Team Six

SQS Training Website

11/14/2016

CS 499, Fall 2016

Authors

- Josh DeLong
- Nick Schorr
- Stephen Williams
- Nathan Hunter

SQS Contacts

- Luke Robinson
- Jun Yagi

Website

- cs4996.wordpress.com

Table of Contents

Source Code

The source code for this project can be found at: https://github.com/jdelong747/CS499

Quality Review

Our team spent some time looking over the code, and found several issues, however due to the nature of the the site, many of these flaws were intentional. For example, all of the SQL is done in a (purposefully) unsafe manner, which would never show up in actual production code. Since code quality is important to us, and our client, we have each person review the branches to be merged, prior to merging. This lets us ensure that committed code is functional, and makes sense within the context of our project.

User Manual

Introduction

SQS is a leading specialist in software quality assurance, and is always looking to maintain their edge. This means that new hires need to both be intelligent, and have the right mindset for finding and solving the software quality issues that SQS' clients have been troubled by. To help with preparing their new hires, we've developed a web application that they can visit, which will present them with situations that they may find out in the field. Whether it's as simple as a brute force attack, or as devious as SQL injection, this application can help give them them track down these issues in a timely manner. Below, is a guide to using the site. It contains an explanation of every feature on the site, as well as a visual guide to using it.

Home Page

Anyone who visits the training site will be greeted by the home page. This page exists as a landing to tell the user a bit more about what the site does, and to allow anyone to subscribe the the email newsletter put out by SQS. As you can see in Figure 1 below, the page itself is designed to be simple and clean, and allow the user to easily navigate it. On this home page, there are a couple of key features to be aware of. Namely, the green navigation bar located in the middle of the image, and the subscription button located at the bottom. Note: The lorem ipsum placeholder text is just there to fill out the page, and give an idea of what it will look like when SQS has taken delivery of the web application, and has customized the text to their needs. In this case, the text on the homepage will likely give an overview of SQS and the training they wish to do.

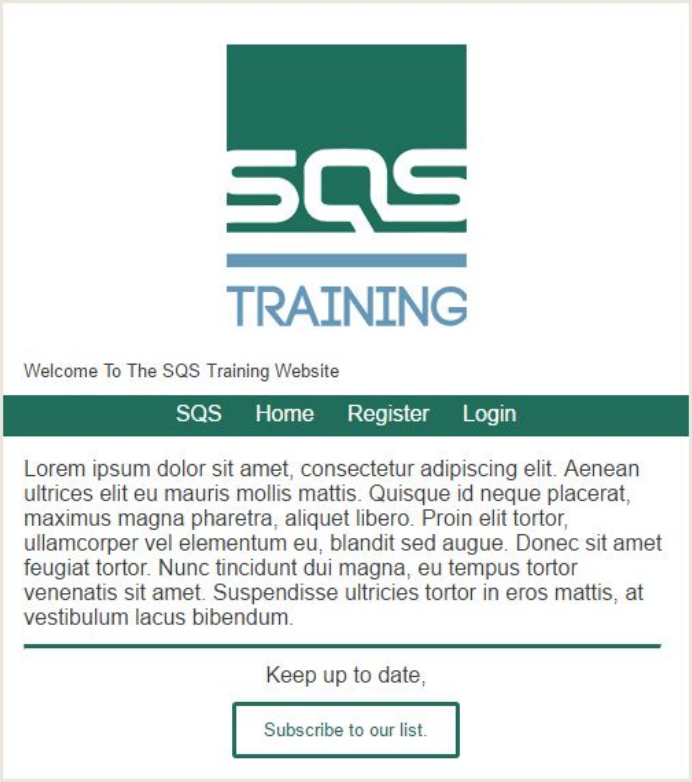

Figure 1: The home page.

Navigation Bar

The navigation bar is the simplest way to traverse the site, and you're likely already familiar with how it works. Basically, it will give you options on where to go based on conditions unique to your session. The navigation bar seen in Figure 1 is how anyone who isn't currently logged into the system will see it. Likewise, the navigation bar seen in Figure 6 is how anyone who is logged into the system will see it.

The navigation bar has five different places that it can link you to. These include a link back to SQS' main home page, a link back to this training site's home page, a link to the registration page of this site, a link to the login page of this site, and a link to log yourself out of the site. See Figure 2, Figure 3, Figure 4, Figure 5, and Figure 6 respectively to see the navigation bar in use. Otherwise, please skip the appropriate section below to see how the other parts of the site work.

