# Test Results

## Test Set – Visual Inspection of the objects

Goal – To verify that input fields can be changed and key objects updated with the new info and the new info is displayed in acceptable format e.g. percentiles are formatted with percentages, large numbers with commas, etc..

Methodology - Change an entry on HTML form and verify that those change show properly in the objects by checking key arrays and objects have been updated, if a calculate field is involved use the expected results spreadsheet to verify results. Also verify when numbers are changed that the web page result is automatically reformatted correctly for readability. The key arrays and objects are checked using Google Developers console or Safari Developers console.

**Building Tab**

|  |  |  |
| --- | --- | --- |
| Input Field | Object / Notes | Test Result |
| Name | current\_building | Passed |
| Address | current\_building | Passed |
| City | current\_building | Passed |
| State | current\_building | Passed |
| Zip | current\_building | Passed |
| Property Size | current\_building | Passed |
| Purchase price | current\_building | Passed |
| Purchase date | current\_building | Passed |
| Closing Costs | current\_building | Passed |
| Improvements | current\_building | Passed |
| Terminal Cap | current\_building | Passed |

Verified that baseline object, bldgDiversey still contained the orginal hard coded data.

**Tenant Tab**

|  |  |  |
| --- | --- | --- |
| Input Field | Object / Notes | Test Result |
| Unit Size | Object: tenants array | Passed |
| Unit Size | Table Rent / SF Updated correctly | Passed |
| Tenants | Correct Table built per Selection. | Passed |

Verified the baseline\_tenants still containted the hardcode data while tenants showed the new data.

**Expenses Tab**

Goal- to verify expenses was updated and new value was formatted correctly as a percentage.

Test Passed. Verified baseline\_expenses reflected the old data and expense object reflected the updated value.

**Financing Tab**

Goal – The finance tab has three key elements of functionality. (a) updating the loan fields, (b) retrieving US Treasury bond rates from “commercial loan rate” website using their API and (c) reset button.

Methodology – To test the operation of the input fields, we follow are similar approach as above. To test the bond rate API, we will first go to website, <https://www.commercialloandirect.com/commercial-rates.php>, to see the rates are and then check the Harry Tools financing tab to see if they match. Finally, to test reset button, we will use the methods as described above.

|  |  |  |
| --- | --- | --- |
| Input Field | Object / Notes | Test Result |
| Input values | Current\_loan | Passed |
| API | Get bond rates | Passed |

Verified that Baseline\_loan continue to hold hardcoded loan values and current\_loan object is updated.

## Reset Button Tests

Goal – To verify that reset will reset that tab and only that tab’s information back to original, hard coded values.

Methodology - After the reset button is click visually inspect the web page and check key objects to verify that it now contains the hardcoded data.

|  |  |  |
| --- | --- | --- |
| Tab | Object / Notes | Test Result |
| Building | current\_building | Passed |
| Tenants | Tenants array | Passed |
| expenses | expenses | Passed |
| Financing | Current\_loan | Passed |

## Testing Graphs

Goal - To verify that graphs (a) reflect the data key metric arrays produced in dshbd\_metrics.js and (b) when the data is changed, the graphs on Dashboard are update appropriately.

Methodology - These will be performed in two steps. (a) to verify the graphs, show expected results based on the hardcoded data. The expected results defined in the expected results spreadsheets. Furthermore, the key results arrays produced by dshbd\_metrics.js will be inspected using the console. Note that further test, dshbd\_metrics.js will be tested using Jasmine.

(b) Various key dependent values will be changed, and graphs examined to reflect those changes. The dependent values are:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tab | Item Changed | Graphs impacted | | Test Result |
| Building | Purchase price + improvements | Cash-on-cash | Passed | |
| Building | Purchase Date | All Graphs | Passed | |
| Building | Terminal Cap | Terminal Value | Passed | |
| Expenses | Expense Growth | Revenue  Cash-on-Cash | Passed | |
| Financing | Loan Amount | Cash-on-Cash | Passed | |
| Financing | Interest Rate | Cash-on-Cash | Passed | |

The graphs will be compared to expected results spreadsheet, to verify results. Jasmine will be used to review flow of changed data in more precision than can be achieved through visual inspection of a graph.

## Jasmine Tests

Build two Jasmine specs. The first spec, modelSpec.js verifies that that dshbd\_metrics.js function for creating model years works properly. We first test hardcoded data against testarray to testthe buildModelYears function against existing data.

Then we purposely fail the test, by changing the test data, to verify that Jasmine is working.

Then we create test data for a 7 year model duration and test it against modelYears array generated by the buildModelYears function.

All test passed and failed as expected.

The second spec, renttestSpec.js tests that modelRent generator functions dshbd\_metrics.js are working properly by first testing the inner function that retrieves rent and then by testing outer function by creating array of rent. To run these test, we crated a new modelYear array, based on 7 years and new building purchase date and created a new tenant array, of three tenants.

All test passed as expected.