Test Plan

Most functions of this project are automatically tested by OUnit2. This includes all functions declared in lexer.mli, parser.mli and most functions in collection.mli, database.mli, value.mli, and document.mli. The OUnit2 tests are developed generally in a black-box approach: the tests are developed with minimum exposure to implementations and based mostly solely on specifications. When the method doesn't do a lot of work, only one test case is developed. Otherwise, edge cases and extreme values are identified and multiple test cases are devised to ensure that the function demonstrates correct behavior in a wide range of scenarios. For example, in parser.mli, a large amount of potential syntax errors are identified based on the grammar of JSON strings, and the parsing function is tested against each of them.

In some cases, test cases are omitted. This happens when the several functions are almost entirely homogeneous but they have to be written separately. This includes the helper functions for the conversion of built-in types to CamlStore types.

Some functions that produce human-readable output or require interactions with the IO are tested manually. This is because strings, when expressed as an OCaml literal, may not be human-readable and present an adverse factor on testing efficiency. If the function outputs or inputs a string, some data is generated manually by hand and the tester will thoroughly examine the output after the function is invoked to ensure that the result matches the expected behavior. Functions that are manually tested include some of the to_json and from_json functions from collection.mli and document.mli and the read and write functions in database.mli.

Our test plan demonstrates well the correctness of the system because our test cases are thorough (as could be measured by physical LOC). We tried to incorporate edge cases as much as possible to ensure that our tests cover a wide range of use cases. We also develop test cases as soon as an implementation is available to maintain the robustness of all functions depending on it.