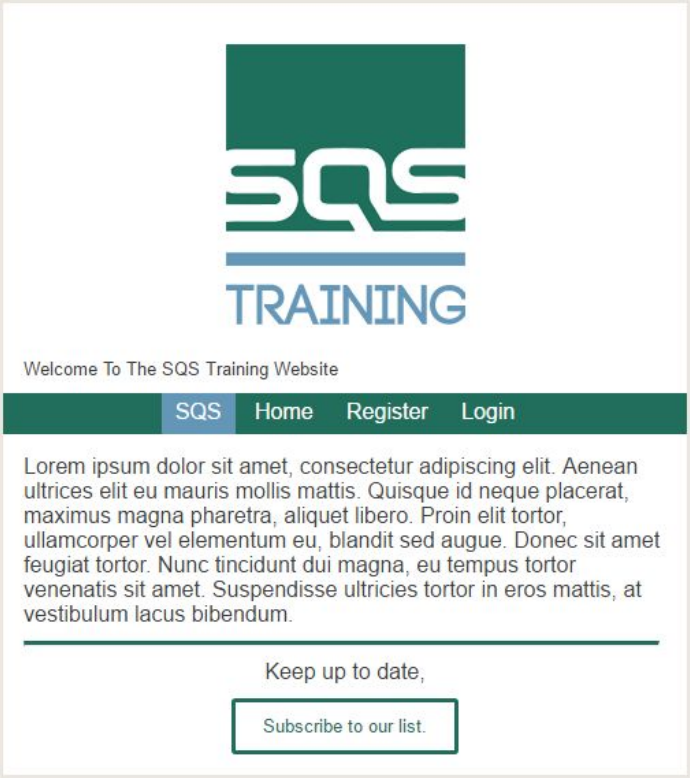

Figure 2: The home page with the SQS navigation bar link being hovered over.

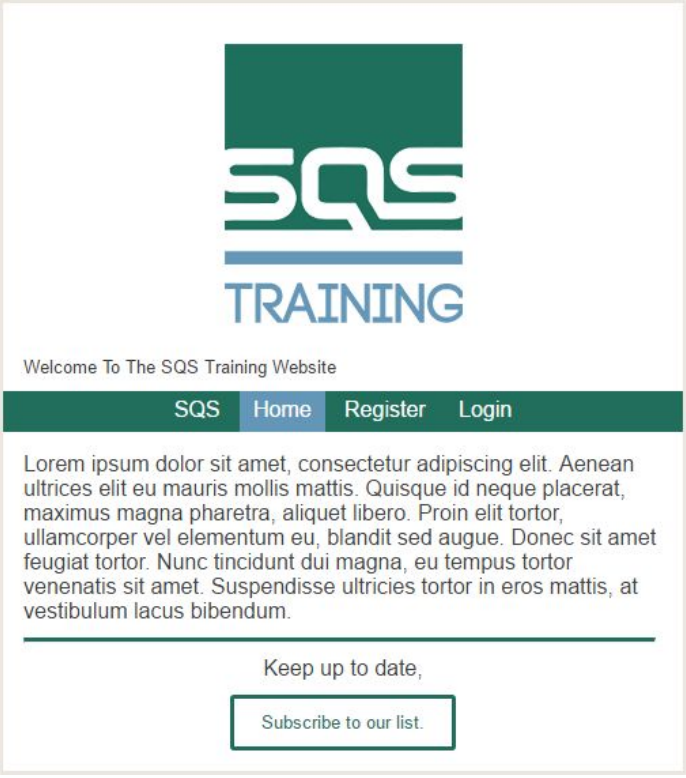

Figure 3: The home page with the Home navigation bar link being hovered over.

