10.0 Software Test Plan

Outline of Software Test Plan

10.1 Unit Test Plan

Unit tests for the CaseX API and CaseX web frontend will be written with the Mocha Testing Framework and Chai Assertion Library. Code coverage will be measured using Istanbul.

10.1.1 Unit Test Descriptions for CaseX Web Frontend

The following section describes tests specific to individual components.

10.1.1.1 Unit Test 1

Test if clicking the CaseX logo navigates to the About page.

10.1.1.2 Unit Test 2

Test if clicking the home button navigates to the Home Dashboard page.

10.1.1.3 Unit Test 3

Test if clicking the input button navigates to the Input Case Data page.

10.1.1.4 Unit Test 4

Test if clicking the explorer button navigates to the Data Explorer page.

10.1.1.5 Unit Test 5

Test if clicking the case button navigates to the Individual Case Information page.

10.1.1.6 Unit Test 6

Test if clicking the admin button navigates to the Administrator Console page.

10.1.1.7 Unit Test 7

Test if inputting a DR number and clicking the search button navigates to the appropriate individual case page.

10.1.1.8 Unit Test 8

Test if the frontend blocks the user from submitting a case form with empty fields.

10.1.1.9 Unit Test 9

Test if clicking the logout button navigates the user back to the login page.

10.1.1.10 Unit Test 10

Test that the frontend blocks the user from submitting an add user form with any empty fields on the Administrator Console page.

10.1.1.11 Unit Test 11

Test that the frontend blocks the user from issuing a delete user command without first selecting a user on the Administrator Console page.

10.1.1.12 Unit Test 12

Test that when a query condition is disabled, the user cannot modify it on the Data Explorer page.

10.1.1.13 Unit Test 13

Test that the query help module loads when the user clicks the "Additional Help" button on the Data Explorer page.

10.1.1.14 Unit Test 14

Test that clicking the delete button prompts the user that they are permanently deleting a case file.

10.1.1.15 Unit Test 15

Test that the login button on the Login page prompts the user if the email and password fields are empty.

10.1.1.16 Unit Test 16

Test that the login button on the Login page navigates the user to the Home page if the email and password fields are correct.

10.1.2 Unit Test Descriptions for CaseX API

The following section describes tests specific to individual components.

10.1.2.1 Unit Test 1

Test if POST /cases creates a case and returns status code 201.

10.1.2.2 Unit Test 2

Test if POST /cases returns status code 404 if the incorrect victim, user, or suspect id is passed in the body of the request.

10.1.2.3 Unit Test 3

Test if POST /cases returns status code 404 if the DR Number passed in the body of the request already exists.

10.1.2.4 Unit Test 4

Test if GET /cases retrieves all cases and returns status code 200.

10.1.2.5 Unit Test 5

Test if GET /cases/:id retrieves a case by DR Number and returns status code 201.

10.1.2.6 Unit Test 6

Test if GET /cases/:id returns status code 404 if the DR number doesn't exist.

10.1.2.7 Unit Test 7

Test if PUT /cases/:id modifies a cases by DR Number and returns status code 201.

10.1.2.8 Unit Test 8

Test if PUT /cases/:id returns status code 400 if there is an error.

10.1.2.9 Unit Test 9

Test if DELETE /cases/:id deletes a cases by DR Number and returns status code 201.

10.1.2.10 Unit Test 10

Test if DELETE /cases/:id returns status code 400 if there is an error.

10.1.2.11 Unit Test 11

Test if POST /victims creates a victim and returns status code 201.

10.1.2.12 Unit Test 12

Test if POST /victims returns status code 400 if there is an error.

10.1.2.13 Unit Test 13

Test if GET /victims retrieves all victims and returns status code 200.

10.1.2.14 Unit Test 14

Test if GET /victims/:id retrieves a victim by _id and returns status code 200.

10.1.2.15 Unit Test 15

Test if GET /victims/:id returns status code 404 if the id doesn't exist.

10.1.2.16 Unit Test 16

Test if PUT /victims/:id modifies the victim with _id = id, and returns status code 201.

10.1.2.17 Unit Test 17

Test if PUT /victims/:id returns status code 400 if there is an error.

10.1.2.18 Unit Test 18

Test if DELETE /victims/:id deletes a victim with _id = id, and returns status code 201.

10.1.2.19 Unit Test 19

Test if DELETE /victims/:id returns status code 400 if there is an error.

