*SCATS Application*

Unit test documentation

Version *1.0*

*01/19/2020*

Contents

[Purpose 3](#_Toc30413721)

[Unit Test Plan 3](#_Toc30413722)

[Unit Test Summary 3](#_Toc30413723)

[Unit Test Details 4](#_Toc30413724)

[**Module (Case#001)**: AddDocumentController.java [Test Failed] 4](#_Toc30413725)

[**Module (Case#002)**: EditSystemUserController.java [Test Passed] 6](#_Toc30413726)

[**Module (Case#003)**: EngagementOptionsController.java [Test Passed] 8](#_Toc30413727)

[**Module (Case#004)**: ManageConsultantsController.java [Test Failed] 11](#_Toc30413728)

[**Module (Case#005)**: ManageSystemUsersController.java [Test Passed] 13](#_Toc30413729)

[**Module (Case#006)**: NewCertificationController.java [Test Passed] 15](#_Toc30413730)

[**Module (Case#007)**: NewConsultantController.java [Test Passed] 17](#_Toc30413731)

[**Module (Case#008)**: UpdateCertificationController.java [Test Passed] 19](#_Toc30413732)

[**Module (Case#009)**: UpdateEngagementController.java [Test Failed] 22](#_Toc30413733)

[**Module (Case#010)**: NewEngagementController.java [Test Passed] 26](#_Toc30413734)

[**Module (Case#011)**: ViewConsultantController.java [Test Failed] 29](#_Toc30413735)

## Purpose

This SCATS Application Test Report provides a summary of the results of tests performed as outlined within this document.

## Unit Test Plan

Java-based modules to manage data inputs from users interacting with FXML interface designs (controllers) will be the primary focus of Unit Testing. Unit tests will address the core functionality of each interface controller, ensuring the modules perform as expected, generate accurate data points, and handle exceptions gracefully.

Interface Controllers to be tested:

* AddDocumentController.java
* EditSystemUserController.java
* EngagementOptionsController.java
* ManageConsultantsController.java
* ManageSystemUsersController.java
* NewCertificationController.java
* NewConsultantController.java
* UpdateCertificationController.java
* UpdateEngagementController.java
* ViewConsultantController.java
* NewConsultantController.java

## Unit Test Summary

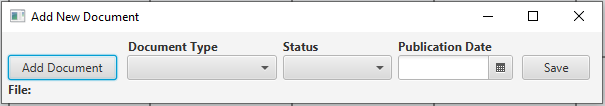
The SCATS application underwent unit testing on its functional controller modules to ensure the core functionality of the module performed accurately and as expected. The testing performed assessed 11 individual controller modules, each containing various features and functions that stood to execute one primary function of the application. Of the 11 cases assessed, 4 did not pass the unit tests performed.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Module** | **Test Case ID** | **Pass/Fail** | **Summary of Defect** | **Comments** |
| AddDocumentController.java | Case001 | Fail | The save date of the file is one day behind. | Fix: address the new document function in the Database Manager. |
| ManageConsultantsController.java | Case004 | Fail | Functionality Not Implemented | The Delete Consultant functionality is not implemented/mapped to the menu event. Manage Consultants can also benefit from an item count label. |
| UpdateEngagementController.java | Case009 | Fail | The end date of the Fieldwork Duration is inaccurate. | Fix: address the UpdateEngagement function in the Database Manager. |
| ViewConsultantController.java | Case011 | Fail | The system cannot find the image file because the filepath used within the function contains bad characters. | Fix is to implement relative path functionality to retrieve the file from the file system. |

# Unit Test Details

## **Module (Case#001)**: AddDocumentController.java [Test Failed]

**Interface**:



**Inputs to Module**:

* File document – selected from the file system to be associated with an Engagement
* Document Type <String> -- selected from an existing set of values
* Document Status <String> -- selected from an existing set of values
* Publication Date <Date> -- selected from a DatePicker to represent the publication date of the document

**Outputs from Module**:

* Module will successfully display the filename of the File Documented selected for association
* Module will validate all inputs into the form, ensuring all elements are populated
* Module will successfully create a local directory within the SCATS workspace based on the name of the Company specified in the Engagement
* Document attributes will be successfully persisted in the backend database

**Logic Flow Diagram**:

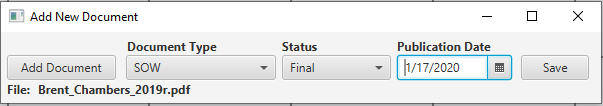
1. User clicks <Add Document> and selects a document from the local file system
2. User selects a document type from the drop-down menu
3. User selects a document status from the drop-down menu
4. User specifies a publication date
5. User clicks save

**Test Data:**

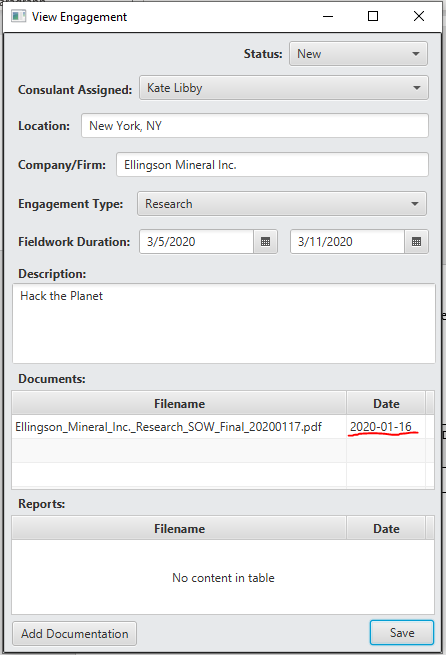
* Document: **%DOCUMENTS%/Brent\_Chambers\_2019r.pdf** as the Input Document

**Positive Test Cases:**

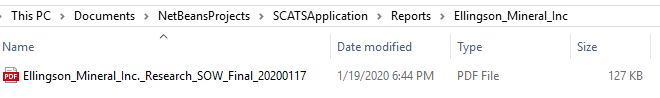
Associating the Document File with “Ellingson Mineral Inc.” Engagement



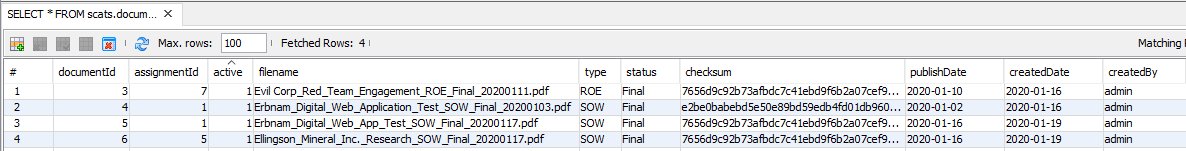
Upon Saving, the file is successfully loaded and associated with the engagement, but the date is misrepresented by one day.



The Ellingson Mineral Inc. folder is successfully created upon saving the file, and the file is successfully renamed and placed in the directory.



Finally, the File database record can be observed within the database:



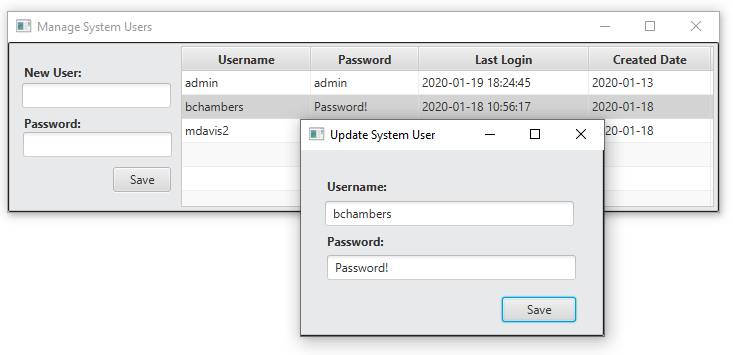
Test Results Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case002 | 1/19/2020 | Failed | Minimal | The save date of the file is one day behind. | Fix: address the new document function in the Database Manager. |

## **Module (Case#002)**: EditSystemUserController.java [Test Passed]

**Description**: Allows the user to update the username and/or password of a system user.

**Interface**:



**Inputs to Module**:

* SystemUser Username <String> -- provided as text
* SystemUser Password <String> -- provided as text

**Outputs from Module**:

* Module will ensure password are no less than 8 characters.
* Module will apply appropriate changes to the existing System User.
* SystemUser changes will be successfully persisted on the backend database.