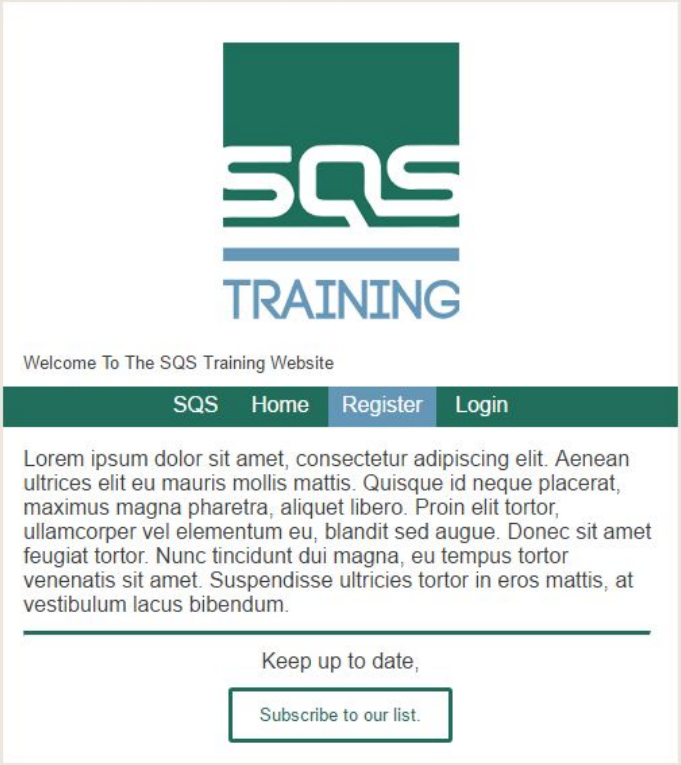

Figure 4: The home page with the Register navigation bar link being hovered over.

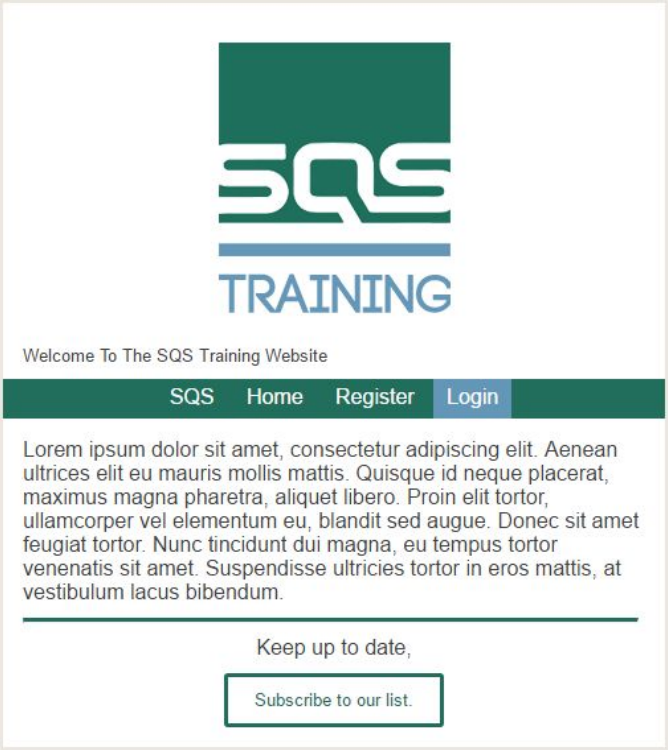

Figure 5: The home page with the Login navigation bar link being hovered over.

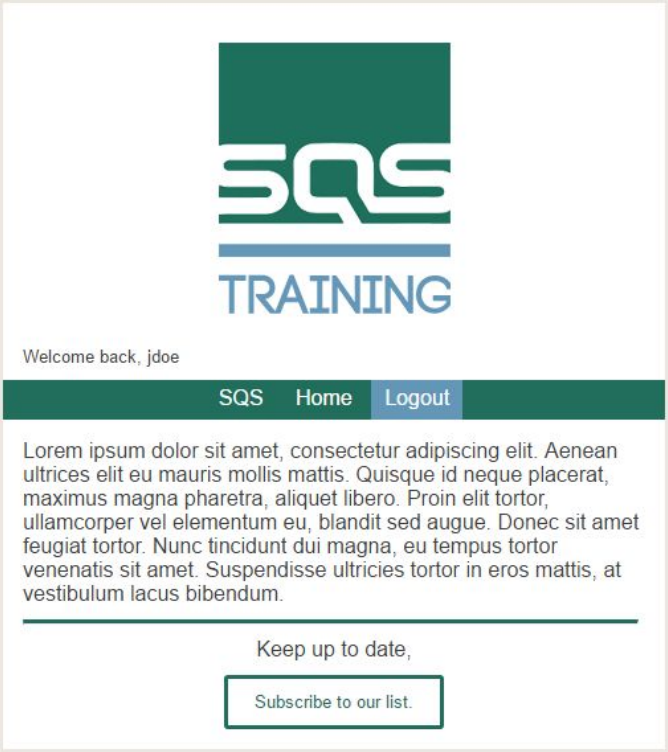

Figure 6: The homepage of a logged in user who is hovering over the Logout navigation bar link.

