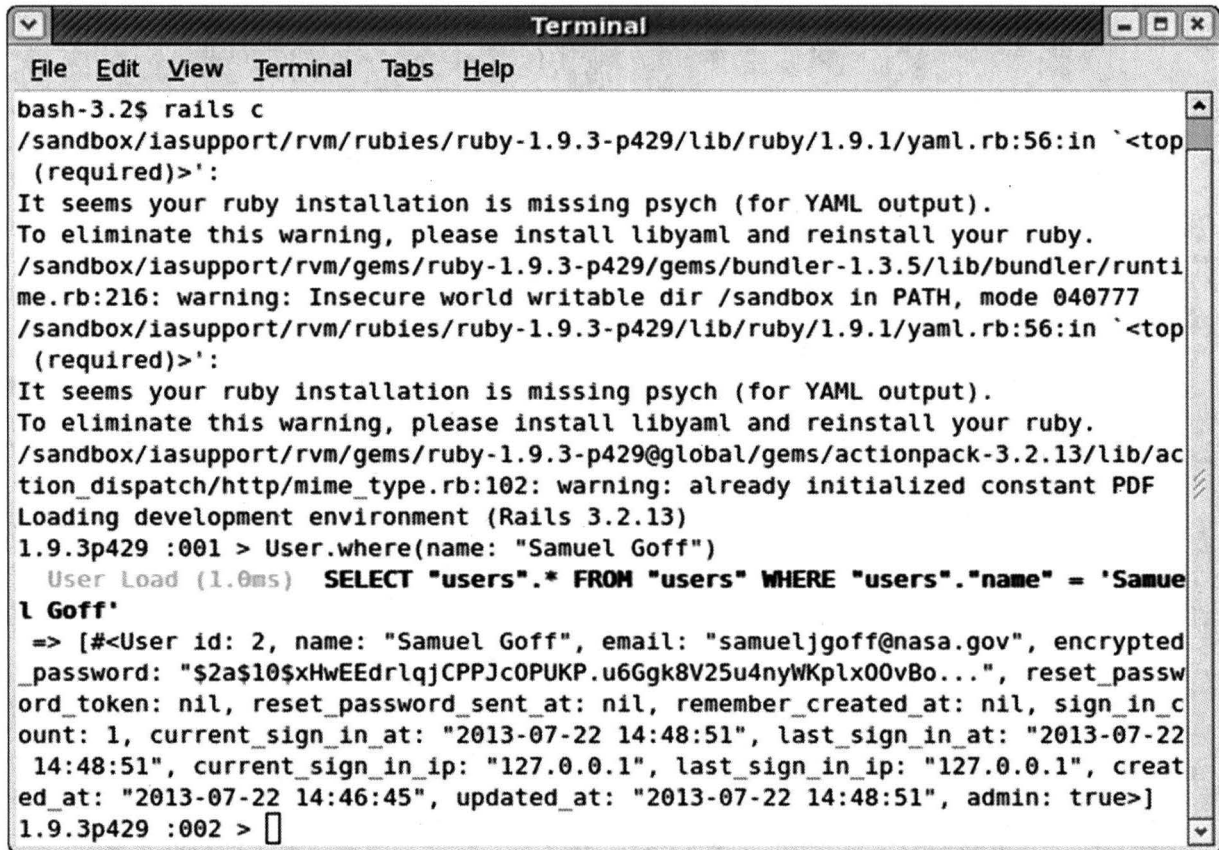
7 scenarios (7 passed)
65 steps (65 passed)
0m50.114s
bash-3.2$

```

Figure 4. The result of a successful Cucumber test.

application in plain English. Once the features are written out I can write methods in ruby that handle each of the statements in the feature. Once the steps and features are written out correctly I can run cucumber in the terminal by the command `$ bundle exec cucumber`. One of the important things cucumber does is handling errors every step of the way. If Cucumber finds an error it will display the entire error message and then skip to the next scenario. This is great for testing a new feature that may or may not interfere with other features in the application. Another advantage of Cucumber is that it runs tests on your features without interfering with your database or any of the data that you have. Cucumber uses the test database that Ruby on Rails provides.

