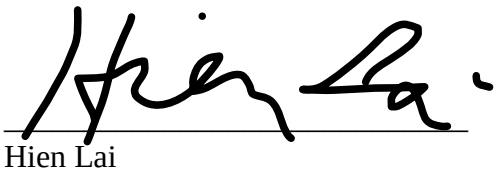
Mac VM Self Service Portal

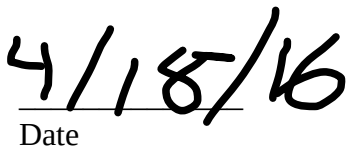
By

Hien Lai

Submitted to
The Faculty of the School of Information Technology
In Partial Fulfillment of the Requirements for
The Degree of Bachelor of Science
In Information Technology

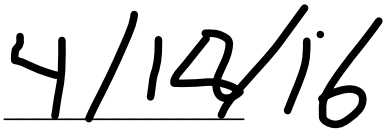


Hien Lai



Date

Jim Scott, Faculty Advisor



Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2016

Table of Contents

Abstract

The quality assurance process at Seapine Software requires the appropriate tools to efficiently deploy and manage virtual environments in order to smoothly facilitate testing of our software. The self-service portal for the creation and administering of OSX virtual machines was created to allow both quality assurance technicians and developers to quickly tap into available resources and deploy their own testing environments without the need of administrative intervention. This is a cost effective alternative to having physical Apple products for each user as it uses preexisting host machine assets and hypervisor software to give users access to the resources on a per needs basis. The created application will hopefully relax administrator duties and simplify the manner in which users deploy their OSX testing environments. This will also provide a platform for future growth and scaling in the QA department for the management of testing resources.

Introduction

Seapine Software, Inc. is based in Mason, Ohio and offers software development lifecycle products. One of the most crucial aspects of software development is the quality assurance step. Seapine Software understands that it is important that the QA process be as efficient and smooth as possible in order to thoroughly test the products in a timely manner. Because of this, virtual machines are the preferred method of quickly setting up the specific testing environments to reduce the cost of resources and ease of deployment. Having the ability to manage our own testing environments allows QA to utilize shared resources more efficiently. Hypervisors allow the entire department to create multiple virtual environments on only a few hosts rather than dedicated systems for each environment.

Project Description

At this moment, Seapine uses VMs for all major platforms of Windows, Linux, and Mac OS. However, only the Windows and Linux VMs are available through a self-service portal for QA to create their own VMs on demand. The OSX VMs are controlled by an administrator who receives tickets for VM related requests. When QA testers require an OSX VM, a ticket must be submitted to the administrator who fulfills the requests at his convenience. This middle man approach is both inconvenient and cumbersome as it requires the intervention of another individual which may slow down the QA process rather than it being an automated process in the form of a self-service client. The problem is that the VM administrator is also a full time developer. It is simple enough of a task to fulfill VM requests but in the situation that his attention is occupied by a meeting or a client support case, there will be a period of time where QA testers are without the resources needed.

This project involved creating a web application that will give users the ability to manage their own VMs without the need for administrative intervention. The web client will authenticate users through our Lightweight Directory Access Protocol (LDAP) server and communicate with the host machine running VMware Fusion using CLI commands via a Secure Shell (SSH) connection.

User Profile

Application: OSX VM Self Service Portal
Potential Users: QA testers and Developers
Software and Interface Experience: Users should be familiar with OSX and/or Linux operating systems and file system. Primarily will be working through the GUI for testing but will occasionally use the terminal for more administrative actions.
Experience with Similar Applications: Users will already have experience with VM portals through the system center for accessing VMs hosted on Hyper-V machines for Windows/Linux operating systems. The general layout and operations will be similar so users will already have experience in managing their own VMs.
Task Experience: Users will already have experience with VM portals through the system center for accessing VMs hosted on Hyper-V machines for Windows/Linux operating systems. The general layout and operations will be similar so users will already have experience in managing their own VMs.
Frequency of User: On a weekly basis, depending on frequency of new builds and necessity for creating new testing environments. Generally speaking, a new "clean" VM should be created or snapshot should be rolled back to whenever a new build of our software is released for testing.
Key Interface Design Requirements that the Profile Suggests: Login page that authenticates using users' LDAP credentials Listing of user's currently deployed VMs Menu for creating new VMs with a drop down or list of available templates Submenu for managing VMs that includes actions such as starting, shutting down, restarting, deleting, snapshot creation, and loading of snapshots.