Subscription

As mentioned previously, users who are viewing this page also have the option of signing up to receive an email newsletter put out by SQS for this training site. The form for signing up for this letter can be reached by clicking on the big “Subscribe to our list.” button at the bottom of the page. Below, Figure 7 illustrates that process.

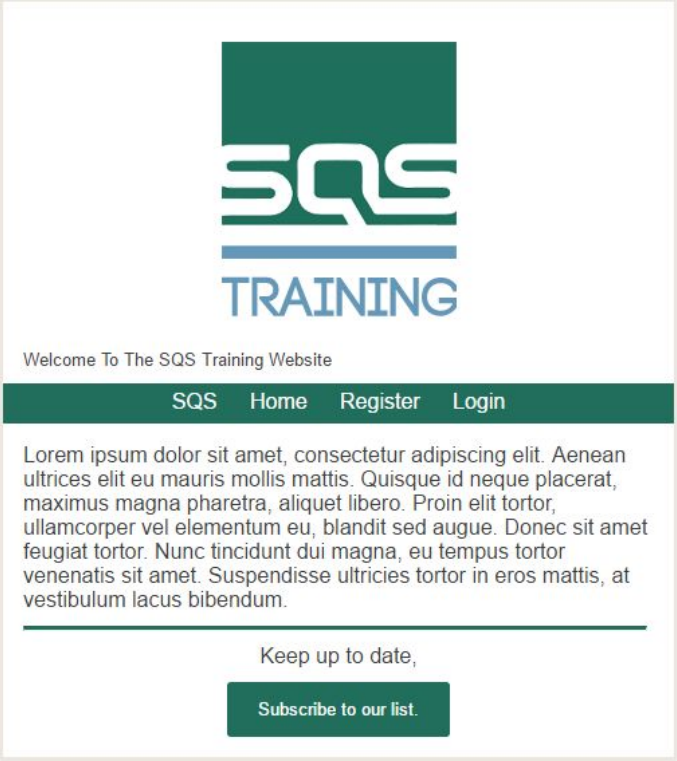

Figure 7: The home page with the “Subscribe to our list.” button being hovered over.

After clicking on the subscription button, you’ll be taken to a new page, seen in Figure 8, that allows you to sign up to receive the email newsletter. This can be done by entering the desired email address, and then clicking on the subscribe button on the bottom of the page. See Figure 9 for an example of this operation.

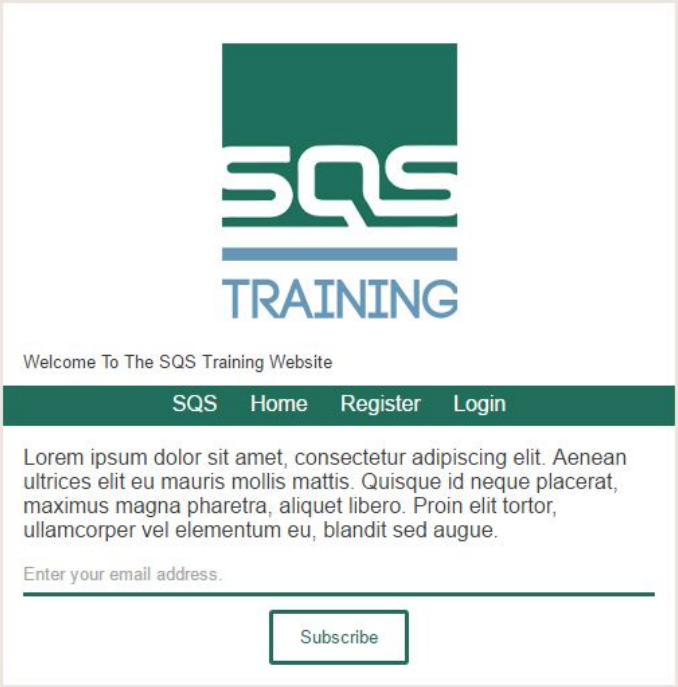

Figure 8: The subscription page.