10.1.2.20 Unit Test 20

Test if POST /suspects creates a suspect and returns status code 201.

10.1.2.21 Unit Test 21

Test if POST /suspects returns status code 400 if there is an error.

10.1.2.22 Unit Test 22

Test if GET /suspects retrieves all suspects and returns status code 200.

10.1.2.23 Unit Test 23

Test if GET /suspects/:id retrieves a suspect by _id and returns status code 200.

10.1.2.24 Unit Test 24

Test if GET /suspects/:id returns status code 404 if the id doesn't exist.

10.1.2.25 Unit Test 25

Test if PUT /suspects/:id modifies the suspect with _id = id, and returns status code 201.

10.1.2.26 Unit Test 26

Test if PUT /suspects/:id returns status code 400 if there is an error.

10.1.2.27 Unit Test 27

Test if DELETE /suspects/:id deletes a suspect with _id = id, and returns status code 201.

10.1.2.28 Unit Test 28

Test if DELETE /suspect/:id returns status code 400 if there is an error.

10.1.2.29 Unit Test 29

Test if POST /users creates a user and returns status code 201.

10.1.2.30 Unit Test 30

Test if POST /users returns status code 400 if there is an error.

10.1.2.31 Unit Test 31

Test if GET /users retrieves all user and returns status code 200.

10.1.2.32 Unit Test 32

Test if GET /users/:id retrieves a user by _id and returns status code 200.

10.1.2.33 Unit Test 33

Test if GET /users/:id returns status code 404 if the id doesn't exist.

10.1.2.34 Unit Test 34

Test if PUT /users/:id modifies the user with _id = id, and returns status code 201.

10.1.2.35 Unit Test 35

Test if PUT /users/:id returns status code 400 if there is an error.

10.1.2.36 Unit Test 36

Test if DELETE /users/:id deletes a user with _id = id, and returns status code 201.

10.1.2.37 Unit Test 37

Test if DELETE /users/:id returns status code 400 if there is an error.

10.1.2.38 Unit Test 38

Test if POST /authenticate creates an JWT authentication token and returns status code 201.

10.1.2.39 Unit Test 39

Test if POST /authenticate returns status code 400 if there is an error authenticating a user.

10.2 Integration Test Plan

10.2.1 Integration Test Descriptions

The following section describes tests specific to interactions between two or more components.

10.2.1.1 Integration Test 1

Test that clicking the Import Excel file button on the Input Case Data page fills the UI with the found data.

10.2.1.2 Integration Test 2

Test that when submitting a case form with an exiting DR# on the Input Case Data page, the frontend correctly displays the error generated by the backend.

10.2.1.3 Integration Test 3

Test that when a user submits an invalid existing victim ID on the Input Case Data page, the frontend correctly displays the error generated by the backend.

10.2.1.4 Integration Test 4

Test that when a user submits an invalid existing suspect ID on the Input Case Data page, the frontend correctly displays the error generated by the backend.

10.2.1.5 Integration Test 5

Test that the frontend correctly displays a list of current users on the delete user tab on the Administrator Console page.

10.2.1.6 Integration Test 6

Test that after deleting a user, the frontend correctly updates the list of current users on the delete user tab on the Administrator Console page.

10.2.1.7 Integration Test 7

Test that after adding a new user, the frontend correctly updates the list of current users on the delete user tab on the Administrator Console page.

10.2.1.8 Integration Test 8

Test that when a user tries to add a new user with an invalid employee ID, the frontend correctly displays the error generated from the backend.

10.2.1.9 Integration Test 9

Test that clicking the edit button in the individual case page allows the user to edit the case information.

10.2.1.10 Integration Test 10

Test that editing an individual case page and clicking the save button successfully updates the case data.

10.2.1.11 Integration Test 11

Test that after clicking the delete button and receiving the warning prompt, the user is able to delete a case file.

10.2.1.12 Integration Test 12

Test that the Data Explorer page correctly displays the list of current cases in the results table.

10.2.1.13 Integration Test 13

Test that the Data Explorer page updates the results table after successfully adding a new case.

10.2.1.14 Integration Test 14

Test that the Data Explorer page updates the results table after successfully deleting an existing case.

10.2.1.15 Integration Test 15

Test that the login button on the Login page prompts the user if the email and password fields are incorrect.

10.3 Module Dependencies

The following section describes conditions where certain tests must pass before other tests are performed to ensure proper results.