**Assignment 6**

**SWE 437**

**Spring 2018**

**Team Members: Ahmad Aram, Khang Chau Vo and Ranjit Singh**

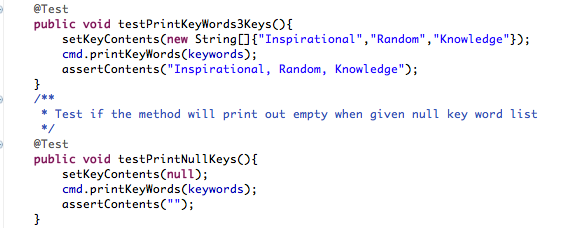
**Source Code:** [**https://github.com/davey108/SWE-437-Quotes-Server**](https://github.com/davey108/SWE-437-Quotes-Server)

The Quotes application story summary involves the user opening the application and being presented with the main menu, from here the user may select to search by keyword. This will allow them to type in a keyword that is listed and search for any quotes tagged with that keyword. It will then display the results on the screen for the user.

**User Story One:**

“As a user I will be presented with a list of keywords I can look for quotes in the quotes application.”

This user story allows us to begin implementation by ensuring that the keywords that we have preselected are displayed properly for the user to see. Below

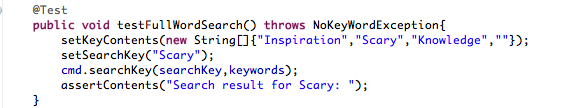


We added a new menu option to the QuotesCMD.java class to allow for the displaying of the Keywords, the above test was what we used as a basis to ensure it was operating properly.

**User Story Two:**

“As a user I will be able to use the quotes application to search a quote using a keyword, that will be presented on the screen by the application.”

This user story continues to improve the functionality of the Quotes Application by allowing the user to search for a quote using the displayed keywords that were implemented in the previous iteration. Our base test case is as follows.

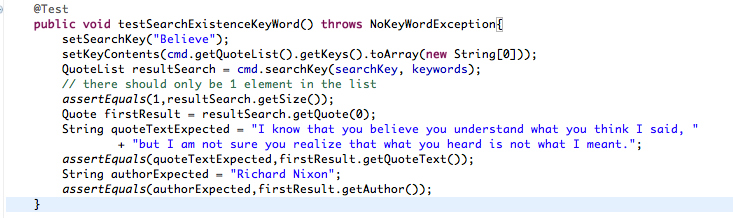


We had to change a good bit for this implementation. We began by adding a new tag “category” for every quote in the quotes.xml file. Then we added a category tag to the QuoteSaxParser.java to allow it to process the quotes. Inside of Quote.java we included a new field “category” to tag each quote based on a certain category/keyword. And finally added a getKeys() method inside of QuoteList.java to return a list of all the keys in the quotes list, which does not display duplicates.

**User Story Three:**

“As a user of the quotes application, I will be able to search for a quote using a keyword, after looking at some predefined keywords on the screen, and have the quote/quotes be displayed on the screen to look at.”

This User Story had us implement the actual searching functionality and to its completion by having the search results printed to the screen showing which quote tags/categories matched the desired query entered. Below is one of the few tests we ran to build this feature off of.



This story builds upon everything we previously did so a lot of the foundation was already there and not much was changed. During this process we did add writing to the XMLWriter.java to include the category field. And an additional custom exception to be thrown when a invalid keyword was entered as a query, this is located in the NoKeyWordException.java file.

**Screenshots:**