**Logic Flow Diagram**:

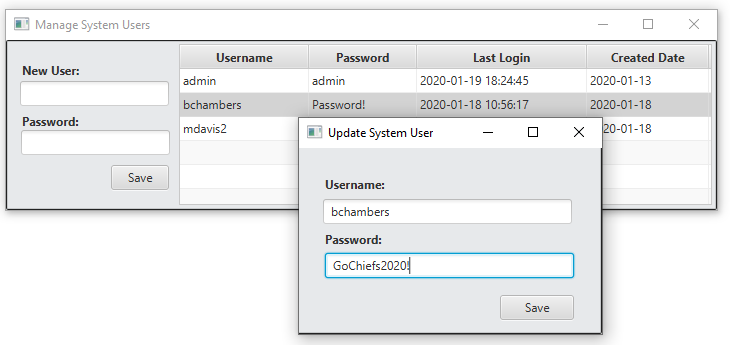
1. User right-clicks an existing SystemUser from Manage System Users tool.
2. Module will populate the form with the username and password of the existing SystemUser.
3. User will change the username and password of the existing SystemUser.
4. User clicks save

**Test Data:**

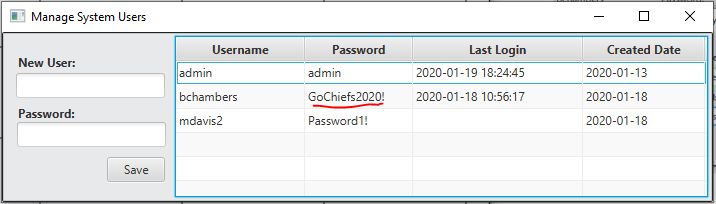
* Case#001
  + <String> Username
  + <String> Password: GoChiefs2020!

**Positive Test Cases:**

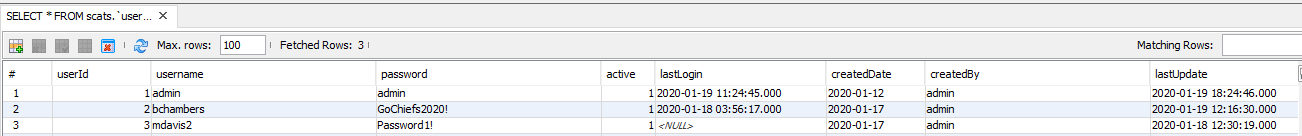
Changing the password of the bchambers SystemUser account.



Upon Saving, the password of bchambers is successfully changed.



The SystemUser database record can be observed within the database:



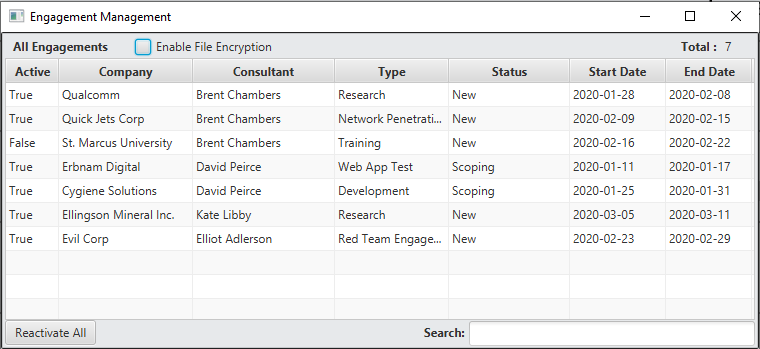
Summary

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case001 | 1/19/2020 | Passed | None | NA | NA |

## **Module (Case#003)**: EngagementOptionsController.java [Test Passed]

**Description**: Provides an interface to activate/deactivate, search, and view all active and inactive engagements

**Interface**:



**Inputs to Module**:

* Search Query <String> -- provided as text

**Outputs from Module**:

* Module will load all active and inactive engagements.
* Module will provide the ability to toggle the active and inactive modes of engagements.
* Module filter engagement results based on specified search query.
* Module will accurately display the number of active and inactive engagement.
* Module will set active to all engagements upon “Reactivate All” button press.

**Logic Flow Diagram**:

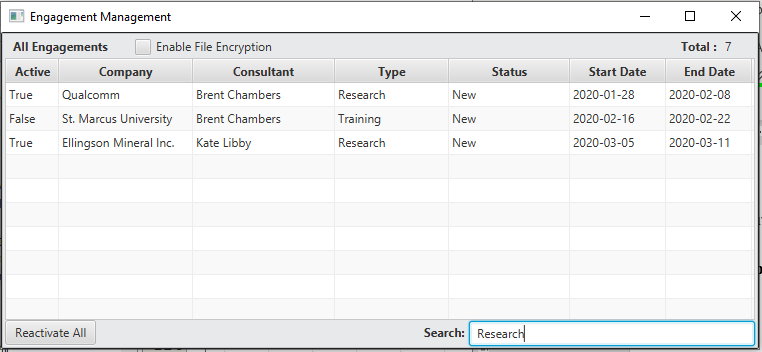
1. User select Manage Engagements from main SCATS UI window.
2. User will search Engagement Management for engagements classified as “Research”.
3. User will deactivate engagements classified as “Research”.
4. User will reactivate all engagements using the “Reactivate All” button.

**Test Data:**

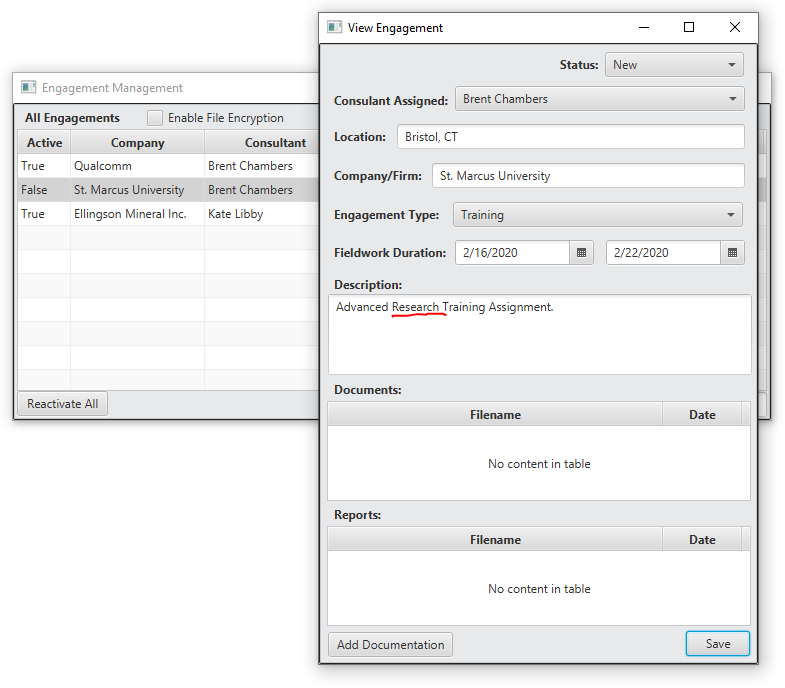
* Case#001
  + <String> Search Query: “Research”