Fig. 1 – User Profile

The VM administrator will have full control of the host machine either through physical access or via a client connection such as VNC. Users such as developers and QA will access only certain controls of the hypervisor through the web client application and given relevant access. If additional resources are required for testing such as more memory or processing power, a ticket should be submitted to the VM administrator who will access the host machine directly.

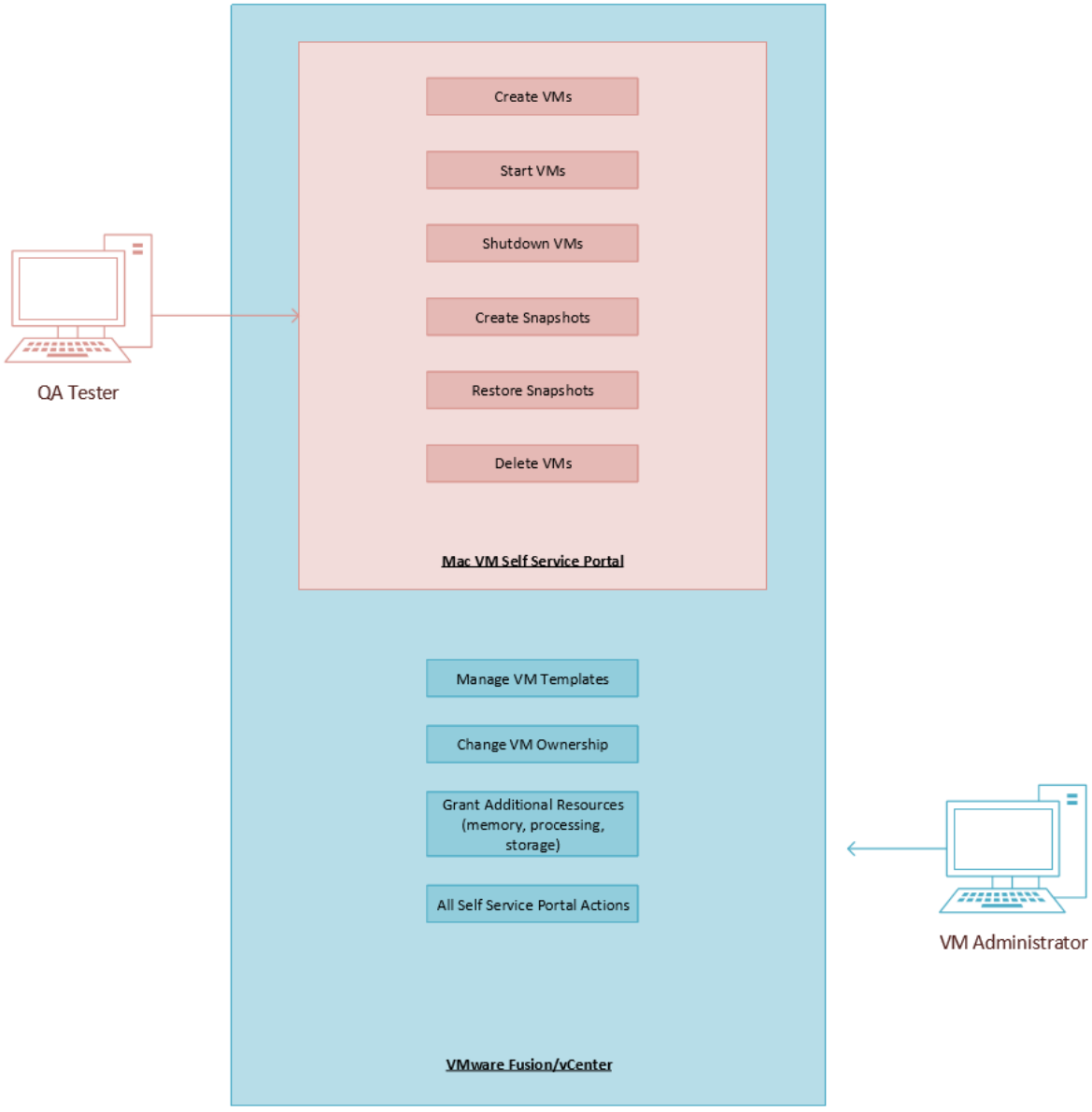


Fig. 2 – Use Case Diagram

Proposed Budget

The proposed budget will include labor of development, the software licenses for both the hypervisor and integrated development environment license, and also the hardware for the host machine. All of this is provided already by Seapine so the final calculated budget will be a theoretical amount for a company that has none of the resources available.

