Testing Document
Software Development Group 14

For the testing of our Excel Data Extractor we use JUNIT tests and User testing. User testing consisted of bringing our application to Kimberly after we had the GUI finished and having her run it. Examine the output and giving feedback. It also consisted of running it on different computers including those that have low resolutions in the labs. We asked for feedback from her and made improvements based on areas she wanted changed or appeared to struggle with. After observing our user tests we also got ideas of where we could add useful functionality that had not been suggested by our client.

In order to test the GUI portion of our application, we decided to use Marathonite, an automation testing framework from Marathon. Marathonite allows the user to start up a copy of their application and then record actions such as button presses and file uploads which it will then generate a script for. The user can specify what language they want this script to be based in; we chose Ruby because we were more familiar with it. The user can then modify the generated script as needed and add "assert" statements to make the scripts into actual tests. The two tests we wrote were to verify the functionality of swapping axes of the resulting graph and to test the overall process. The underlying logic was not tested here because these are only meant to be tests for the graphical components and the actual code is covered with unit tests. The first test for the axes simply checks that the labels on the graph are switched when the user clicks a button. The other test goes through the whole process including uploading some files from the fixtures directory and verifies that if the user does all the steps correctly that the application will successfully exit.

Within our maven build we have all our JUNIT tests run. If any tests fail the build will fail. Below is a list of our JUNIT tests:

- TestDataGroupType
  - Tests the toString methods for Lookzone and Slide Metric
- TestDataType
  - Tests the toString methods for Statistic, Subject, and Stimuli
- TestDataTypeModelMangaer
  - Tests to make sure that the model manager makes all 6 data models
- TestFourDimArray
  - Tests getters and setters in different configurations and with overlap
  - Tests length of stored data
- TestOutputConfiguration
  - Tests for tab, column and row setters
  - Tests getting configurations after setting them

TestSheetConfiguration
    Tests setters and getters for the sheet configuration fields
TestGuiModel
    Tests updating group datatype
    Tests swapping and getting axis
    Tests updating sheet type from swap
    Tests adding good and bad files
    Tests adding file to file list model
    Tests removing selected files from list model
    Tests adding new items to the model
    Tests clearing models
    Tests selecting all a model
    Tests deselecting all in a model
    Tests setting items as true and false
    Tests getting the current active model
    Tests getting selected data
TestCheckboxListItem
    Tests setting items selected status
    Tests creating new items
TestCheckboxListModel
    Tests adding elements
    Tests contains element method
TestWorkbookReader
    Tests extracting media names from videos
    Tests extracting media names from images
    Tests getting slide metric data
    Tests getting data breaks
    Tests getting number of statistics
    Tests getting stimulus names
    Tests reading file
    Tests adding lookzone data
TestWorkbookWriter
    Tests creating rows, cells, and sheets
    Tests creating a new workbook
    Tests getting values from data structure

This is what the output should look like when all the tests are complete:

```
Results :
```

```
Tests run: 42, Failures: 0, Errors: 0, Skipped: 0
```