Henry Morrison-Jones ID: 1625275

COSC212 Assignment 1

Testing

While developing the Lilliput web application I took a dynamic approach to testing. This meant that testing happened alongside development and at each stage or with each new addition to the application I would try to see if I could produce errors and correct these as they came. Correcting errors in this manner allowed me to identify problems very quickly, though it meant that development took a lot longer than anticipated. Testing was conducted with one big underlying assumption. This assumption was that within the XML file, all library objects would have valid properties. As a result I implemented error correction if the file contained no libraries, no books or no content at all but did not correct for cases where libraries would be missing addresses, I.Ds, or any other basic property. I thought this assumption was justified as if the initial XML file was valid then it would continue to be valid through the administration form's validation methods.

Test Case 1: Missing XML file.
This first test case was the easiest test case to deal with. Error correction is handled within the ajax request with the 'error:' statement. In this case the window will generate an alert which informs the user that the XML file could not be retrieved and lets the user know the name of the XML it was looking for. The program also appends the the summary statistics table with the line "could not retrieve library data".

Test Case 2: Missing instances of library elements.
This test case was made with the idea that an XML file may exist but might not contain any 'library' objects or any content at all. This case was managed through a simple jQuery search through the XML file after the ajax request is made. If the search finds no library instances at all, then the script follows the same procedure as the first case whereby the user is informed that there are no libraries available through an alert and the summary statistics table is updated to display a message to the same effect.

Test Case 3: Libraries without books.
The instances of books and book objects in the XML file are essential to how the web application runs. If there are no books for a given library there will be a number of problems for the application. In this case, a library will still be displayed on the map as normal, however the contents of the library's popup reflect the fact that there are no books recorded and the number of books in the summary statistics table is recorded as zero. Hence, in this test case the web application will still function the same, just with a little less information.

Outstanding Issues

As was discussed in the testing portion of the report, an assumption was made that within the XML file, all library objects would have valid properties. As a result, the application fails when properties within any library are omitted entirely. Another outstanding issue lies within the applications administration page. Although the form validation checks IDs, coordinates, addresses, capacities and book numbers against a set of criteria, it does not check individual book entries. Thus, books in a library can be left blank and because the libraries' current book data is not given to the form, a libraries contents could be deleted with a lazy form entry.

JSHint Settings

For my JSHint settings, I made sure to include jQuery and browser environments as well as implement the "use strict" statements at the top of my module for each script. Within the script, I also removed the W117 error as it generated many problems related to the external Leaflet libraries. With these changes, the only issues generated are all soft warning, which also pertain to the external Leaflet libraries.