Figure 9: The filled out subscription page

Registration

New users who wish to access the training site can register an account with the system. Below, Figure 10 shows the registration page of the site. Registration can be accomplished by simply entering in your desired credentials. For example, user John Doe, has registered to get an account below in Figure 11.

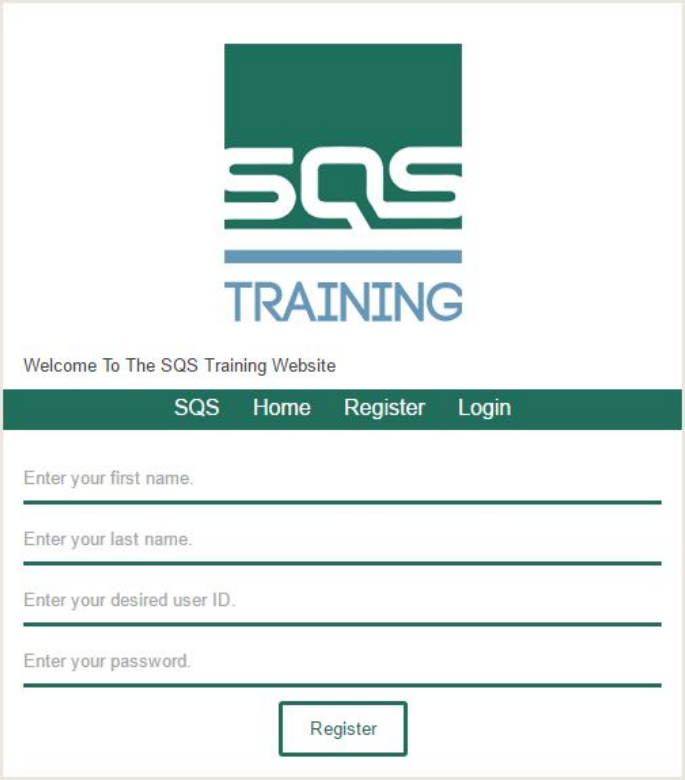

Figure 10: The registration page.

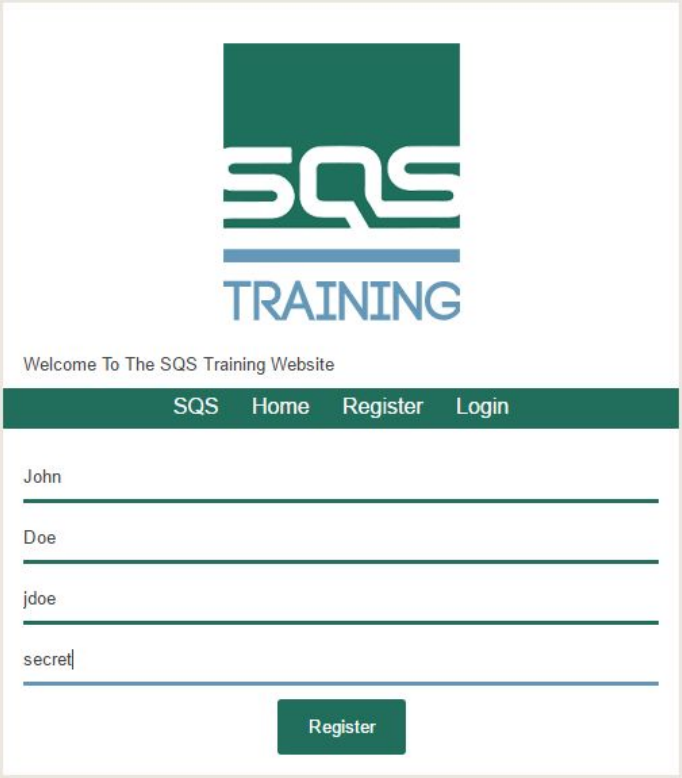

Figure 11: An example registration page filled out by user John Doe.

Login

Logging in can be done after registration has been completed. It simply requires the user to enter their user id, and password successfully before they can be logged into the system. Below, Figure 12 shows the login page, and Figure 13 illustrates the process of our user John Doe loggin in. Incorrect logins will be unsuccessful, and the user will be alerted that their login failed. They will then be able to try again. This can be seen below in Figure 14.

