**CS5002**

**Test Plan**

**Jason Russell**

**My plan is to focus on the functionality and the user requirements. Majority of tests will be done when the project is built into a test environment. In the test environment we will run tests for each function that comes with the Citation and User entity (e.g. Add and Edit). These tests will be used to confirm that functionality between the front-end, back-end and database. There are a couple of tests that will be used to test certain logic that will be required by this program. For example, there will be methods that are used to check if a Citation is past due. These methods will be tested by unit testing with specific libraries. More tests will be added to this plan.**

C1.1 Display Citations Table

C1.2 Verify that the homepage displays all citations correctly.

C1.3 Load the homepage and check if the table displays all citations fetched from the database.

C1.4 None (fetch all citations from the database).

C1.5 A table with all citations, including license plate, violation type, date, and actions (edit/print).

C1.6 Normal

C1.7 Blackbox

C1.8 Functional

C1.9 Integration

C2.1 Add Citation

C2.2 Verify that a new citation can be added successfully to the database.

C2.3 Click “Enter Citation”. Fill out the form with valid citation data in Edit Citation Page (license plate, violation type, date, location, officer name, etc.). Click “Save”

C2.4 Valid citation data (e.g., license\_plate: "ABC123", violation\_type: "Parking in a no-parking zone", date: "2023-10-01", location: "Main St", officer: "John Doe").

C2.5 The new citation appears in the table on the homepage.

C2.6 Normal

C2.7 Blackbox

C2.8 Functional

C2.9 Integration

C3.1 Edit Citation

C3.2 Verify that a citation can be edited successfully.

C3.3 Click the "Edit" button for a citation, modify its details, and save the changes.

C3.4 Updated citation data (e.g., owner\_name from null to “John Doe”).

C3.5 The updated citation details are reflected in the table on the homepage.

C3.6 Normal

C3.7 Blackbox

C3.8 Functional

C3.9 Integration

C4.1 Print Citation

C4.2 Verify that the "Print" button generates a printable version of the citation.

C4.3 Click the "Print" button for a citation and check if the print dialog opens.

C4.4 None (click the "Print" button).

C4.5 Print dialogue shows, citation data appears in correct format in the print preview, and the citation prints.

C4.6 Normal

C4.7 Blackbox

C4.8 Functional

C4.9 Integration

C5.1 Calculate Past Due Citation

C5.2 Verify that citation is marked past due if it is 7 or more days old

C5.3 Call the function with a citation that has a date that is 7 or more days old, then check the returned value

C5.4 Citation Date that is 7 or more days old

C5.5 Return True so the citation can be marked past due

C5.6 Normal

C5.7 Whitebox

C5.8 Functional

C5.9 Unit

C6.1 Invalid Citation Data

C6.2 Verify that the system handles invalid citation data gracefully.

C6.3 Submit a form with missing or invalid citation details. Each invalid detail should generate its own error.

C6.4 Invalid citation data (e.g., empty license plate, invalid date format, invalid CAD\_Number, no violation selected).

C6.5 An error message is displayed based on the invalid detail. Citation is not entered into the database

C6.6 Abnormal

C6.7 Blackbox

C6.8 Functional

C6.9 Integration

U1.1 Add User

U1.2 Verify that a new user can be added successfully.

U1.3 Submit a form with valid user details

U1.4 Valid user data (e.g., username: "officer1", password: "password123", role: "officer", first\_name: “John”, last\_name: “Doe”).

U1.5 The new user appears in the user table.

U1.6 Normal

U1.7 Blackbox

U1.8 Functional

U1.9 Integration

U2.1 Edit User

U2.2 Verify that a user's details can be edited successfully.

U2.3 Click the "Edit" button for a user, modify their details, and click “Save”.

U2.4 Updated user data (e.g., from role: "officer” to “admin”).

U2.5 The updated user details are reflected in the user table.

U2.6 Normal

U2.7 Blackbox

U2.8 Functional

U2.9 Integration

U3.1 Delete User

U3.2 Verify that a user can be deleted successfully.

U3.3 Click the "Delete" button from the Edit User Page and confirm the deletion.

U3.4 None (click the "Delete" button).

U3.5 The user is removed from the user table.

U3.6 Normal

U3.7 Blackbox

U3.8 Functional

U3.9 Integration

U4.1 User Authentication 1

U4.2 Verify that only authenticated users can access the system.

U4.3 Attempt to access the homepage without logging in.

U4.4 None (unauthenticated request).

U4.5 Redirect to the login page.

U4.6 Normal

U4.7 Blackbox

U4.8 Functional

U4.8 Integration

U5.1 User Authentication 2

U5.2 Verify that users can only access when using the correct credentials

U5.3 Try logging using the wrong username and correct password. Then try logging in with the correct username and wrong password

U5.4 Incorrect credentials

U5.5 Same error is given for incorrect password and username

U5.6 Normal

U5.7 Blackbox

U5.8 Functional

U5.9 Integration

DB1.1 Database Connection Failure

DB1.2 Verify that the system handles database connection failures gracefully.

DB1.3 Simulate database connection failure and attempt to fetch citations.

DB1.4 None (simulate a database failure).

DB1.5 An error message is displayed to the user.

DB1.6 Abnormal

DB1.7 Whitebox

DB1.8 Performance

DB1.9 Integration

DB2.1 Large Dataset Performance

DB2.2 Verify that the system performs well with a large number of citations.

DB2.3 Load the homepage with 10,000 citations in the database.

DB2.4 Large dataset (10,000 citations).

DB2.5 The homepage loads within an acceptable time frame (e.g., < 2 seconds).

DB2.6 Boundary

DB2.7 Whitebox

DB2.8 Performance

DB2.9 Integration

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Normal/Abnormal | Blackbox/Whitebox | Functional/Performance | Unit/Integration |
| C1 | Normal | Blackbox | Functional | Integration |
| C2 | Normal | Blackbox | Functional | Integration |
| C3 | Normal | Blackbox | Functional | Integration |
| C4 | Normal | Blackbox | Functional | Integration |
| C5 | Normal | Blackbox | Functional | Unit |
| C6 | Abnormal | Blackbox | Functional | Integration |
| U1 | Normal | Blackbox | Functional | Integration |
| U2 | Normal | Blackbox | Functional | Integration |
| U3 | Normal | Blackbox | Functional | Integration |
| U4 | Normal | Blackbox | Functional | Integration |
| U5 | Normal | Blackbox | Functional | Integration |
| DB1 | Abnormal | Whitebox | Performance | Integration |
| DB2 | Boundary | Whitebox | Performance | Integration |

Test Results (Ongoing):

Throughout the building process, I tested manually that the Citations table displayed correctly. I tested adding and editing citations manually. I also created automated tests using jest that I could run using “npm test”. These tests were all successful.

To test my database connection, I have a function in my server.js that creates a URL named /test-db that I can browse to. The function will show the first citation in my SQL table if the database connection is successful.