**Test Case:**

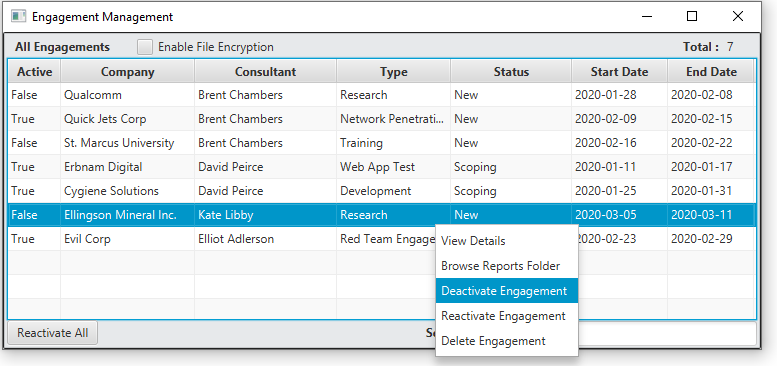
Searching Engagements for Research:



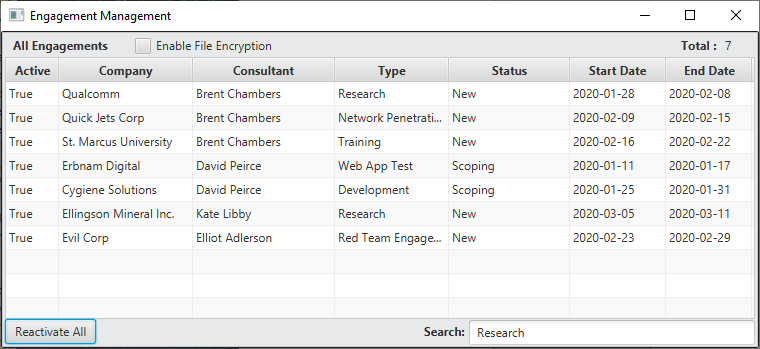
System effectively returns three results, including an inactive engagement at St. Marcus University where the word “research” is in the description.



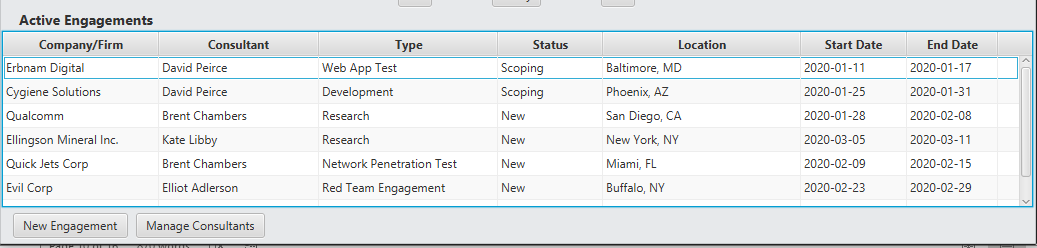
The Deactivate Engagement selection item was applied to both the St. Marcus University engagement, as well as the Ellingson Mineral Inc. engagement.



The Reactivate All button was pressed to successfully reactivate all existing engagements.



The Main SCATS UI screen, which only displays active engagements, does not immediately reflect the changes made to all stored engagements.



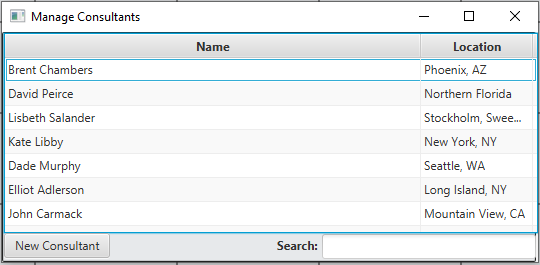
Summary:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case003 | 1/19/2020 | Passed | None | NA | Main UI interface needs to be updated after Reactivate All is executed in Manage Engagements.  The Main UI could also benefit from a Search function and an engagement counter. |

## **Module (Case#004)**: ManageConsultantsController.java [Test Failed]

**Description**: Provides an interface to add, delete, and search for consultants housed in the SCATS system.

**Interface**:



**Inputs to Module**:

* Search Query <String> -- provided as text

**Outputs from Module**:

* Module will load all consultants
* Module filter engagement results based on specified search query.
* Module will provide functionality to add a new consultant to the SCATS system.
* Module will provide functionality to delete a consultant from the SCATS system.

**Logic Flow Diagram**:

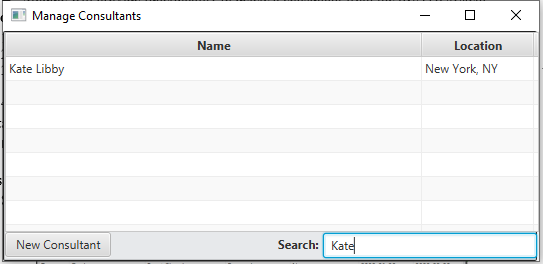
1. User select Consultants -> Manage from main SCATS UI window
2. User will search Consultants Management tool for consultants named “Kate”
3. User will create a new consultant “Jesse Bradford”, providing all necessary data points for the new consultant.
4. User will delete newly created consultant “Jesse Bradford”.

**Test Data:**

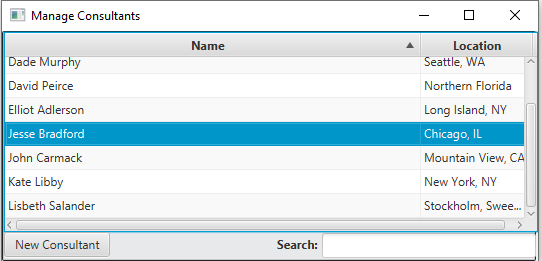
* Case#001
  + <String> Search Query: “Kate”

**Test Case:**

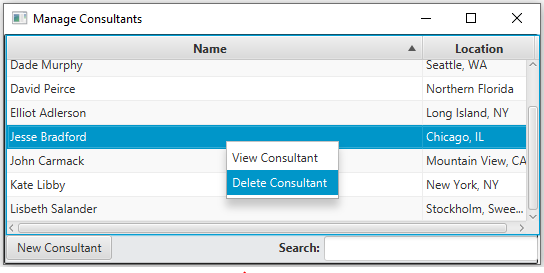
Searching Engagements for “Kate” was successful.



The successful creation of new consultant “Jesse Bradford” was successful using the “New Consultant” button.



The attempted deletion of consultant “Jesse Bradford” was unsuccessful. The functionality is not mapped to the “Delete Consultant” menu item.



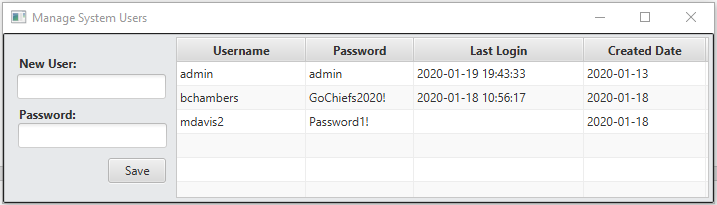
Summary:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case004 | 1/19/2020 | Fail | Medium | Functionality Not Implemented | The Delete Consultant functionality is not implemented/mapped to the menu event.  Manage Consultants can also benefit from an item count label. |

## **Module (Case#005)**: ManageSystemUsersController.java [Test Passed]

**Description**: Provides an interface to add, delete, and modify System User Accounts.

**Interface**:



**Inputs to Module**:

* SystemUser Username <String> -- msmith
* SystemUser Password <String> -- Password1!

**Outputs from Module**:

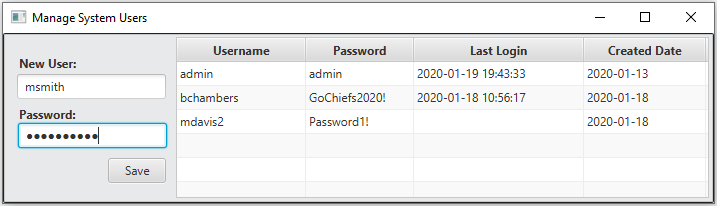
* Module will load all existing System Users
* Module will provide functionality to add and delete a System User.

