ENG 3050 – Software Engineering Design I

Winter 2015

Artificial Intelligence Literature Searching

Assignment 4 – Software Testing

Group #3

Contributing Members:

John Simko, Brandon Stanley, Shahood Mirza

Date Submitted: 3/9/2015

Approval

This document has been read and approved by the following team members responsible for its implementation:

Print Name	Signature	Comments
John Simko		

Print Name	Signature	Comments
Brandon Stanley		

Print Name	Signature	Comments
Shahood Mirza		

Revision History

Date	Author	Comments
March 7	Brandon Stanley	Created and laid basic framework of document. Wrote the Introduction section (System overview and Test approach)
March 8	Brandon Stanley	Added to the test plan section
March 12	Brandon Stanley	GUI Test cases written and completed. Added to test plan section
March 13	John Simko	Added suggested Integration and Server Back-End test cases
March 15	Shahood Mirza	Added some integration test cases
March 15	John Simko	Completed several Server Back-End test cases
March 16	Shahood Mirza	Added Client-Side test cases. Updated integration test cases
March 16	John Simko	Completed more Server Back-end test cases, fixed formatting, Completed several Integration test cases.

Table of Contents

1. Introduction

1.1 System overview

This project is a client server system. A client system will be used by all user and admins to gain access to a server system. In most cases, this server system is a remote system, but the server will also have an option admin only login in screen.

The server will be a database of Artificial Intelligence literature of various types, including but not limited to: standards, journals, articles, textbooks, and research papers.

To gain access to the server, a user or admin must login to the system through their client system. Their account type (admin or user) will determine their permissions with the system.

Both account types (user and admin) are capable of searching through the database, sorting and/or filtering any results, and accessing a tutorial. Additionally, an admin is capable of some further functionality which consists of: adding and removing users, adding and removing attributes, and finally adding, modifying and removing resources.

The server is capable of connecting several concurrent connections, each running on its own thread on the system.

1.2 Test approach

The testing approach that will be used throughout this document will be the black box testing method. The testing will consist of four sections of test cases. Of these four tests sections, three are Unit Tests, and the last is the Integration Tests.

The three Unit Tests covered in this document will be each of the member's software system. These are: the server, the client, and the GUI. Each test will be described in detail by that system's owner.

The Integration Tests are a test of the systems as a whole. These will be tests that cover the fully working system. This section will be the responsibility of all members of the project.

All tests will be peer reviewed. All test cases will be reviewed by other members, and may be modified with group consent. All project code, and test case code is to undergo a code review process where it is covered as a group.

2. Test Plan

2.1 Features to be Tested

2.1.1 The Server Back-End
2.1.2 The Client/Server connections
2.1.3 The GUI
2.1.4 System Integration Testing

2.2 Features not to be Tested

2.2.1 Network response time
The network is outside of our control. In theory the response time should be good, but there is no way to be sure of this, or to reliably test it.

2.2.2 Client system response time
Very hardware dependent, would need to test across many hardware platforms to reliably get an idea. In theory should be good, as this application is quite light weight.

2.3 Testing Tools and Environment

Our working environment will be Windows OS, and Visual Studios IDE. Our primary testing tools will be Visual Studios built in Unit Test feature, manual testing for some features, as well as Sikuli wherever possible.

3. Unit Test Cases

3.1 Server Back-End Test Cases

3.1.1 Search Database

Purpose

The purpose of this test is to ensure valid search criteria are returned. It must be able to handle searches with no results, and searches with one or more results, up to and including the entire resources database.

Inputs

A searchable field and a search string

Expected Outputs

A vector of resources, which is filled with results matching the given search string in the appropriate field.

Pass Criteria

The test will pass if all matching results are returned, with no mismatches included. In the case of no matches, returning an empty vector will be correct.

Test Procedure

Create a smaller resources vector of known values, and give search criteria that should result in zero, one, and all results.

3.1.2 Sort Searched Vector

Purpose

The purpose of this test is to take a pre-existing vector of resources with filterable keywords, and create a subset of the vector, which only contains resources that contain the given keyword. It must handle keywords that match all resources, one resource, or no resources.

