Overall Test Plan

My approach for the testing is to create tests that demonstrate the functionality of the various components of my application. Testing will run against a sample MongoDB database with dummy data inserted. Testing will cover any conditions from normal to abnormal by testing different user inputs that could cause issues, such as input strings that contain characters that are used by MongoDB to differentiate between objects (ex. brackets, and quotation marks). Testing will ensure that the application satisfies all of the requirements that are expected of it.

Test Case Descriptions

1.1 **Register Accounts**

1.2 Test will ensure new user's can register a user and admin account.

1.3 The user will request an account and enter dummy information for the fields.

1.4 Inputs: User will need to be logged out of the application, then from the landing page they must select the Register link. User will then enter their name, email, and password into the fields and select the User radio button and press the register. This process will be repeated except the user will enter admin for the name, admin@gmail.com, and admin for the name email and password. They will also select the admin radio button and click the register button.

1.5 Output: Both the user and admin registered above are added to the MongoDB Database in the Users Collection with the password encrypted.

1.6 Normal

1.7 Blackbox

1.8 Functional

1.10 Integration Test

2.1 Valid User Login

2.2 Test will ensure that users can log into the application upon successful login

2.3 User will provide both incorrect and correct credentials, in this case their username and password. After submitting the required information the user will confirm that they have access to the web application and confirm that their login failed when using incorrect credentials..

2.4 Inputs: User will provide an incorrect username and password and hit the submit button. Then they will provide their username and password and hit the submit button again. They will then log out and log in as the admin created in test 1,

2.5 Output: User will confirm that they receive a message that they have entered an incorrect username/password. Then upon entering the correct credentials they are redirected to the All Documents Page. Upon logging in the second time as an admin they are redirected to the All Reviews Page.

2.6 Normal

2.7 Blackbox

2.8 Functional

2.10 Unit Test

5.1 Add Document

5.2 This test will ensure that qualified users can add a new Document to a collection within the database.

5.3 From the All Documents Page the user will select the option to add a new document. The user will then be redirected to a Document Editing Page. They will then enter “Sample Text” into the two empty text boxes on the screen and click the “Save Document” Button.

5.4 Input: User presses the add Document button from the document viewing page. They will then enter “Sample Text” into the two empty text boxes on the screen and click the “Save Document” Button. They then Click the “Save” button within the confirmation dialog

5.5 Output: User confirms that they are redirected to the document editing page when hitting the Add Document button. User confirms that upon hitting Save button their url changes from /addDoc to /document<id of newly created doc>. Tester can also confirm that a new Document has been added to the In Work Collection on the Database.

5.6 Normal

5.7 Blackbox

5.8 Functional

5.10 Unit Test

6.1 Delete Document

6.2 This test will ensure that qualified users can delete a document.

6.3 A user will attempt to delete a document and confirm that the document is removed from the database.

6.4 Input: From the All Documents Page the user will select the Delete button in the row containing the document created in Test 5. They will then select the Delete button in the dialog that appears.

6.5 Output: User will confirm that they are prompted with the are you sure dialog when selecting delete collection. Upon pressing ”Yes” the user will confirm that the Test Document is removed from the In Work Collection on the database and the documents table.

6.6 Normal

6.7 Blackbox

6.8 Functional

6.10 Unit Test

7.1 Add Key-Value Pair in Document

7.2 This test will ensure that the user will be able to add a key-value pair to the document.

7.3 The user will select the Add button and enter SampleTextItem into the two text boxes in the newly added row.

7.4 Input: User will select the Add button and then enter “SampleTextItem” into the two text boxes in the newly added row. Then the user will press the save document button.

7.5 Output: User confirms that a new row with empty text boxes are added to the table upon pressing “Add” button. Then upon pressing the save button they confirm that the object is added to the correct document within the database.

7.6 Normal

7.7 Blackbox

7.8 Functional

7.10 Integration Test

8.1 Make HTML Editor Entry

8.2 This test will ensure that the user is able to add a tinyMCE editor into the document and add html to the database.

8.3 The user will press the HTML button in the row for the item created in Test 7 and enter dummy values before saving.

8.4 Input: The user will press the HTML button in the row for the item created in Test 7. The User will then select Heading 1 from the drop down within the text editor that appears. They will then type “Sample Header” into the box the hit their Enter key then type Paragraph content. The User will finally press the Save Document button.

8.5 Output: User confirms that upon pressing the HTML button that a TinyMCE HTML editor appeared in the row in the Value column. Upon hitting the Save Document button the contents of the document were added to the In Work Collection correctly.

8.6 Normal

8.7 Blackbox

8.8 Functional

8.10 Integration Test

9.1 Make List Entry

9.2 This test will ensure that the user is able to add a list editor into the document and an array

9.3 The user will press the List button in the row for the item created in Test 7 and enter dummy values before saving. This will add an array to the database.