Resource	Projected Cost
Labor	$45 (wage) x 100 (estimated time) = $4500
VMware Fusion Pro License	$200
Mac Pro host machine	$8347
LCD Monitor	$200
IntelliJ IDEA Ultimate License	$500

Fig. 3 – Budget

Technical Elements

Software – The software used for the hypervisor will be VMware Fusion Pro. This is the VMware's solution to providing hypervisor capabilities to Intel-based Macs. We are not running OSX Server on the host machine but rather just OSX and the VMs will have templates for all OSX versions from 10.7 to 10.11.

Hardware – The host machine will be a top of the line Mac Pro with specs that allow for multiple VMs to run simultaneously within acceptable limits for the size of Seapine's QA department. It includes 12 cores and 64 gigabytes of RAM, along with a one terabyte hard drive and should be sufficient to accommodate testing demands without running out of resources.

Infrastructure – The framework that the application will be coded in is an open source Java based platform known as Grails which uses a language called Groovy. LDAP/active directory plugins are available for applications to integrate with those servers. The web client will be relaying CLI commands to the host machine via SSH that will execute commands in a utility called vmrun to operate the hypervisor.

Network – The portal will be hosted on a web server or another server that will ensure almost 100% uptime. Users will connect to the portal from their desktops or laptops which will give them access to the host machines configured for the portal. Each host machine can have multiple templates defined for them which will allow users to deploy VMs based off of those templates. VMs are stored on the host machine that contains the parent template.

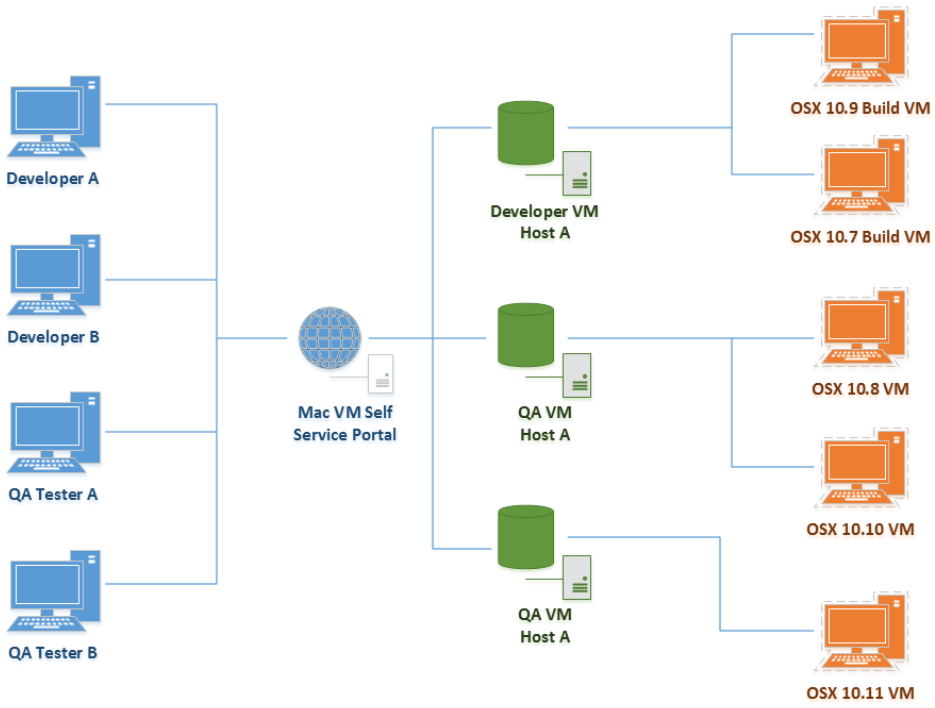
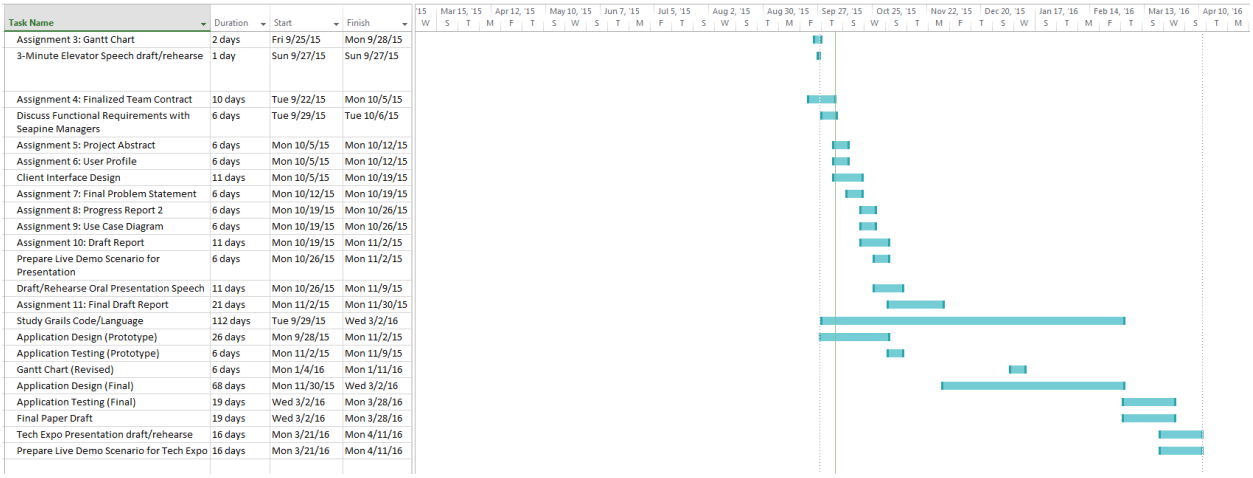