Inputs

A single keyword in string format.

Expected Outputs

A second vector of resources, containing all, some, or none of the resources in the original vector.

Pass Criteria

A vector with all resources that contain the given keyword. In the case of no matches, an empty vector will be treated as a correct result.

Test Procedure

Using a small vector of resources with known values, create a stub client that sends filter request and keyword to match each case of: all resources, no resources, and some subset of resources.

3.1.3 Read in Resources from CSV

Purpose

The purpose of this test is to check that our server system will correctly read in our data from CSV format and store it in a resources vector.

Inputs

A CSV (comma-separated values) file formatted with title, author, publish date, abstract, and keywords, followed by any other fields added by system admins.

Expected Outputs

A vector of resources, where each resource matches to a row in the CSV.

Pass Criteria

Each row in the CSV is properly matched to a resource in the vector, with all data points stored in their correct fields.

Test Procedure

Create a smaller, manageable set of simpler, known data, and import this. Verify that each resource matches exactly to the given input.

3.1.4 Export Data to CSV

Purpose

This test is to verify that the server correctly exports its stored vector of resources to a CSV file on demand. It will be tested for both locally saving the CSV, as well as sending the CSV to an admin on a client program.

Inputs

A request for CSV export from an admin on client.

Expected Outputs

Depending on boolean flags, a CSV passed through the socket connection to client and/or an export of CSV to save over the previous CSV file.

Pass Criteria

A CSV which matches the currently stored resources in the Resources vector is sent to client and/or backed up on the server host, based on given flags.

Test Procedure

Create a vector of known values and set flag for modified values (recently added). Send export request.

3.1.5 Accept New Connection

Purpose

The purpose of this test is to verify that the correct login success/failure response is returned based on the given login name and password. It must test a failed login, a successful user login, and a successful admin login.

Inputs

A username and password string.

Expected Outputs

An integer representing failure [0], success (user) [1], or success (admin) [3].

Pass Criteria

The login function successfully checks username/password combinations and returns the appropriate integer value.

Test Procedure

Create a set of known logins and call the login function with matching information for an admin account, a user account, and mismatched information.

3.1.6 New Login

Purpose

The purpose of this test is to verify that add-user calls are correctly processed. This test will need to take a new user/password string and add it to the vector of logins. If the given login already exists, the method should return a failure.

Inputs

A string containing a username and password.

Expected Outputs

A boolean representing if the new login account was successfully added or not.

Pass Criteria

New login information successfully being added. Duplicate information being successfully rejected.

Test Procedure

Send a string containing new login information, receive a success. Send a string with duplicate information, receive a failure.

3.2 Client Test Cases

3.2.1 User Login

Purpose

To ensure regular users are able to login and the correct features are available to them

Inputs

Username and password of a non-admin account

Expected Outputs

Users should only have access to the ‘Search’ and ‘Tutorial’ features

Pass Criteria

-Able to login using correct credentials
-Able to access ‘Search’ or ‘Tutorial’ tabs
-No access to admin features such as adding users etc.
-Incorrect credentials are rejected

Test Procedure

1. Type in username and password
2. Verify credentials allow login
    a. Verify incorrect credentials do not allow login
3. Confirm only ‘Search’ and ‘Tutorial’ tabs are visible and accessible

3.2.2 Admin Login

Purpose

To ensure administrators able to login and the correct features are available to them

Inputs

Username and password of an admin account

Expected Outputs

Admins should have access to all features: ‘Search’, ‘Tutorial’, ‘Users’, ‘Resources’ and ‘Attributes’

Pass Criteria

-Able to login using correct credentials

-Able to access any or all tabs

-Access to admin-only features (user, resource and attribute management)

Test Procedure

1. Type in admin username and password
2. Verify credentials allow login
3. Confirm only ‘Search’, ‘Tutorial’, ‘Users’, ‘Resources’ and ‘Attributes’ tabs are visible and accessible

3.2.3 User/Admin Search

Purpose

Testing the search feature on both user and admin sides

Inputs

Search criteria and method (ie. ‘Title’, ‘Author’ etc.)