**Logic Flow Diagram**:

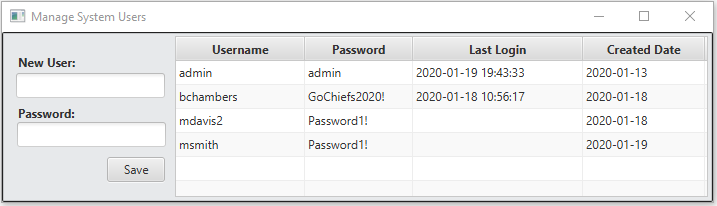
1. User selects File -> Manage Users from main SCATS UI window
2. User will add new system user: Mike Smith (username: msmith::Password1!)
3. User will delete new system user Mike Smith
4. Module will reflect all changes made.

**Test Case:**

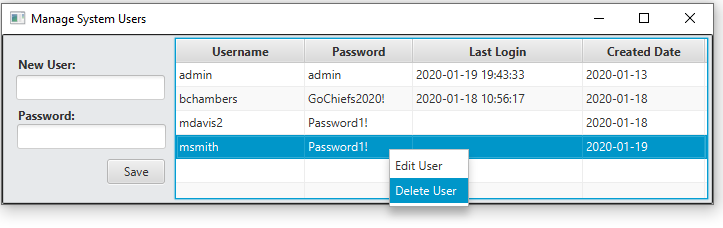
System User information is populated in the New User and Password fields.



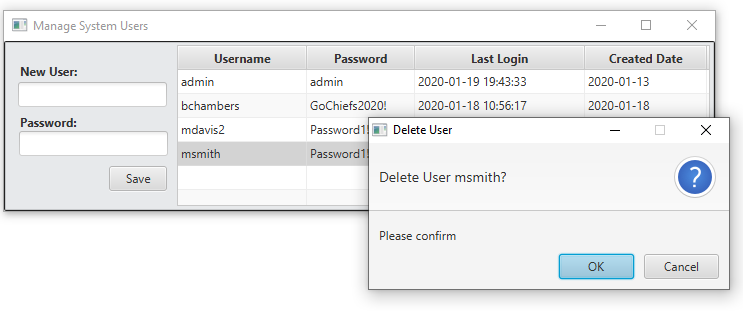
Upon clicking SAVE, the user is added to the Manage System User’s primary table.

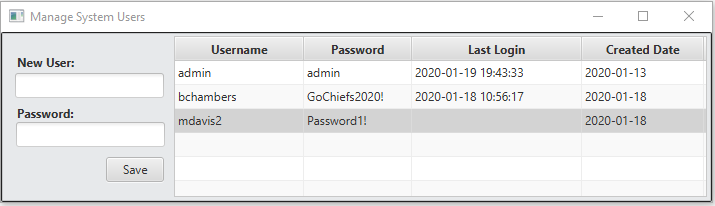


The Delete User menu item is selected after right clicking the new user.



The System User was successfully removed from the SCATS system after being prompted for confirmation.





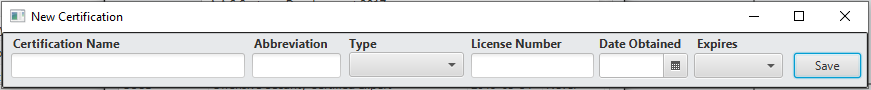
Summary:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case005 | 1/19/2020 | **Pass** | None | None | None |

## **Module (Case#006)**: NewCertificationController.java [Test Passed]

**Description**: Provides an interface to add a new Certification to be associated with a Consultant.

**Interface**:



**Inputs to Module**:

* Certification Name <String> -- provided as text
* Abbreviation <String> -- provided as text
* Certification Type <String> -- selected from existing items
* Certification License Number <String> -- provided as text
* Date Obtained <Date> -- selected from a DatePicker to represent the date the certification was obtained
* Certification Expiry <String> selected from existing items

**Outputs from Module**:

* Module will validate all inputs into the form, ensuring all elements are populated
* Module will successfully associate the new certification with the consultant from which the New Certification window was spawned
* Certificate attributes will be successfully persisted in the backend database

**Logic Flow Diagram**:

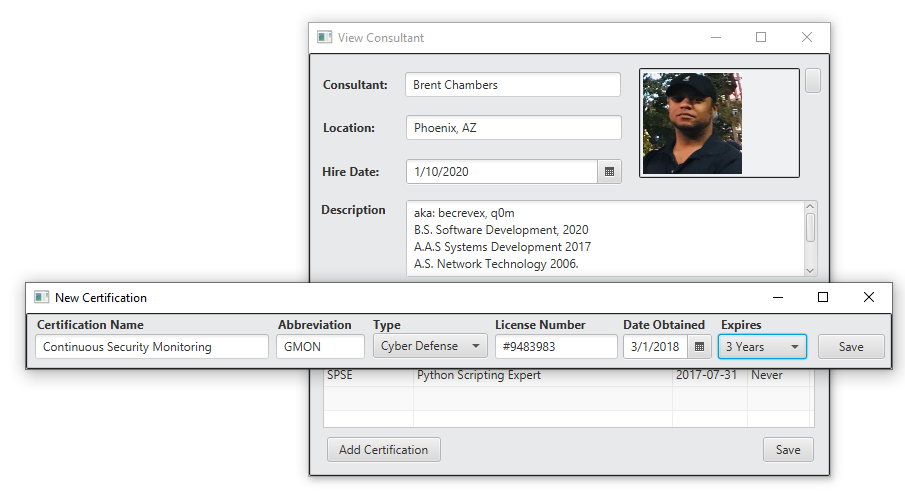
1. User clicks <Add Certification> after viewing the details of an existing Consultant
2. User supplies a certification name “Continuous Security Monitoring”
3. User supplies a certification abbreviation “GMON”
4. User selects a certification type “Cyber Defense”
5. User supplies a license number “#9483983”
6. User specifies the date obtained
7. User selects an expiration time frame
8. User clicks SAVE

**Test Data:**

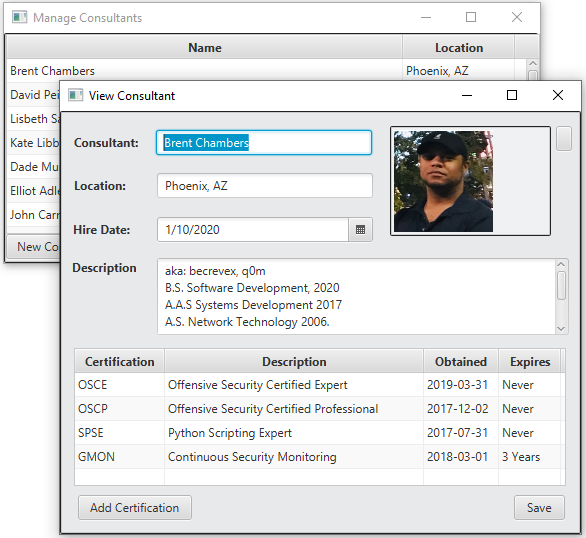
* Name: “Continuous Security Monitoring”
* Abbreviation: “GMON”
* Type: Cyber Defense
* License Number “#9483983”
* Date Obtained: 3/1/2018
* Expiration: 3 Years

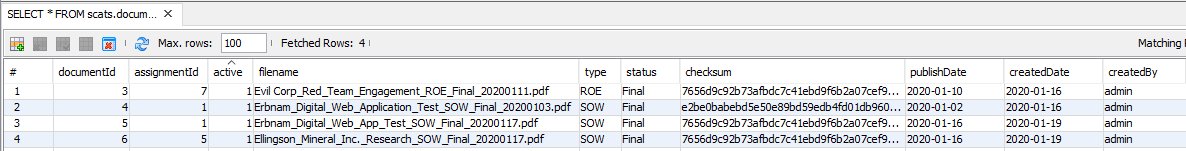
**Test Case:**

The “Add Certification” dialog was launched from the View Consultant details of “Brent Chambers” and successfully populated with test data.