Fig 4 – Network Diagram

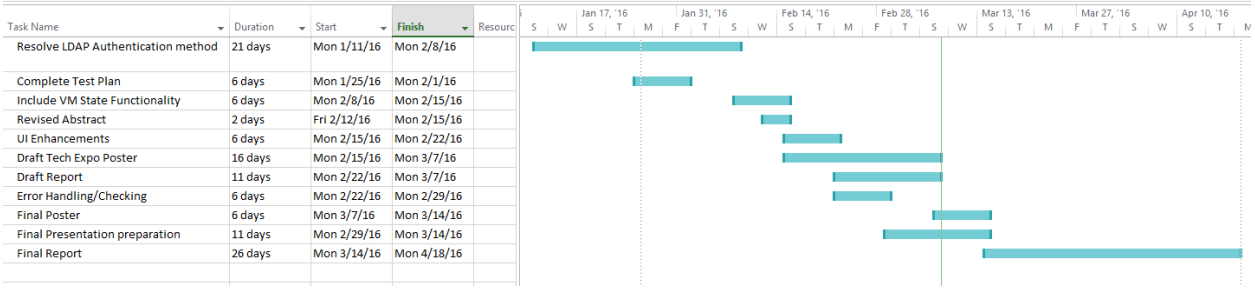
Timeline

These Gantt charts show projected deliverables and milestones for both fall and spring semester.