Expected Outputs

A listing of all relevant resources matching the search criteria (if any)

Pass Criteria

-Ability to search for any string (special characters etc.)

-Returned list displays relevant resources

-If no results are found, an appropriate message is displayed

Test Procedure

1. Searching with matches:
    a. Enter search criteria with known results
    b. Confirm returned resources are as expected
2. Searching with no matches:
    a. Enter search criteria with no relevant resources
    b. Confirm no results are returned and a message is displayed

3.2.4 Adding/Removing Users (Admin)

Purpose

Testing the administrator features to create new users and remove existing users

Inputs

New user credentials (if adding a user)

Expected Outputs

New users are able to login using their credentials and have access to correct features
Removed users are unable to login

Pass Criteria

-New users are added and can use the necessary features of the system
-Usernames existing in the system may not be reused
-Removed users are no longer stored in the system
-Removed user cannot login or access any feature

Test Procedure

1. Create admin user
    a. Enter new username and password
    b. Ensure ‘IsAdmin’ is selected
    c. Validate using ‘Admin Login’ test case
2. Create regular user
    a. Enter a new username and password
    b. Ensure ‘IsAdmin’ is not selected
    c. Validate using ‘User Login’ test case
3. Remove admin/user
    a. Login as admin
    b. Under the user listing, select ‘Delete’
    c. Verify the selected user was removed using the login test cases

3.2.5 Adding & Updating Resources

Purpose

To verify resources are able to be correctly added or modified by administrators

Inputs

Information about the resource being added/changed (ie. ‘Title’, ‘Author’ etc.)

Expected Outputs

New resource is added to the database
Updated resources are changed accordingly

Pass Criteria

-Added resources are able to be found via search functions
-Updated resources will display any modified information when searched for

Test Procedure

1. Adding a resource
    a. Enter all required information about the new resource
    b. Ensure relevant keywords are added for searching
    c. Verify the new resource can be found by using the search test case
2. Updating a resource
    a. Locate the resource to be modified
    b. Select the ‘To Modify’ checkbox
    c. Change field information as necessary

Verify the resource can be found using updated criteria in the search test case

3.3 GUI Test Cases

All tests within this subsection apply to the GUI module.

3.3.1: User Login

Purpose

The purpose of this test is to ensure that the login functionality works. It must make sure it can handle there being no connection, a failed login, a user login, and an admin login.

Inputs

A username, a password, the server's IP, and the server's port.

Expected Outputs

Either a failing login message, or else a new GUI pane built for the login type.

Pass Criteria

Connection times out, a failed login displays the message, or the client connects to the server with the password matching the provided username's in the database.

Test Procedure

Create a Login test stub, and manually type in code to see if it works.

Scenario 1: No Connection

Output

Error message in login screen that displays "Error: Unable to connect to server."

Scenario 2: Login Fails

Output

Error message in login screen that displays "Error: Username and Password mismatch."

Scenario 3: User Login Successful

Output

New pane built for a regular user.

Scenario 4: Admin Login Successful

Output

New pane built for an admin

3.3.2: Search for Resources

Purpose

This test case makes sure that the search function of the GUI works correctly.

Inputs

A user selected attribute type, and a search string typed into the search text field.

Expected Outputs

Either a display message saying “No Search Results Found” or else the list of resource results.

Pass Criteria

The connection properly times out, or else the expected outputs are displayed.

Test Procedure

Create stub to provide a small selection of resources to mimic the client functionality, manual input of search criteria into the GUI,

Scenario 1: Connection Dropped

Output

GUI returns to the login screen

Scenario 2: No Results found

Output

Display the message “No Results Found”

Scenario 3: Results found

Output

A page of the search results, and allow for the next page button to be pressed. The filter resources section will be populated.

3.3.3: Filter Resources

Purpose

To test that the filter resources functionality of the GUI works correctly.

Inputs

A user click on a filter resource checkbox.

Expected Outputs

A filtered list of resources based on the selected keyword's checkbox.