The new certification is visible within the View Consultant details of consultant “Brent Chambers”.





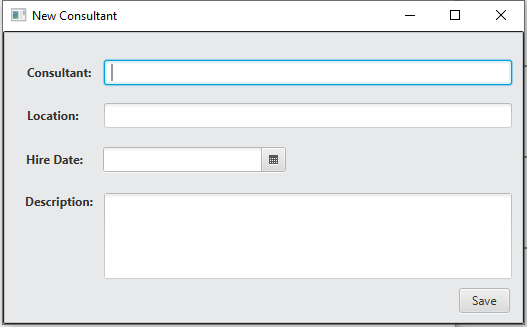
Test Results Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case006 | 1/19/2020 | Pass | None | None | None |

## **Module (Case#007)**: NewConsultantController.java [Test Passed]

**Description**: Provides an interface to add a new Consultant to the SCATS System.

**Interface**:



**Inputs to Module**:

* Consultant Name <String> -- provided as text
* Location <String> -- provided as text
* Hire Date <Date> -- selected from a DatePicker to represent the hire date of the consultant
* Description <String> selected from existing items

**Outputs from Module**:

* Module will validate all inputs into the form, ensuring all necessary elements are populated
* Module will successfully store a new consultant to the SCATS System
* Certificate attributes will be successfully persisted in the backend database

**Logic Flow Diagram**:

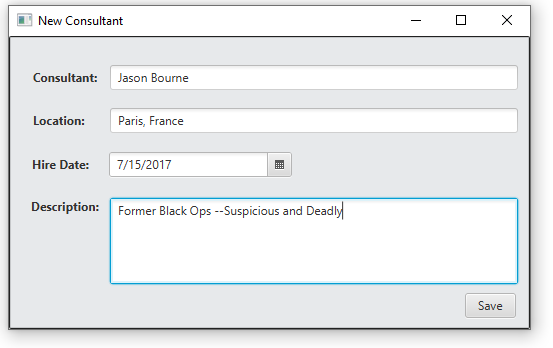
1. User supplies a Consultant name
2. User supplies a Consultant location
3. User selects a hire date
4. User supplies a Consultant description
5. User clicks SAVE

**Test Data:**

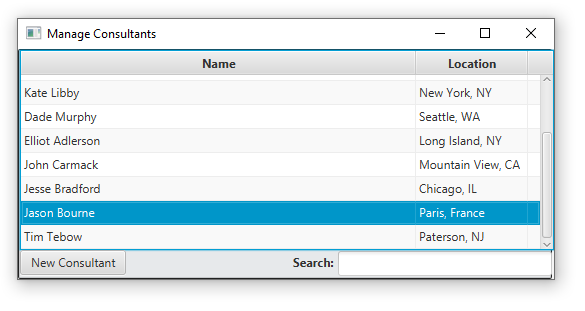
* Name: “Jason Bourne”
* Location: Paris, France
* Hire Date: 7/15/2017
* Description: “Former Black Ops --Suspicious and Deadly

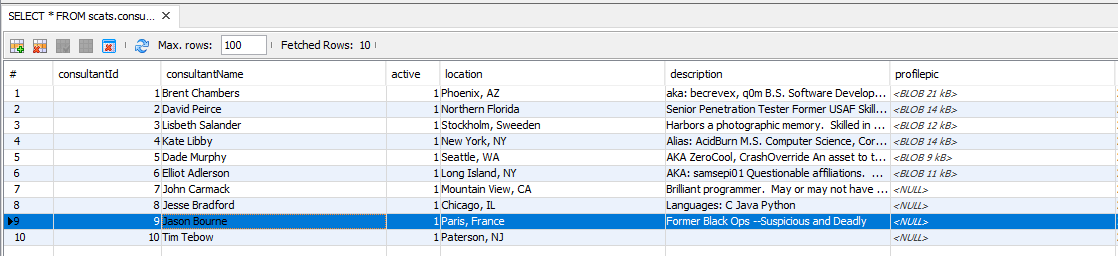
**Test Case:**

The “New Consultant” dialog was launched from the New Consultant menu item on the main SCATS UI interface and successfully populated with test data.



Upon population, the SAVE button was clicked and the consultant was successfully added to the SCATS system.





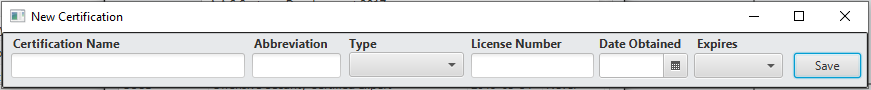
Test Results Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case007 | 1/19/2020 | Pass | None | None | None |

## **Module (Case#008)**: UpdateCertificationController.java [Test Passed]

**Description**: Provides an interface to modify a Certification associated with a Consultant.

**Interface**:



**Inputs to Module**:

* Certification Name <String> -- provided as text
* Abbreviation <String> -- provided as text
* Certification Type <String> -- selected from existing items
* Certification License Number <String> -- provided as text
* Date Obtained <Date> -- selected from a DatePicker to represent the date the certification was obtained
* Certification Expiry <String> selected from existing items

**Outputs from Module**:

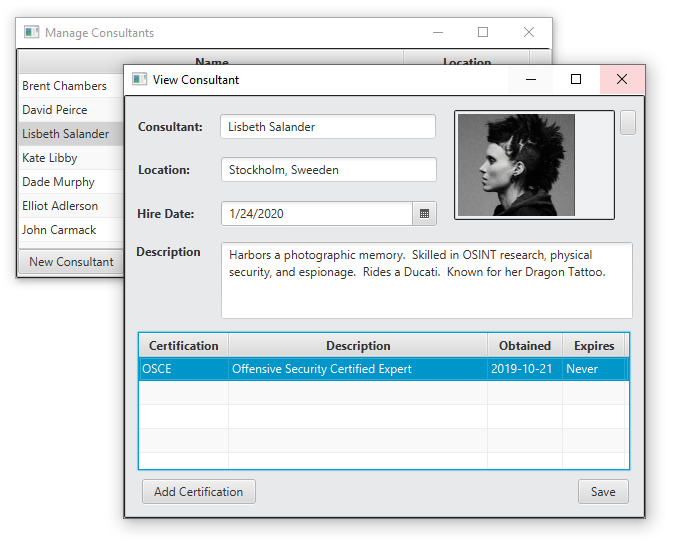
* Module will validate all inputs into the form, ensuring all elements are populated
* Module will save new changes and update the existing certification record associated with the consultant
* Certificate attributes will be successfully persisted in the backend database

**Logic Flow Diagram**:

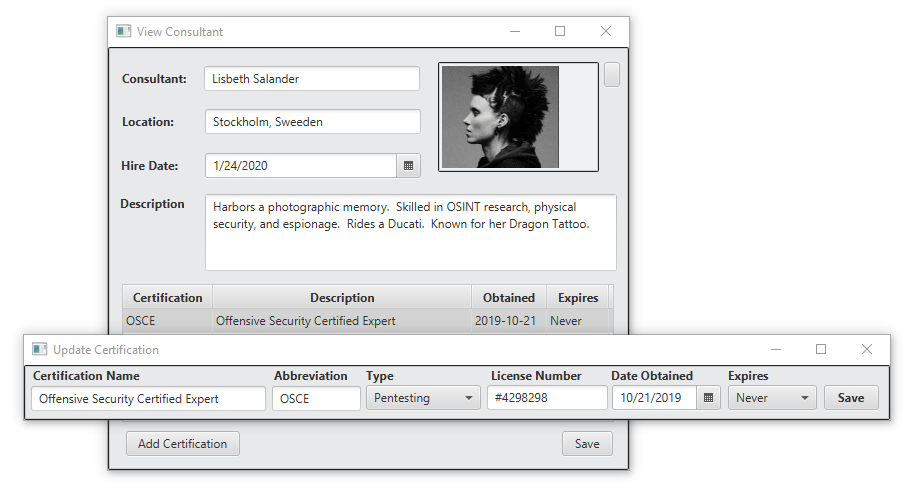
1. From View Engagement details of a Consultant, User right-clicks an existing certification, and selects “Update Certification”.
2. Update Certification dialog is spawned with the form fields populated with Certification attributes.
3. User changes three Certification attributes.
4. User clicks SAVE
5. Changes to the Certification should be observed within the Consultant form and Database.