Task Name	Duration	Start	Finish
Assignment 3: Gantt Chart	2 days	Fri 9/25/15	Mon 9/28/15
3-Minute Elevator Speech draft/rehearse	1 day	Sun 9/27/15	Sun 9/27/15
Assignment 4: Finalized Team Contract	10 days	Tue 9/22/15	Mon 10/5/15
Discuss Functional Requirements with Seapine Managers	6 days	Tue 9/29/15	Tue 10/6/15
Assignment 5: Project Abstract	6 days	Mon 10/5/15	Mon 10/12/15
Assignment 6: User Profile	6 days	Mon 10/5/15	Mon 10/12/15
Client Interface Design	11 days	Mon 10/5/15	Mon 10/19/15
Assignment 7: Final Problem Statement	6 days	Mon 10/12/15	Mon 10/19/15
Assignment 8: Progress Report 2	6 days	Mon 10/19/15	Mon 10/26/15
Assignment 9: Use Case Diagram	6 days	Mon 10/19/15	Mon 10/26/15
Assignment 10: Draft Report	11 days	Mon 10/19/15	Mon 11/2/15
Prepare Live Demo Scenario for Presentation	6 days	Mon 10/26/15	Mon 11/2/15
Draft/Rehearse Oral Presentation Speech	11 days	Mon 10/26/15	Mon 11/9/15
Assignment 11: Final Draft Report	21 days	Mon 11/2/15	Mon 11/30/15
Study Grails Code/Language	112 days	Tue 9/29/15	Wed 3/2/16
Application Design (Prototype)	26 days	Mon 9/28/15	Mon 11/2/15
Application Testing (Prototype)	6 days	Mon 11/2/15	Mon 11/9/15
Gantt Chart (Revised)	6 days	Mon 1/4/16	Mon 1/11/16
Application Design (Final)	68 days	Mon 11/30/15	Wed 3/2/16
Application Testing (Final)	19 days	Wed 3/2/16	Mon 3/28/16
Final Paper Draft	19 days	Wed 3/2/16	Mon 3/28/16
Tech Expo Presentation draft/rehearse	16 days	Mon 3/21/16	Mon 4/11/16
Prepare Live Demo Scenario for Tech Expo	16 days	Mon 3/21/16	Mon 4/11/16

Fig. 5 – Gantt Chart Fall Semester



Task Name	Duration	Start	Finish	Resource
Resolve LDAP Authentication method	21 days	Mon 1/11/16	Mon 2/8/16	
Complete Test Plan	6 days	Mon 1/25/16	Mon 2/1/16	
Include VM State Functionality	6 days	Mon 2/8/16	Mon 2/15/16	
Revised Abstract	2 days	Fri 2/12/16	Mon 2/15/16	
UI Enhancements	6 days	Mon 2/15/16	Mon 2/22/16	
Draft Tech Expo Poster	16 days	Mon 2/15/16	Mon 3/7/16	
Draft Report	11 days	Mon 2/22/16	Mon 3/7/16	
Error Handling/Checking	6 days	Mon 2/22/16	Mon 2/29/16	
Final Poster	6 days	Mon 3/7/16	Mon 3/14/16	
Final Presentation preparation	11 days	Mon 2/29/16	Mon 3/14/16	
Final Report	26 days	Mon 3/14/16	Mon 4/18/16	

Fig. 6 – Gantt Chart Spring Semester

Test Plan

Testers may include both QA technicians as well as the VM administrator. Thorough testing of all specific functionality should be performed as well as ad-hoc testing to discover defects. Each test will consist of relevant pre-conditions and scope as well as specific steps to arrive at an expected result. Results will be recorded in a table and be given a Pass/Fail scenario based on meeting the criteria of the scope.

Functional Requirements:

1. All LDAP users will be able to authenticate into the portal.
2. Users will be granted general User level authority with given permissions/functionality.
3. Administrator will use admin specific user credentials for full functionality.
4. Users should be able to deploy and delete VMs based on templates defined by Administrator.
5. Users should be able to perform all VM based management such as Stop, Start, Suspend, Resume, Delete and State commands.
6. Users should only be able to manage and view their own VMs; Administrators can manage/view all VMs.

Test Cases of Functional Requirements

1. LDAP Authentication
    a. Password field characters are obscured
    b. NonLDAP users outside of the predefined Administrator account have no authorization
    c. Feedback on erroneous credentials should be provided.
2. Users granted specific authority and permissions.

a. User roles can only access the VM portion of the application.

b. Administrators have access to Host and Template management along with VM management. Administrators can also manage user accounts outside of LDAP.

3. Users should be able to manage all aspects of VM deployment.

    a. VMs can be deployed based on templates.

    b. VMs can be Started, Stopped, Suspended, Resumed, Paused, Unpaused, and all State commands such as creating/deleting states and listing/reverting to states.

4. Authority to View/Manage VMs.

    a. Users can only view and manage their own specific VMs. They have no access to VMs deployed by other users.

    b. Administrators can view and manage all VMs deployed.