Pass Criteria

The connection properly times out, the list of resources is properly filtered based on the selected keyword.

Test Procedure

Create stub to mimic the client functionality to handle filter resource selection.

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: Filter Successful

Output

A filtered list of resources is displayed.

3.3.4: Sort Resources

Purpose

To test that the sort functionality of the GUI works as expected.

Inputs

A user clicks on an attribute's ascending or descending button.

Expected Outputs

The connection properly times out, or a sorted list of resources.

Pass Criteria

The connection properly times out, the display is a sorted list of resources.

Test Procedure

Create stub to mimic the client functionality to handle sort functionality.

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: No search results

Output

Nothing

Scenario 3: Sort Successful

Output

A sorted list of resources is displayed.

3.3.5: Add New User

Purpose

To test if the GUI is able to pass the data to the client and then display an appropriate message.

Inputs

A new username and password.

Expected Outputs

A message displaying the result.

Pass Criteria

The connection properly times out, a new user has been added, or the detection of a user already existing.

Test Procedure

Create stub to mimic the client functionality to handle user functionality

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: User Added Successfully

Output

A message displaying “New user <username> has been added”

Scenario 3: User Already Exists

Output

a message displaying “User <username> already exists.”

3.3.6: Remove Existing User

Purpose

To make sure that the remove user GUI functionality is working.

Inputs

A user is selected from a list of existing users to be deleted

Expected Outputs

A message displaying the result

Pass Criteria

The connection properly times out, the user is successfully removed.

Test Procedure

Create stub to mimic the client functionality to handle user functionality

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: User Removed Successfully

Output

A message displaying “User <username> has been successfully removed.”

3.3.7: Add New Resource

Purpose

To make sure that the add new resource GUI functionality is working as expected.

Inputs

A string for all required attributes and any optional attributes that maybe be provided.

Expected Outputs

An error or success message.

Pass Criteria

The connection properly times out, the Resource is added, or the missing important attribute is detected,

Test Procedure

Create stub to mimic the client functionality to handle resources functionality

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: Add New Resource Successful

Output

A message displayed saying “New resource has been added successfully”

Scenario 3: Resource Missing Important Attribute

Output

A message displayed saying “New resource missing <attribute>. Could not be added.”

3.3.8: Modify Existing Resource

Purpose

To make sure that the modify resource GUI functionality is working as expected.

Inputs

A selected resource to be removed.

Expected Outputs

A success or fail message.

Pass Criteria

The connection properly times out, the resource is successfully modified, or the missing important attribute is detected.

Test Procedure

Create stub to mimic the client functionality to handle resources functionality

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: Modify Successful

Output

A message display saying “Resource has been successfully modified.”

Scenario 3: Missing Important Attribute

Output

A message display saying “Resource missing <attribute>. Could not be modified.

3.3.9: Remove Existing Resource

Purpose

To make sure that the remove resource GUI functionality is working as expected.

Inputs

A resource selected to be deleted.

Expected Outputs

A success message.

Pass Criteria

The connection properly times out, the resource is successfully deleted.

Test Procedure

Create stub to mimic the client functionality to handle resources functionality

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: Resource Removed Successful

Output

A message display saying “Resource has been successfully removed”

3.3.10: Add New Attribute

Purpose

To make sure that the add new attribute GUI functionality is working as expected.

Inputs

A new attribute name, and optionally a default value

Expected Outputs

A success or fail message.

Pass Criteria

The connection properly times out, a new attribute is added, or the existence of the attribute already existing.

Test Procedure

Create stub to mimic the client functionality to handle attribute functionality

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: Attribute Added Successful

Output

A message displaying "Attribute <attribute> successfully added."

Scenario 3: Attribute already exists

Output

A message displaying "Attribute <attribute> already exists, could not be added."

3.3.11: Remove Attribute

Purpose

To make sure that the remove attribute GUI functionality is working as expected.

Inputs

A user selected attribute to remove

Expected Outputs

A message displaying that the removal of the attribute was successful, or that the attribute is protected.