**Test Case:**

Consultant, Lisbeth Salander is chosen from the Manage Consultants tool. Upon viewing the consultant details, it is observed Consultant Salander has one certification: OSCE.



The OSCE certification is right-clicked, and “Update Certification” is chosen. The Update Certification tool is displayed with the attributes of Salander’s OSCE certification populated within the form:

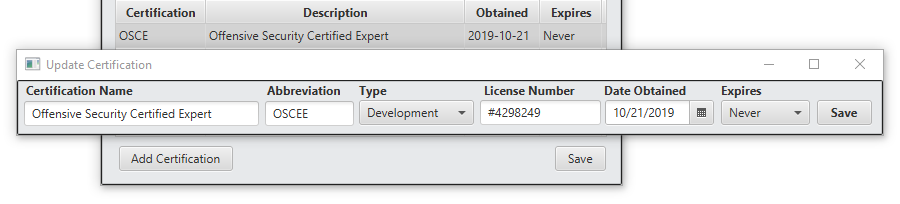


Attributes Abbreviation, Type, and License number are all changed to different values:

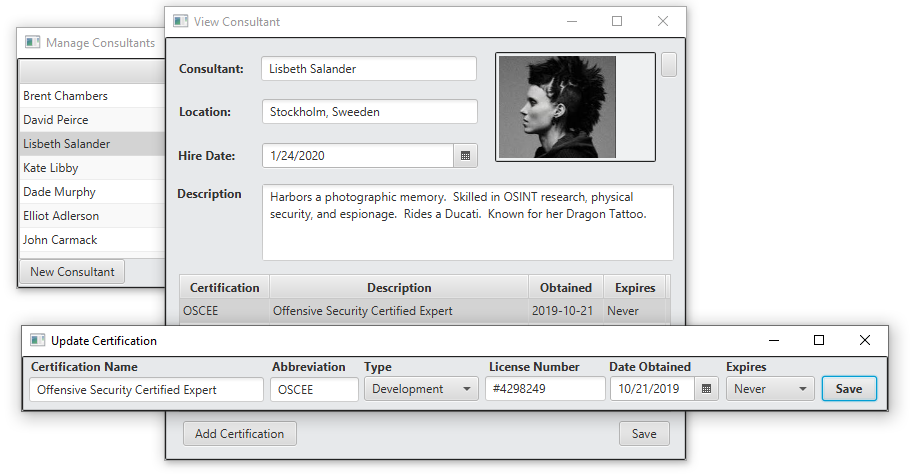
Abbreviation: OSCE 🡪 OSCEE

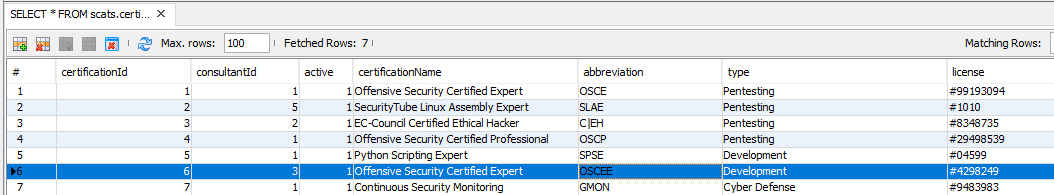
Type: Pentesting 🡪 Development

License Number: #4298298 🡪 #4298249



Upon saving, changes made to the Lisbeth OSCE certification are visible in the View Consultant details, the Update Certification dialog tool, as well as the backend database.





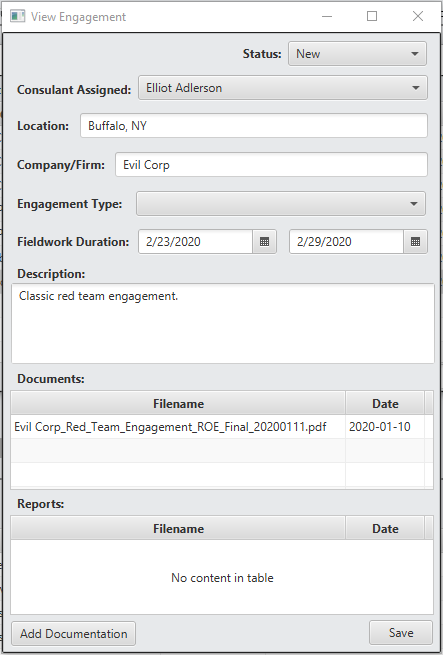
Test Results Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case008 | 1/19/2020 | Pass | None | None | None |

## **Module (Case#009)**: UpdateEngagementController.java [Test Failed]

**Description**: Provides an interface to modify an Engagement.

**Interface**:



**Inputs to Module**:

* Engagement Status Name <String> -- selected from existing items
* Engagement Consultant Assigned <String> -- selected from existing items
* Engagement Location <String> -- provided as text
* Engagement Company/Firm <String> provided as text
* Engagement Type <String> selected from existing items
* Engagement Fieldwork Duration <Dates>

**Outputs from Module**:

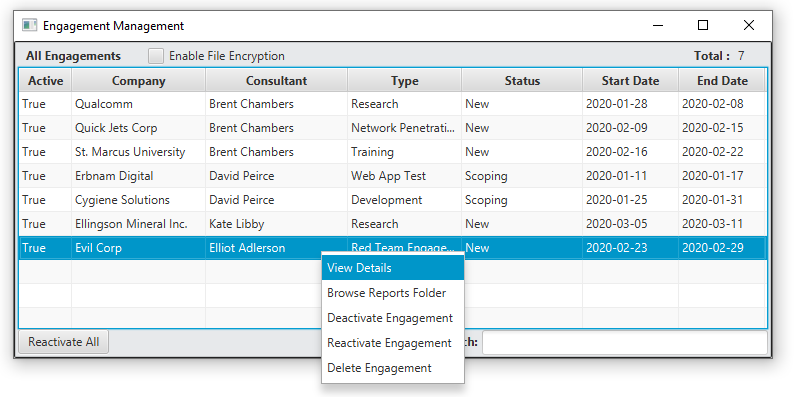
* Module will validate all inputs into the form, ensuring all elements are populated
* Module will save new changes and update the existing engagement record
* Engagement data fields will be successfully persisted in the backend database

**Logic Flow Diagram**:

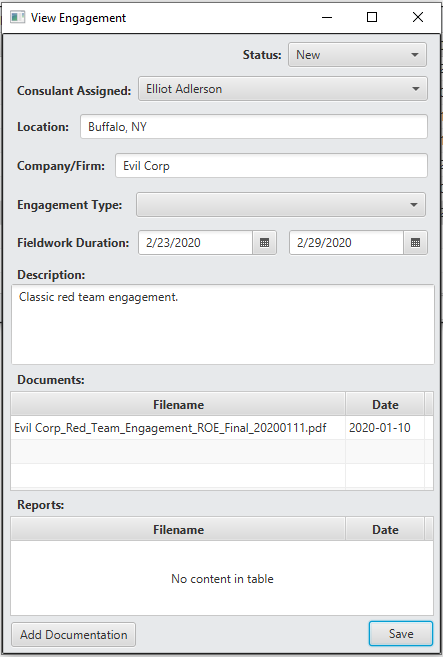
1. The Engagement Management tool is launched from the main SCATS menu bar (Engagements 🡪 Manage Engagements). From the Engagement Management tool, an existing engagement is right-clicked and “View Details” is selected to make changes to the Engagement.
2. User changes three Engagement attributes.
3. User clicks SAVE
4. Changes to the Engagement should be observed within the Engagement Management tool and the backend database.