Figure 12: The login page

Figure 13: An example login page filled out by user John Doe (See registration section above).

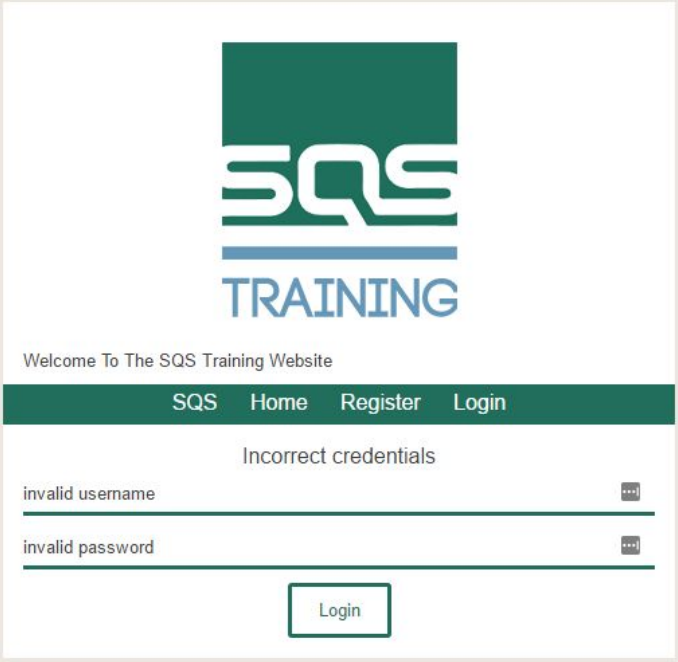

Figure 14: An example of what happens when the invalid credentials “invalid username” and “invalid password” are attempted to log in with.

Logout

Logging out can be performed by anyone who has already been logged in to the system. This can be done by simply clicking on the “Logout” link in the navigation bar (See Figure 6). You will then be redirected back to the original home page (See Figure 1).

Administrator Manual

This website is designed to be run with very little configuration and very little resource overhead. The current system is run locally on an embedded tomcat server, however development instances could be stood up in the future if that is deemed necessary. The primary tool that will be needed to run the website is the spring tool suite, which can be downloaded from spring's website. In addition to spring tool suite, an active directory will be required as well. For the purposes of building the website, a windows server 2012 R2 virtual machine was created and will be used as the source for the active directory. While this is the current system use, any active directory can be connected to by modifying the configuration for the application.

Spring tool suite will be the primary tool that is used for deploying and working with the website. The tool suite contains an IDE based on eclipse called STS, and different version of Tomcat that can be used to deploy the application. Once the latest version of spring tool suite has been downloaded, navigate to sts-bundle\sts-#.#.#.RELEASE\STS.exe to run the IDE. The spring IDE is a powerful development environment that provide a convenient way to configure and deploy the application.

Once STS is open, a version of the code will have to be obtained. The code is currently in a git repository which can be cloned directly in STS and imported easily. In the top right of STS click open new perspective and open the git perspective. The git perspective is an easy way to manage git repositories directly from STS. On the left there is a Git Repositories view, click the clone git repository button to open a dialog to clone the repository. You can clone either via GitHub or using a URL to access the git repository. Once the repository is cloned it will appear in the Git Repository view. Right click on the repository and select the last option Import Maven Projects. This will detect the pom.xml file in the project and automatically add the project to your workspace.

Now that the code is cloned, a tomcat server will need to be configured to run the website on. Return to the spring view by clicking the spring logo in the top right corner. You should see a tab called Servers in the bottom left. Right click in the tab and select New > Server. This will bring up a dialog box that allows you to configure a server. Click on the Pivotal folder and select Pivotal tc Server v3.0 - v3.1. You can change the host and server name if needed, and the default runtime can be used. Click next and check Existing Instance and in the dropdown box select base-instance. Finally click Finish to add the server to your servers tab.

With the project imported and the server configured, you are now ready to run the application. Right click on the server in the server tab and select Run to start the server. With the server running, simply drag the project onto the server to deploy it. Alternately you may right click the project, select Run As > Run On Server, and select your server from the list. This will deploy the application which can now be accessed by going to localhost:8080/training.