9.4 Input: The user will press the List button in the row for the item created in Test 7. The User will then enter Item 1 into the text box that appears in the Value column. The user will then

Press the Append Item to List button twice. The user will then press the X button beside the final row in the table of list items. The user will then click the delete button in the dialog that appears. In the remaining empty text box the user will enter item2. The user will then hit the Save document button.

9.5 Output: User confirms that upon pressing the List button that aList editor appeared in the row in the Value column. Upon hitting the X button a dialog appeared and upon hitting Delete the list item text box was removed from the row. Upon hitting the Save Document button the contents of the document were added to the In Work Collection correctly.

9.6 Normal

9.7 Blackbox

9.8 Functional

9.10 Integration Test

10.1 Remove Key-Value pair in Document

10.2 This test will ensure that the user will be able to remove a key-value pair from the document.

10.3 The user will click the Delete button in the same row as the item created in Test 7 and select the Delete object button in the dialog that appears. They will then select the save document button and confirm that the object is removed from the document.

10.4 Input :The user will click the Delete button in the same row as the item created in Test 7 and select the Delete object button in the dialog that appears. The User then selects the save document button.

10.5 Output: User confirms that a dialog appears when selecting the Delete button from the table. User confirms that the row is removed from the table when the Delete option is selected. Then confirm that the object is removed from the database when the save button is selected.

10.6 Normal

10.7 Blackbox

10.8 Functional

10.10 Integration Test

11.1 Submit document for Review

11.2 This test will ensure that the user will be able to Submit a document for Review.

11.3 The user will click the Submit for Review button for one of the documents and ensure that the status and actions in the row are updated.

11.4 Input : From the All Documents Page the user will click the Submit for Review button for one of the In Work Documents.

11.5 Output: User confirms that the status in the same row is changed to Submitted for Review and there are no longer action buttons in the first column of the same row. The tester will then confirm that the _status item in the corresponding document on the In Work collection is changed to “Submitted for Review” and a review object was created in the Review Collection on the database with the correct date and user.

11.6 Normal

11.7 Blackbox

11.8 Functional

11.10 Integration Test

12.1 Accept Changes for Submitted Document

12.2 Confirms that an admin can review and accept changes to a document using the application.

12.3 From the All Reviews Page an admin will navigate to the Review created in Test 11 and hit the accept Changes button. This will confirm that the accepted change will be made to the Approved Collection, the Review will be removed from the Review Collection and the Action will be Logged into the Logging Collection.

12.4 Input: From the All Reviews Page press the Review Button in the same row as the Review created in Test 11. Now press the Accept Changes Button.

12.5 Output: Upon pressing the Review button the admin is redirected to the Review page for the corresponding document. Upon pressing the Accept Changes button the admin is redirected to the All Reviews Page. The Tester will then look in the In Work Collection and confirm that the corresponding document has had its _status changed to Approved. Then confirm that the review was deleted from the Reviews Collection, and that a log containing the correct information/date/admin was added to the Logging Collection and the correct data was added to the Approved Collection.

12.6 Normal

12.7 Blackbox

12.8 Functional

12.10 Integration Test

13.1 Reject Changes for Submitted Document

13.2 Confirms that an admin can review and accept changes to a document using the application.

13.3 From the All Reviews Page an admin will navigate to the Review created in Test 11 and hit the accept Changes button. This will confirm that the accepted change will be made to the Approved Collection, the Review will be removed from the Review Collection and the Action will be Logged into the Logging Collection.

13.4 Input: User must first complete the step in Test 11. From the All Reviews Page press the Review Button in the same row as the Review created in Test 11. Now press the Reject Changes Button.

13.5 Output: Upon pressing the Review button the admin is redirected to the Review page for the corresponding document. Upon pressing the Reject Changes button the admin is redirected to the All Reviews Page. The Tester will then look in the In Work Collection and confirm that the corresponding document has had its _status changed to Rejected. Then confirm that the review was deleted from the Reviews Collection, and that a log containing the correct information/date/admin was added to the Logging Collection.

13.6 Normal

13.7 Blackbox

13.8 Functional

13.10 Integration Test

Test Case Matrix

Test ID	Normal/Abnormal	Black/White box	Functional/ Performance	Unit/Integrati on	Result
1	Normal	Blackbox	Functional	Integration	Passed
2	Normal	Blackbox	Functional	Integration	Passed
3	Normal	Blackbox	Functional	Integration	Passed
4	Normal	Blackbox	Functional	Integration	Passed
5	Normal	Blackbox	Functional	Integration	Passed
6	Normal	Blackbox	Functional	Integration	Passed
7	Normal	Blackbox	Functional	Integration	Passed
8	Normal	Blackbox	Functional	Integration	Passed
9	Normal	Blackbox	Functional	Integration	Passed
10	Normal	Blackbox	Functional	Integration	Passed
11	Normal	Blackbox	Functional	Integration	Passed
12	Normal	Blackbox	Functional	Integration	Passed
13	Normal	Blackbox	Functional	Integration	Passed