Ruby on Rails has a lot of ways to debug an application as well as create new functionality. One of the ways to



```

Terminal
File Edit View Terminal Tabs Help
bash-3.2$ rails c
/sandbox/iasupport/rvm/rubies/ruby-1.9.3-p429/lib/ruby/1.9.1/yaml.rb:56:in `<top
(required)>':
It seems your ruby installation is missing psych (for YAML output).
To eliminate this warning, please install libyaml and reinstall your ruby.
/sandbox/iasupport/rvm/gems/ruby-1.9.3-p429/gems/bundler-1.3.5/lib/bundler/runti
me.rb:216: warning: Insecure world writable dir /sandbox in PATH, mode 040777
/sandbox/iasupport/rvm/rubies/ruby-1.9.3-p429/lib/ruby/1.9.1/yaml.rb:56:in `<top
(required)>':
It seems your ruby installation is missing psych (for YAML output).
To eliminate this warning, please install libyaml and reinstall your ruby.
/sandbox/iasupport/rvm/gems/ruby-1.9.3-p429@global/gems/actionpack-3.2.13/lib/ac
tion_dispatch/http/mime_type.rb:102: warning: already initialized constant PDF
Loading development environment (Rails 3.2.13)
1.9.3p429 :001 > User.where(name: "Samuel Goff")
  User Load (1.0ms)  SELECT "users".* FROM "users" WHERE "users"."name" = 'Samue
l Goff'
=> [#<User id: 2, name: "Samuel Goff", email: "samueljgoff@nasa.gov", encrypted
_password: "$2a$10$xHwEEdrLqjCPPJcOPUKP.u6Ggk8V25u4nyWKplX00vBo...", reset_passw
ord_token: nil, reset_password_sent_at: nil, remember_created_at: nil, sign_in_c
ount: 1, current_sign_in_at: "2013-07-22 14:48:51", last_sign_in_at: "2013-07-22
14:48:51", current_sign_in_ip: "127.0.0.1", last_sign_in_ip: "127.0.0.1", creat
ed_at: "2013-07-22 14:46:45", updated_at: "2013-07-22 14:48:51", admin: true>]
1.9.3p429 :002 >

```

Figure 5. Example of a rails console command.

accomplish this is the rails console. The rails console allows you to access and experiment with the data in the database in order to write correct code in the model and controller. Writing out snippets of code in the console can be very useful because you can see exactly what that line of code does before you implement it into your application. Also, the console can be used to directly check the database for information and to make sure that data is not missing, or nil in Ruby syntax.

Another important part of debugging is checking the routes of my application. Routes are the URLs that the views in my application refer to. For example, typing in localhost:3000/index in the browser will bring up the index page on my project. An easy way to keep track of all of the routes in a project is by entering the command `$ rake routes`. Rake routes displays an easy view of all of the routes as well as the controller action each route maps to. It also has the proper syntax for rerouting to that route within the views.

VII. Conclusion

All in all the Project Assessment Skills Web Application is a more intuitive way to keep track of training courses than a complex Excel spreadsheet. Users and product group leads will find that it is much easier to keep track of their training and their group's training progress. The application will greatly reduce time spend on finding and sending information to the appropriate people. Some further applications of my project are expanding it to include more than just NE-C but all of NE or even center wide.

Acknowledgments

The author thanks Andrew Davis for guiding him through creating this project and teaching him Ruby on Rails. The author would also like to thank Laurie Griffin and Julie Peacock for creating and supervising a conducive and productive work environment to learn and to improve. Lastly the author thanks the KSC Education Office for offering this fantastic opportunity to learn and to grow.

References

¹Wynne, M., and Hellesøy, A., *The Cucumber Book: Behavior-Driven Development for Testers and Developers*, 1st ed., The Pragmatic Bookshelf, Dallas, 2012, pp. 229-245.

²Bigg, R., Katz, Y., Klabnik, S., *Rails 4 in Action*, Manning Publications, New York, 2013, Chaps. 2, 5, 6.

³JetBrains, RubyMine, Software Package, Ver. 5.4, JetBrains, Berlin, NJ, 2013.