Pass Criteria

The connection properly times out, the attribute is removed, or a protected attribute is properly detected.

Test Procedure

Create stub to mimic the client functionality to handle attribute functionality

Scenario 1: Connection Dropped

Output

GUI returns to the login screen after timing out.

Scenario 2: Attribute is a protected Attribute

Output

A message displaying "Attribute <attribute> is protected and cannot be removed."

Scenario 3: Attribute removed successfully

Output

A message displaying "Attribute <attribute> has been successfully removed."

3.4 Integration Test Cases

3.4.1 New Login

Purpose

The purpose of this test is to verify that an admin logged in through a client can successfully add a new user to the list of users. It will involve the admin being logged in and connected, opening the GUI Users window, adding new user information and sending it to the server. After a successful add, the admin's user pane should be updated to show the new user.

Inputs

A new username and password entered into the admin's user window in the GUI

Expected Outputs

An updated user list with the new username added to it.

Pass Criteria

Pass if valid usernames and passwords are added successfully to the list of names, and all invalid entries (duplicates and invalid characters in usernames) are rejected.

Test Procedure

Starting from a valid admin login (as that is not part of this test): Open the users pane. Add a new username and password, and click "add user". After a moment, the list of users should be updated with the new username and the message "New user <username> has been added" is displayed. Try to add another new user, which has invalid characters in the username. A message "<username> could not be added due to invalid username" should be displayed.

3.4.2 New Search

Purpose

The purpose of this test is to verify correct search results are displayed in the client GUI after a search. It should involve searching by each attribute at least once, and should use a smaller set of known data to verify that all correct matches are found and displayed.

Inputs

A selected search attribute, and an entered search string.

Expected Outputs

A vector of resources displayed in the GUI search window.

Pass Criteria

All matching resources are displayed within the search window (over multiple pages, if necessary). Searches with no results should display an empty results field.

Test Procedure

Set up the server system with a small set of known entries. After being successfully logged in to a client program (as either user or admin), navigate to the search window, select an attribute from the dropdown, and enter a search string which is known to match a resource currently stored in the server system's resources vector. Repeat with each attribute, and with both matching and nonmatching search strings.

3.4.3 Apply Filter

Purpose

The purpose of this test is to verify that search filters apply correctly. After already receiving a set of search results, a list of filterable keywords should populate the box to the right of the resources. After any filter is selected, the list of resources should repopulate with all search results containing the selected keyword.

Inputs

A keyword being selected in the side pane.

Expected Outputs

A new page of resources, which all contain the selected keyword.

Pass Criteria

All resources that were returned by the original search string, and which also contain all selected keywords, are displayed within the search window. Also, an updated list of keywords showing both the already selected keyword (marked) and all other filterable keywords (if applicable).

Test Procedure

Set up the server system with a small set of known entries. After being successfully logged in to a client program (as either user or admin), and receiving a set of search results, apply a filter that will give a known subset of resources. Verify the set matches what it should be. Repeat with different filters, and multiple filters at once.

3.4.4 Concurrent Connections

Purpose

Load testing multiple simultaneous users logged on to the system at the same time to ensure server capability.

Inputs

Multiple login credentials on multiple system

Expected Outputs
Multiple active logins able to use all features at one time

Pass Criteria
-Users are able to login at the same time
-Multiple active requests for searching etc.

Test Procedure
1. Login on multiple systems at the same time
2. Verify searching capabilities
3. Verify adding/removing users
4. Verify adding/removing resources

3.4.5 Volume Testing

Purpose
Confirm the ability to retrieve and manage large amounts of data over the connection.

Inputs
Search criteria and method (ie. ‘Title’, ‘Author’ etc.)

Expected Outputs
Listing of large amount of returned resources

Pass Criteria
-Ability to search for criteria involving a large amount of returned results
-Returned list is complete and accurate
-Results are returned in a realistic timeframe

Test Procedure
1. Enter search criteria that returns a known large listing of resources
2. Ensure the list returned is complete
3. Confirm returned resources are as expected
4. Verify the server connection remains active for other users during the retrieval process