**Test Case:**

The Evil Corp engagement performed by Elliot Alderson was selected from the Engagement Management tool and viewed using the “View Details” drop-down menu selection item.



The engagement details are effectively populated within the form.

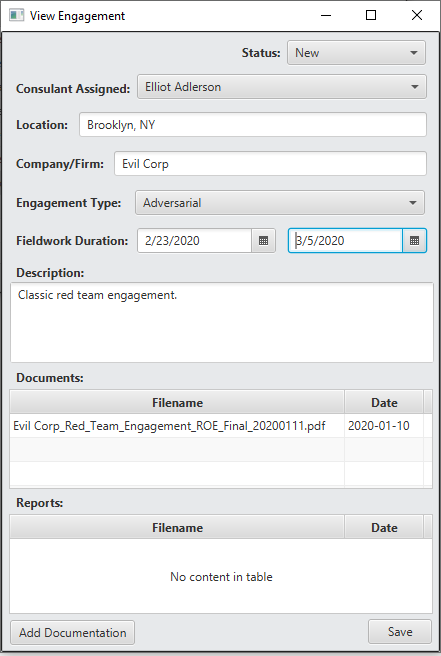


Attributes: Location, Type, and Duration End Date are all changed to different values:

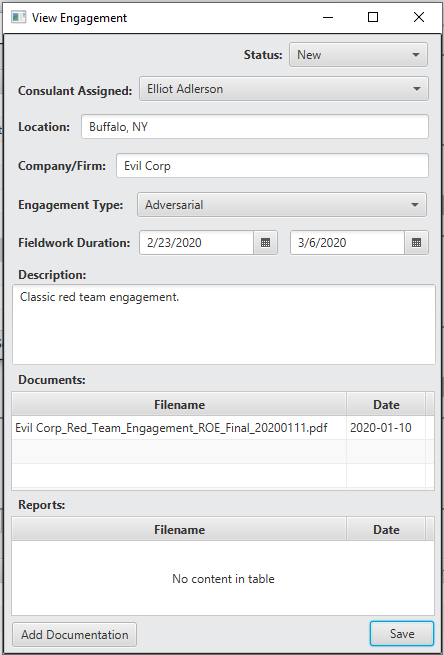
Location: Buffalo, NY 🡪 Brooklyn, NY

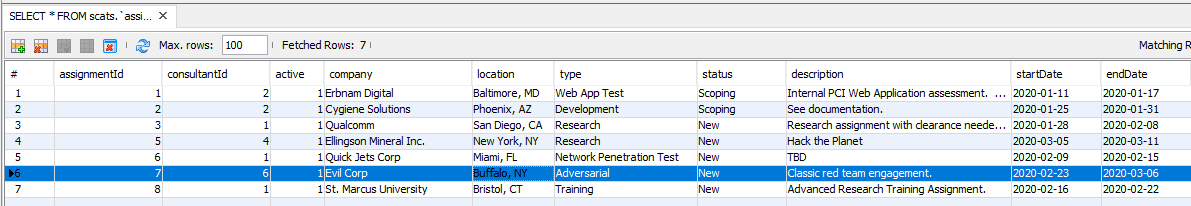
Type: Red Team Engagement 🡪 Adversarial

Duration End Date: 2/29/2020🡪 3/5/2020



2/3 changes made to the engagement are successful. The Fieldwork duration time is stored one day ahead   
of the specified value.





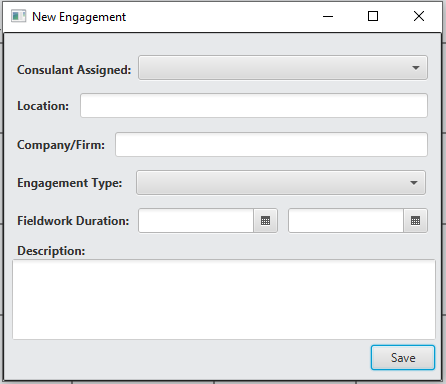
Test Results Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case009 | 1/19/2020 | FAIL | Minimal | The end date of the Fieldwork Duration is inaccurate. | Fix: address the UpdateEngagement function in the Database Manager. |

## **Module (Case#010)**: NewEngagementController.java [Test Passed]

**Description**: Provides an interface to add a new Consultant to the SCATS System.

**Interface**:



**Inputs to Module**:

* Consultant Name <String> -- selected from existing items
* Location <String> -- provided as text
* Company/Firlm <String> -- provided as text
* Type <String> -- selected from existing items
* Fieldwork duration dates <Date> -- selected from a DatePicker
* Description <String> provided as text

**Outputs from Module**:

* Module will validate all inputs into the form, ensuring all necessary elements are populated
* Module will successfully store a new engagement to the SCATS System
* Engagement attributes will be successfully persisted in the backend database
* New Engagement will be visible within the SCATS Interactive Calendar View

**Logic Flow Diagram**:

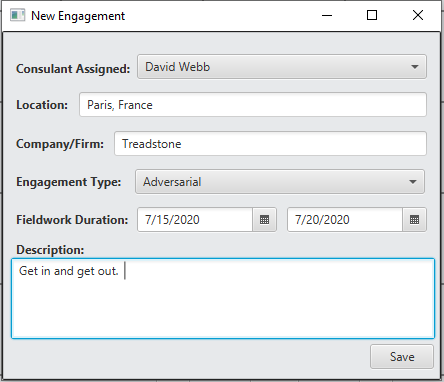
1. User selects an Engagement name
2. User supplies an Engagement location
3. User supplies a Company or Firm name
4. User selects an Engagement type
5. User specifies a start and end date to estimate the fieldwork duration
6. User clicks SAVE

**Test Data:**

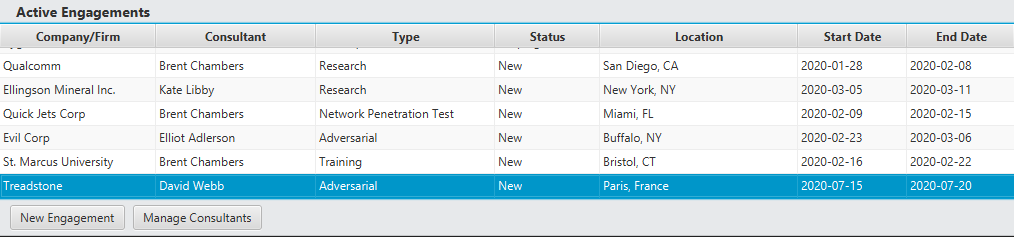
* Engagement Assignment: “David Webb”
* Location: Paris, France
* Type: Adversarial
* Company: Treadstone
* Duration: 7/15/2020 – 7/19/2020
* Description: “Get in, Get out”

**Test Case:**

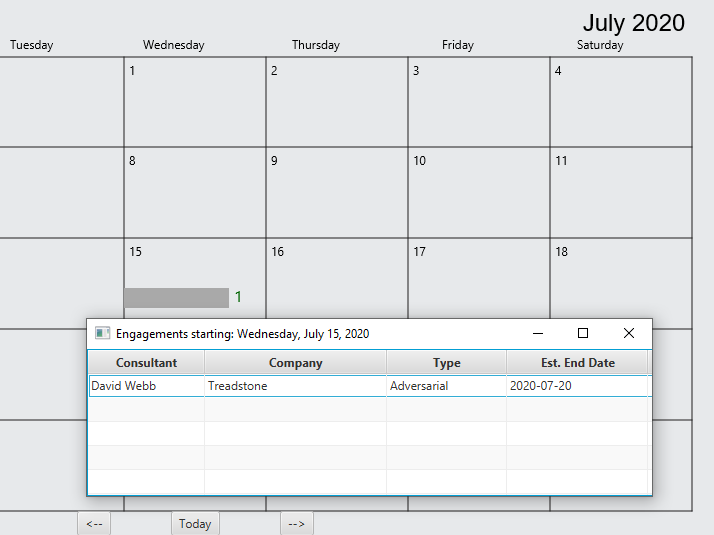
The “New Engagement” dialog was launched from the main SCATS UI interface and successfully populated with test data.



