Group 2 Testing Overview

# Database Unit Tests

Group 2 prioritized unit testing on the DB Controller objects (Table\_Access abstract class and its descendants). This controller includes the most complex logic and is most central to the functionality of the Library and Social system. The modular design of our Table\_Access family allowed us to draft tests which broadly covered the core CRUD operations of all core data model objects / tables. Unit tests can be found at test.db.TableAccessTest.java. Tests include testDelete(), testFind(), testReadAll(), testReadWrite() and testUpdate(). The TableAccessTest class includes BeforeEach and AfterEach function (setUp() and tearDown() respectively) which initialize the test database, and then reset the database. This adds considerable runtime to the unit tests, but it ensures that the testing environment is immutable. Test authors need not worry about one test impacting another test. Records in the test database before execution can be expected to be present in the same state for all tests.

Current test cases cover all basic functional cases and some edge cases, especially null values. Time did not permit comprehensive drafting of all edge cases.

A number of convenience queries were drafted in the concrete classes of Table\_Access, such as User\_Access.addGroup(). Time did not permit the development of tests for these methods, but their functionality was covered by end-to-end testing.

# End-to-End Testing

A manual End-to-End test script was developed (see docs folder) and testing was performed against it. The script covers all input scenarios in the Library view, and covers base and edge cases for data input AND business logic. As demonstrated by that test, the library program is fully functional.

# Outstanding

Time did not permit testing of all functionality. More robust coverage could be obtained by providing the additional testing described herein.

* Unit tests for LibraryController, SocialController, LibraryView, SocialView. These were seen as lower priority as their functionality was either demonstrated through the DB Controller unit testing or End to End testing.
* Social Media View End-to-End testing. This was not core to the project, and it had to be abandoned.
* Unit Testing for Database concrete-class convenience queries (like User\_Access.addGroup())
* Additional test cases for TableAccessTest class. Adding additional corner / edge tests may reveal additional bugs, and would provide better QA.