LDAP Authentication Tests			
Scope: Password field characters are obscured			
Step #	Action	Expected Result	Pass/Fail
1	Enter alphanumeric characters into password	Password obscured	Pass
2	Enter nonalphanumeric characters into password	Password obscured	Pass
3	Paste characters into password	Password obscured	Pass
4	Copy characters from password field	Cannot copy	Pass
Passed			
Scope: Users may only login if provided credentials match an LDAP user			
Step #	Action	Expected Result	Pass/Fail
1	Enter in wrong username and password	Error message	Pass
2	Enter in correct username and wrong password	Error message	Pass
3	Enter in no username and password	Error message	Pass
4	Enter in correct username and password	Portal access	Pass
Passed			

Fig. 7 – LDAP Test Results

Granted Permissions Tests			
Scope: Admin user has access no restrictions for portal management options			
Step #	Action	Expected Result	Pass/Fail
1	Log in as administrator	Log in successful	Pass
2	Verify existence of admin options	Options exist	Pass
3	Click on Host Management	Host Management displays	Pass
4	Click on New Host Machine	Add Host Machine displays	Pass
5	Click on Host Machine List	Return to Host Management	Pass
6	Click on Template Management	VM Template List displays	Pass
7	Click on New VM Template	Add VM Template displays	Pass
8	Click on VM Template List	Return to VM Template List	Pass
Passed			
Scope: LDAP/regular users only have access to VM management			
Step #	Action	Expected Result	Pass/Fail
1	Log in as regular user	Log in successful	Pass
2	Verify only VM Management available	No other options	Pass
3	Click on VM Management	VM List displays	Pass
4	Click on New VM	Create VM displays	Pass
5	Click on VM List	Return to VM List	Pass
Passed			

Fig. 8 – Permissions Test Results

VM Management Tests			
Scope: Users are able to use all VM functionality			
Step #	Action	Expected Result	Pass/Fail
1	Create VM from template	VM created	Pass
2	Click on Start button	State returns running	Pass
3	Click on Pause button	State returns paused	Pass
4	Click on Unpause button	State returns running	Pass
5	Click on Suspend button	State returns suspended	Pass
6	Click on Resume button	State returns running	Pass
7	Click on Create Snapshot button	State returns creating	Pass
8	Click on Stop button	State returns stopped	Pass
9	Click on Revert button	VM snapshot is reverted	Pass
10	Click on Delete button	VM is deleted	Pass
Passed			

Fig. 9 – VM Management Test Results

VM Authority Tests			
Scope: Admin can manage all VMs, Users can only view/manage their own VMs			
Step #	Action	Expected Result	Pass/Fail
1	Create VM as a User	VM created	Pass
2	Create VM as Admin	VM created	Pass
3	Check VM list as User	Only have access to User VM	Pass
4	Check VM list as Admin	Have access to both VMs	Pass
Passed			

Fig. 10 – VM Authority Test Results

Lessons Learned

Being a networking student with limited knowledge of coding, I vastly underestimated the commitment involved in learning enough of the Groovy/Java language to begin implementing the features of this project. I am lucky enough to have a base to work off of and in house experts to come to for advice but in hindsight, I should have formed a partnership with a software development track student for an easier time implementing the features and perhaps adding additional features. I originally wanted to have a pop out console for the VMs within the web client as a means to quickly access them from the browser similar to how Seapine's system center for our Hyper-V hosted VMs work but I am not optimistic about that prospect anymore. Connections to the Mac VMs may have to go through a VNC client like we currently use such as VNC Viewer or Tight VNC.

Conclusion

This application will prove to be a valuable resource to Seapine when introduced into the development environment. It will introduce an automated system to allow both developers and QA testers to deploy Mac VMs on demand without the need for the additional overhead of processing VM requests through the ticketing system. This smoother process for setting up virtual environments is not only something I will use, but is also what many co-workers have been anticipating. It will also be scalable in the future when the company grows and the departments get larger by simply configuring the application for additional resources and allowing users to tap into those resources.

Bibliography

Beckwith, Burt. n.d. *Grails Spring Security LDAP Plugin.* Accessed November 10, 2015. http://grails-plugins.github.io/grails-spring-security-ldap/.

Shapochnik, Yan, and Vincent, Paul. Interview by Hien Lai. 2015. Personal interview (September 17).

"Documentation." *The Groovy Programming Language – Documentation*. Accessed October 2015. http://www.groovy-lang.org/documentation.html.

Using vmrun To Control Virtual Machines. 2009. Ebook. 1st ed. VMware, Inc. http://www.vmware.com/pdf/vix180_vmrun_command.pdf.