Upon population, the SAVE button was clicked and the consultant was successfully added to the SCATS system.



The Engagement is also successfully added to the Interactive Calendar, as the Engagement fieldwork start date is scheduled for 7/15/2020, with an estimated date of conclusion on 7/20.



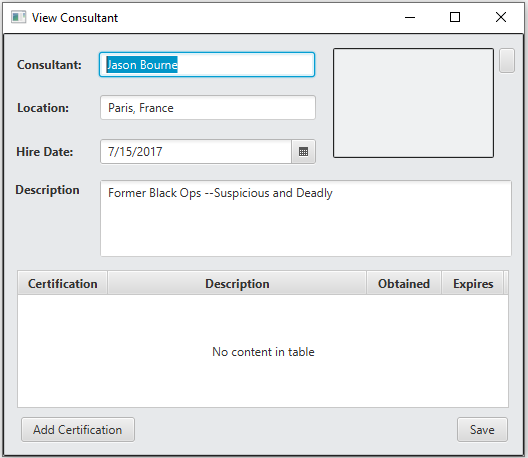
Test Results Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case010 | 1/19/2020 | Pass | None | None | None |

## **Module (Case#011)**: ViewConsultantController.java [Test Failed]

**Description**: Provides an interface to view and update the details of an existing consultant.

**Interface**:



**Inputs to Module**:

* Consultant Name <String> -- provided as text
* Location <String> -- provided as text
* Hire Date <Date> -- selected from a DatePicker to represent the hire date of the consultant
* Description <String> selected from existing items
* ProfilePic <Image/Blob> selected from Browse Dialog

**Outputs from Module**:

* Module will validate all inputs into the form, ensuring all necessary elements are populated
* Module will successfully update a new consultant to the SCATS System
* Module will successfully associate an uploaded profile picture with the Consultant, saving the image to the database as a blob
* Consultant attributes will be successfully persisted in the backend database

**Logic Flow Diagram**:

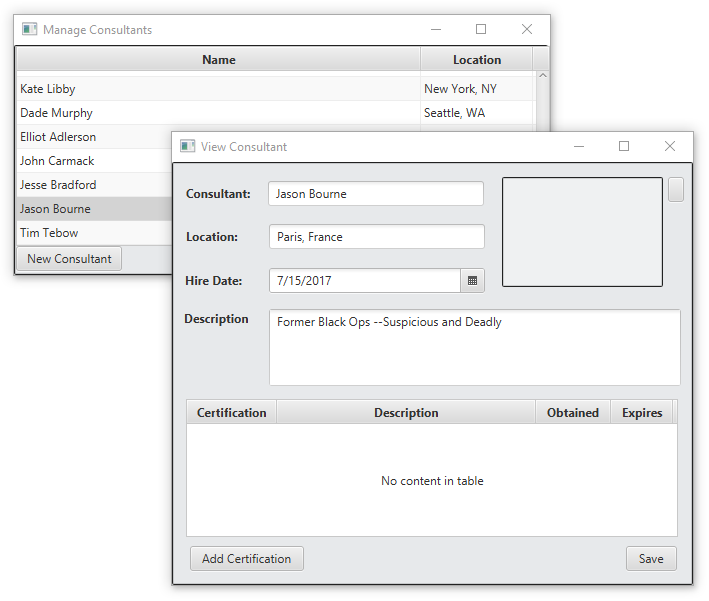
1. From the Manage Consultants tool, User right-clicks an existing consultant, and selects “View Consultant”.
2. View Consultant dialog is spawned with the form fields populated with consultant attributes.
3. User changes two Certification attributes and uploads a profile pic to be associated with the consultant.
4. User clicks SAVE
5. Changes to the Consultant should be observed within the Consultant View Consultant details form as well as the Database.

**Test Data:**

* Profile Pic: “jb.jpg”

**Test Case:**

The “View Consultant” dialog was launched from the Manage Consultants tool with consultant Jason Bourne as the item selected for modification. The View Consultant dialog successfully populated its forms with the necessary data elements that describe consultant Jason Bourne.

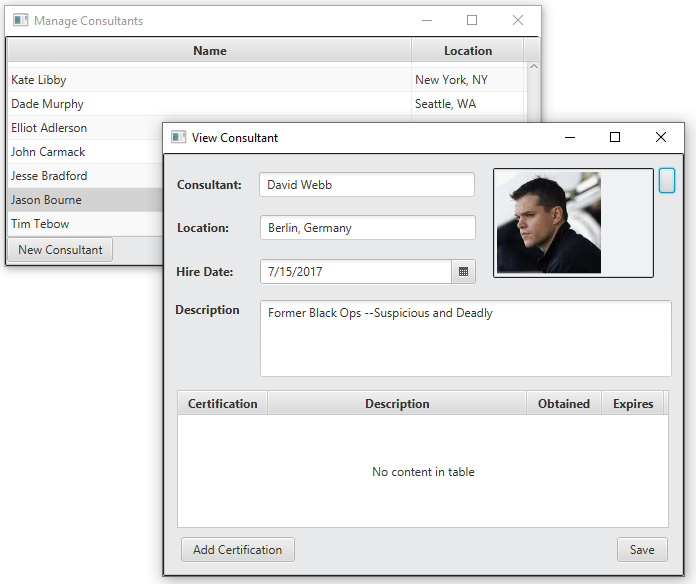


Attributes: Consultant (Name) and Location are changed to different values:

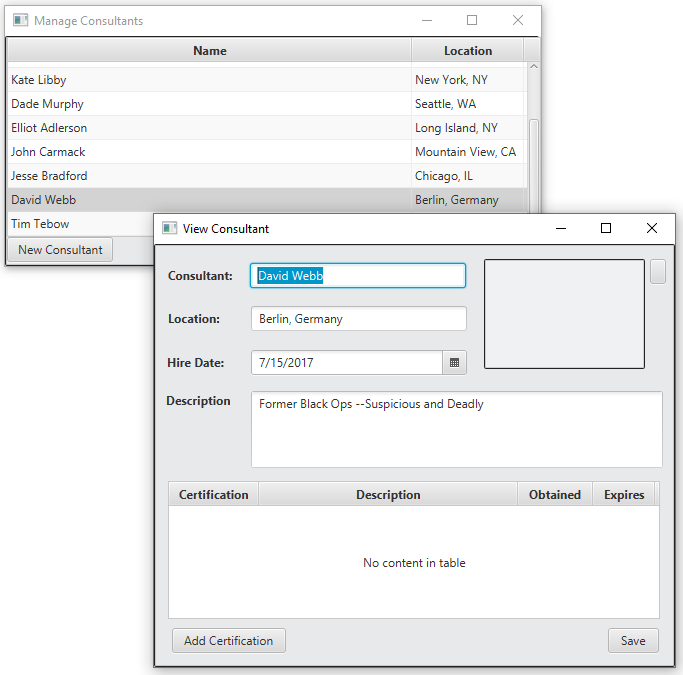
Consultant : Jason Bourne 🡪 David Webb

Location: Paris, France 🡪 Berlin, Germany

The file jb.jpg is also selected using the ProfilePicture button and rendered within the form successfully.



SAVE button was clicked and the consultant data was successfully updated to the SCATS system evident by viewing the consultant after modification, but the image file was not effectively stored.



Test Results Table:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Pass/Fail | Severity of Defect | Summary of Defect | Comments |
| Case011 | 1/19/2020 | FAIL | Medium | The system cannot find the image file because the filepath used within the function contains bad characters. | Fix is to implement relative path functionality to retrieve the file from the file system. |