There are multiple aspects of the application that can be further configured to suit future needs. One of which is the database that the application is using. Currently the website is running an HSQLDB in memory database. This means that whenever the application is deployed, a new database is created in memory that is destroyed when the application is undeployed. There are two sql scripts entitled hsqldb-data.sql and hsqldb-schema.sql located in the src/main/resources/sql folder of the project. These two scripts are run to create the schema and enter test data in the database. If a new table or field needs to be added or more test data needs to be created, simply modifying these scripts will implement the changes the next time you deploy

the application. The in memory database is not tied with the functionality of the application, and could be replaced with a single instance of a database by simply modifying the connection information.

Another configurable aspect of the application is the active directory. As previously mentioned, for the development of the website a virtual machine running Windows Server 2012 R2 was used to serve as an active directory server. This virtual machine will not be maintained after the project is complete and the code is delivered, so a new active directory server will need to be created, or an existing active directory will need to be connected to. This can be done by changing the connection information to the active directory, as well as changing any groups to reflect the organization of the new active directory.

Testing

Unit Testing

The lowest level unit testing for the SQS Training website was divided into two categories: non-logged-in user and logged-in user. All of the tests were performed manually. **Table 2.1.1** lists all of the unit test cases, as well as their passing status.

#	Test Case	Pass Condition	Fail Condition	Status
1	Click on “Register” button on the homepage	User is redirected to the registration page	User is not redirected **or** is redirected to the wrong page	Pass
2a	User attempts registration with valid email/password	System registers user **and** redirects to homepage	System does not register user **or** does not redirect to home	Pass
2b	User attempts registration with invalid email/password	System does not register user **and** tells the user the reason they were not registered	User is registered with invalid login information **or** is not told why attempted registration is invalid	Pass
3	User clicks “Login” button on the homepage	User is redirected to the login page	User is not redirected **or** is redirected to the wrong page	Pass
4a	User attempts login with registered information	Session created **and** user alerted of successful login **and** user redirected to homepage	Session not created **or** user not alerted **or** user not redirected	Pass
4b	User attempts login with non-registered information	Notify user login information is not in Active Directory **and** allow another attempt	No notification of invalid information or session is created **or** user is given access to the site	Pass
5	User clicks “Logout” button	User’s session is killed **and** user is redirected to homepage	Session is not deleted **or** user is not redirected	Pass
7a	Non-logged in User clicks “Subscribe” button	User is redirected to subscription page	User is not redirected, or is redirected to the wrong page	Pass
7b	Non-logged-in user that is not subscribed enters email to subscribe	Adds email to subscription list	Email not added	Pass
7c	Non-logged-in user that is	User is notified that email is	User is not notified	Pass

	already subscribed enters email to subscribe	already in the system		
7d	Logged in user that is not subscribed clicks “Subscribe” button	Adds email from current session to subscribed database	Email is not added	Pass
7e	Logged in user that is already subscribed clicks “Subscribe” button	User notified they are already subscribed **and** no email added	User is not notified	Pass
8a	User clicks the “Unsubscribe” button	User is redirected to the unsubscribe page **and** “Confirm” button appears	User is not redirected **or** “Confirm” button does not appear	Pass
8b	User clicks the “Confirm” button	Email is removed from subscription database, user notified of success, redirects to homepage	User is not removed, notified **or** redirected home	Pass
9a	Non-logged-in user goes to homepage	“Login” “Register” and “Subscribe” buttons are shown with the homepage	Any of the three buttons are missing **or** an incorrect button appears	Pass
9b	Logged in user goes to homepage	“Logout” and “Subscribe” buttons are shown	Either of the buttons are missing **or** an incorrect button is shown	Pass
10	Logged in user navigates to training section	Training section is displayed	Training section is not displayed	Pass
11	Logged in user navigates to a specific training page	Requested page is displayed	Page is not displayed	Pass

Table 2.1.1

Acceptance Testing

At a much higher level, acceptance tests were performed on the website based on the requirements specified in the Requirements Specification document. Once again, these requirements were split up into two groups: non-logged-in users and logged-in users. **Table 2.2.1** lists the specifications for non-logged-in users and **Table 2.2.2** lists the specifications for logged-in users. Both tables also display the status of each test.

#	**Requirement**	**Verification**	**Method**	**Status**
1	The system shall allow users to log in to existing accounts on the Active Directory.	Log in with a previously created account.	Manual	Pass
2	The system shall allow users to register new accounts on the Active Directory.	Create an account, get confirmation from the system.	Manual	Pass

3	The system shall allow users to browse the training site's homepage section.	Click all links on the homepage and verify that the appropriate responses are given by the system.	Manual	Pass
4	The system shall allow users to sign up for a subscription letter, with their email address.	Enter email address and click the subscribe button on the Subscribe page, get confirmation from the system.	Manual	Pass
5	The system shall not allow users to log out.	Ensure that before logging in, the "Logout" button is not visible.	Manual/ By Design	Pass
6	The system shall not allow users to participate in the training process.	Attempt to access training content before logging in.	Manual	Pass

Table 2.2.1 - Acceptance Testing for Non-Logged-In users

#	**Requirement**	**Verification**	**Method**	**Status**
1	The system shall allow users to log out of the system.	Click the login button and verify the website reverts to a non-logged-in user view.	Manual	Pass
2	The system shall allow users to sign up for a subscription letter, using their existing email address on the Active Directory.	Click the subscribe button while logged in. User need not enter an email address. Get confirmation from system that the email address has been added to the list.	Manual	Pass
3	The system shall allow users to browse the homepage, and training sections of the site.	Verify that a "Login" tab is not present on the homepage. Verify that the training content is presented when clicked.	Manual/ By design	Pass
4	The system shall not allow users to log in, or register accounts.	Verify that neither the "Login" or "Register" tabs are visible on the home page.	Manual/ By design	Pass
5	The system shall provide a link to the training section of the training site.	Verify that the "Training" link appears on the homepage when logged in.	Manual/ By design	Pass
6	The system shall allow users to participate in the training process.	Verify that the training content is given by the system when the "Training" button is clicked.	Manual	Pass

Table 2.2.2 - Acceptance Testing for Logged-In Users

Metrics

The size of the project was estimated as eight user stories (User Registration, User Login, User Logout, User Subscribe to Letter, User Unsubscribe to Letter, User Views Homepage, User Views Training Section, Users Views Training Page), x number of test cases (), and 12 methods (getID(), setID(Integer), getEmail(), setEmail(String), getFirstName(), setFirstName(String), getLastName(), setLastName(String), getUserID(), setUserID(String), getPassword(), setPassword(String)). No method was estimated to take significant time to code since there are no complex algorithms and consist mainly of transferring data to and from systems.

There were approximately 400 lines of code used for the current build of the project with four controller Classes each having two methods operating as user stories, as well as six separate .jsp files used for the interface of the webpage.

To estimate the complexity of the eight methods, the Cyclomatic Complexity model will be used working from the formula Cyclomatic complexity = E - N + P where E is the number of branches in the flow graph, N is the number of nodes, and P is the number of exit nodes.

Authentication Controller: User Login	E	N	P	CComp.
	4	3	2	3

Authentication Controller: User Logout	E	N	P	CComp.
	2	1	1	2

Register Controller: RegisterUser	E	N	P	CComp.
	5	4	2	3

EmailController: Subscribe	E	N	P	CComp.
	2	1	1	2

EmailController: Unsubscribe	E	N	P	CComp.
	4	3	1	2

DefaultController: User Views Homepage	E	N	P	CComp.
	1	1	1	1

DefaultController: User Views Training Page	E	N	P	CComp.
	1	1	1	1

DefaultController: User Views Training Section	E	N	P	CComp.
	1	1	1	1

DefaultController: User Views Training Page	E	N	P	CComp.
	1	1	1	1

For most of our methods, there were not very many multiple lines of logic to maintain as it was generally data flow with try and catch conditions. Some of the simpler methods just had one possible route of logic to follow.

For the complexity of the overall project we will use the $CBO = \frac{NumberOfLinks}{NumberOfClasses}$ formula where CBO stands for Coupling Between Object Classes. For our particular project we have $CBO = \frac{2}{8} = .25\ CBO.$

The final number of user stories presented and communicated with the customer is twelve, and over the twelve user stories we have approximately 35 tests applied.

Total number of hours put into this assignment as of 11/14/2016 is:

Josh DeLong: 12 hrs

Nathan Hunter: 5 hrs

Nick Schorr: 7 hrs
Stephen Williams: 5 hrs
(total: 29 hrs)

Website and Developer Notebooks

The website is located at cs4996.wordpress.com. This site also contains links to our developer notebooks for the various assignments.